University of the Philippines Manila
College of Arts and Sciences
Department of Physical Sciences and Mathematics

Tess2Speech: An Intelligent Character Recognition-To-Speech Application for Android Using Google's Tesseract Optical Character Recognition Engine

A special problem in partial fulfillment
of the requirements for the degree of
Bachelor of Science in Computer Science

Submitted by:

Anter Aaron M. Custodio

June 2016

Permission is given for the following people to have access to this SP:

Available to the general public	Yes
Available only after consultation with author/SP adviser	No
Available only to those bound by confidentiality agreement	No

ACCEPTANCE SHEET

The Special Problem entitled "Tess2Speech: An Intelligent Character Recognition-To-Speech Application for Android Using Google's Tesseract Optical Character Recognition Engine" prepared and submitted by Anter Aaron M. Custodio in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science has been examined and is recommended for acceptance.

Gregorio B. Baes, Ph.D.
Adviser

EXAMINERS:

	Approved	**Disapproved**
1. Richard Bryann L. Chua, M.Sc.	_____	_____
2. Avegail D. Carpio, M.Sc.	_____	_____
3. Perlita E. Gasmen, M.Sc.	_____	_____
4. Ma. Sheila A. Magboo, M.Sc.	_____	_____
5. Vincent Peter C. Magboo, M.D., M.Sc.	_____	_____
6. Marvin John C. Ignacio, M.Sc.	_____	_____

Accepted and approved as partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science.

Ma. Sheila A. Magboo, M.Sc.
Unit Head
Mathematical and Computing Sciences Unit
Department of Physical Sciences and Mathematics

Marcelina B. Lirazan, Ph.D.
Chair
Department of Physical Sciences and Mathematics

Leonardo R. Estacio, Jr., PHD
Dean
College of Arts and Sciences

Abstract

Tes2Speech is an Android mobile application for recognizing handwritten texts and optionally convert it to speech. By training Google's Tesseract, which is a free open-source Optical Character Recognition (OCR) Engine, to recognize handwritten texts, I created an alternative for Intelligent Character Recognition (ICR) Engine that are proprietary and expensive. OCR Engines' main purpose is to convert scanned documents with computer printed texts into an editable and machine encoded text. But by improving an OCR into ICR to recognize handwritten texts and integrating it into a mobile application in this day and age, it greatly improves its usability. It can help speech-impaired people to communicate, help children to recognize the proper pronunciation of words they write, etc. Thus, creating a free and non-proprietary handwriting-to-speech application that is accessible for everyone, and will be very useful for those who are looking for a free, portable, and reliable alternative to ICR, OCR, and other related expensive communication devices. Tes2Speech includes a user-friendly Tesseract trainer desktop application in order for the user to personalize Tess2Speech for their own handwritings.

Keywords: Android, Training, Handwritten Texts, Tesseract, Optical Character Recognition (OCR), Computer Printed Texts, Intelligent Character Recognition (ICR), Free and Non-Proprietary, Speech-Impaired

Contents

List of Figures

List of Tables

I. Introduction

A. Background of the Study

A machine performing human-like capabilities is a dream we only see in Sci-Fi movies. But over these past decades, that dream is slowly growing into reality. Many applications and programs that make machine exhibit human-like behavior were being developed such as OCRs and Text-to-Speech systems.

Optical Character Recognition (OCR) is a technology in the field of computer vision that enables us to convert scanned paper documents, PDF files, and images of computer-typed text, into an editable and searchable machine encoded-text that the computer can manipulate [1].

Intelligent Character Recognition (ICR) is an advanced OCR which allows different fonts to be learned by the computer. These fonts include different styles of human handwritings [2]. In general, ICR systems have potential applications in extracting data from filled in forms, interpreting handwritten addresses from postal documents, automatic reading of bank cheques etc.[3]

Although, there is still no existing OCR that can read all characters in an image with 100% accuracy especially when it comes to handwritten characters. According to a research paper by Ch, S., Mahna, S., and Kashyap, N. entitled Optical Character Reader on Handheld Devices [4], there are issues that they encountered while using OCR on handheld devices namely: images often suffer from skew and perspective distortion, uneven or insufficient illumination, lack of auto-focus technology on the handheld devices camera, processors of handheld devices are limited, and lack of arithmetic capabilities. These imply that the accuracy of the OCR system depends on various factors affecting the quality of the image which are: Focus of the camera, resolution of the picture, amount of noise present etc. They tested an OCR called Tesseract using their handheld devices and it achieved an average accuracy of 93% in reading non-handwritten texts.

Tesseract is an open-source OCR library written in C++ that is now being improved extensively by Google [5]. Tesseract also has the functionality to be trained to read different font styles which technically makes it an ICR. This means that Tesseract has the capability to learn human handwritings through training. Tesseract uses '.traineddata' files as a data file for recognizing texts. These files contains the character sets, character shapes, fonts, etc. that Tesseract will be able to recognize. These files are produced by using Tesseract's training tools.

Other existing systems today are under Speech Synthesis, which is a field in Computer Science which pertains to the artificial production of human speech. Android has a built-in Text-to-Speech (TTS) System since Application Program Interface (API) level 4 (Android 1.6 code name 'Donut') and recently introduced an improved TTS in API level 17 (Android 4.2.2 code name Jelly Bean) [6].

With these existing libraries and functionalities, there are numerous possibilities for creating new applications which are yet to be explored. One example of this application is a Handwritten Text-to-Speech Android application in which you write words to the device's screen and convert it to speech. This application can be very useful to patients that are suffering from different speech impairments. One specific example of a speech impairment is Aphasia – a communication disorder that results from damage to the parts of the brain that contain language [7] after a stroke. If a patient has not significantly recovered from aphasia two weeks after a stroke, speech and language therapists will assess the patients alternative means of communication possible [8]. This application could be that alternative communication method.

Smartphones and tablet technology are the ideal platforms for this type of application since these technologies has evolved to the point that almost everyone has immediate access to this device. According to the latest report of US-Based Industry tracker Strategy Analytics, nearly 2.5 billion people or 35% of the global population is expected to have smartphones by the end of 2015 [9].

There are already applications that can save handwritten text to an image file,

but only a limited few, proprietary at that, can convert handwritings on a tablet screen into computer text with significant accuracy. These phones support stylus-based screens and one of those pioneers of handwritten-to-text is the Galaxy note. One example of this is that according to a Korean news agency Yonhap, Samsung agreed to buy 5% of the Japanese company Wacom for $58.2 million just for the technology of their Stylus pens [10]. This is to improve Samsungs Galaxy Notes performance which has handwritten-to-text functionality. With these existing trends in the mobile world, there is a need for creating a free and non-proprietary handwriting recognition software.

Smart phones and tablets are also more suitable for these kinds of applications since they are more interactive compared to desktop. They are portable, have built in camera, and support touchscreen functionality.

B. Statement of the Problem

Existing OCRs today for computer-typed texts are now producing good and accurate results which can even achieve a close to 100% accuracy for certain input images [11]. Unfortunately, handwriting recognition is harder to implement due to high variability of peoples handwriting [12]. Due to this, its hard to find free handwriting ICRs that produces satisfactory results. Most of them are proprietary and expensive such as A2iA which is an ICR that can read handwritten text with satisfactory results [13]. Another problem is that it is not a mobile application and is only available for desktop computers.

By default, Tesseract cannot recognize handwritten text. Even though that it is trainable, there are still no official trained data for it that are existing that specializes in recognizing handwritten text.

People with poor eyesight have difficulties in reading PDF files or images of computer-typed texts. They tend to rely on the zoom-in capabilities of their phone. This causes to limit the words they can see on the screen and constantly swipe it to read the next word. With a portable image-to-speech application, these images can be converted to voice quickly and know the contents of these files without reading them.

People with speech impairment, one for example is Stephen Hawking, relies on Speech Synthesizer to communicate which costs a lot of money. Even with this technology, those who are older and not technologically inclined tend to have difficulties using keyboard of Speech Synthesizers compared to writing using a pen, stylus, or even fingers.

Speech impairments are conditions in which the ability to produce speech sounds that are necessary to communicate with others is impaired [14]. This can include aphasia after having a stroke, throat cancer, trauma, ALS, muscular dystrophy, people placed on vocal rest etc.

People with speech impairment that cannot afford to buy speech synthesizers, like one of those Stephen Hawking has, tend to find an inconvenient alternative

means of communications such as alphabet boards, pictures, drawing board, sign language, communication charts, and books [8]. The use of pen and paper, aside from it is hard to convey emotions through written texts, can also be inconvenient because the writer will need to give the paper to the ones that he/she is communicating to. Without a free and easy-to-use technology that can help these people, communication will remain a challenge for them.

Another problem is that pre-school students sometimes do not know how to pronounce the words that they wrote. A smart phone or tablet application that automatically pronounces the words that they write on the screen has not yet been implemented in a school even though it could prove to be very beneficial.

Even with these existing problems, HandySpeech is the only existing application in Android and iOS that can convert handwritten text to speech [15]. Another problem is that this application is not free and costs $29.99.

Another existing problem is that mobile ICRs like HandySpeech is not re-trainable, which means that the user cannot re-train their application once it is deployed. It is also not personalizable which means that the user's handwriting might not be recognized by the application. The application does not have a way to recognize your handwriting unless you contact the application developers.

C. Objectives of the Study

1. To be able to create a Tesseract Trained Data for handwritten texts with at least 80% accuracy.
2. To be able to create a user-friendly desktop application that performs Tesseract's training automatically.
3. To develop a free mobile based image-to-speech application with the following features:
   (a) Either upload an image that contains the handwriting (can also be computer printed texts) or write the handwritten text itself on the mobile's screen, and output the voice equivalent after the image processing.
   (b) Save the converted text file in either .txt or .pdf format, and the output voice in .wav format.
   (c) To allow users to personalize and re-train Tess2Speech to recognize their own handwritings even after deployment. It can be done by using Tess2Speech Trainer which is a desktop application bundled with this application, or through updates.

D. Significance of the Project

1. Creating Tess2Speech will provide users with free, portable, and non-proprietary Intelligent Character Recognition Engine.

2. Tess2Speech will provide other researchers or users with re-trainable ICR Engine. Since it is re-trainable, it will continue to improve through time [5]. This will help other researchers to develop their own ICR without spending money and time. This study also gives the researcher an idea to whether the accuracy of Tesseract is satisfactory, not only on computer-typed text, but also for handwritten text when training has been done.

3. According to HandySpeech [15], which is a non-free handwriting-to-speech application, this application can reduce the frustration and helplessness of patients suffering from speech impairments brought about by the inability to talk. Such voice impairments that this application can help are speech disorders resulting from vocal fold pathology, oral and laryngeal cancers, motor speech disorders affecting larynx and oral structures such as apraxia of speech, patients who are placed on vocal rest due to recent surgery, individuals with ALS, muscular dystrophy as long as they have enough motor control to write.

4. Unlike HandySpeech, Tes2Speech will be free and has both OCR and ICR capabilities. This means that it is not only limited on reading handwritings written on the smart phone's screen since the user also has a choice to upload an image (either by taking a photo or browse an existing image file) that contains computer-typed or handwritten texts, and convert it to both editable text and speech.

5. Tess2Speech can also help users in reading or proof-reading their typed or handwritten work because errors in written texts can be detected more apparently and can be criticized more comprehensibly when they are read out loud. Examples of these errors are the construction of sentences and mis-

spelled words [16].

6. Tess2Speech can also save their time by allowing them to convert texts on a printed paper to an editable text in their devices without needing to retype the whole paper.

7. Tess2Speech can also be useful in teaching pre-school students because sometimes pre-school students know how to write but does not know how to pronounce the words that they wrote. By using this application, the students can just write the words on the screen or take a picture of the words that they wrote, and let this application pronounce those words. The students can also check if their spelling is correct since this application also notifies the user if the converted text exists in the English Dictionary.

8. Tess2Speech can be an easier and cheaper substitute for those expensive voice synthesizers.

9. Since the user can save the recognized text into a .wav file, they can listen to their document on-the-go after scanning by using earphones or headphones.

E. Scope and Limitations

1. Handwritten characters that Tess2Speech can convert are limited to non-cursive (printed/block letters) handwritings. The letters are neither joined together nor very close that it overlaps each other to produce satisfactory results.
2. Tess2Speech only accepts PDF, JPEG, GIF, PNG, and BMP image formats as input images. Ebook with .epub extensions are also accepted as input.
3. Words that Tess2Speech can convert to speech are limited to English language. But it can be trained in future to read languages other than English.
4. Handwritten or computer typed texts that Tess2Speech can recognize is only limited to the characters present on the training set provided.
5. The training set can contain any Unicode characters (coded with UTF-8) [5].
6. Image noises such as poor illumination (see Chapter 3 C..6), skewed text, garbage on texts such as strike-through, underline, erasures, broken texts, and other lines that are not related to a character can greatly reduce Tess2Speech's accuracy.
7. Handwritten texts should be written using a black marker pen which have at least an ultra-fine point in a white paper (or any white background). The border of the paper should not be seen as it adds noise to the image and may be interpreted as a character such as 'l' (small L) or '1' (one).
8. The training is not a functionality of Tess2Speech mobile application. Training is done separately using Tess2Speech Trainer which is a desktop application. This is because as of now, Tesseract can only be trained on desktop.
9. Tess2Speech Trainer is a desktop application that can only train for English language because it always includes English dictionary when training for different fonts.

10. The accuracy of Tesseract will be dependent on the quality of the handwriting and quality of the image itself. The higher the quality of the image, the higher the accuracy of this application [5].

11. It is not recommended to use the phone's camera if the desired input contains small texts such as document, handwritten paragraphs, etc. A scanner is recommended for these types of input.

12. The quality of the converted speech will be dependent on the Text-to-Speech Engine installed by the user.

13. The performance of Tess2Speech' writable canvas is dependent on the phone's hardware (touchscreen capability).

14. Tess2Speech does not require internet connection to work.

F. Assumptions

There are several assumptions made prior to the development of this application:

1. The OS of the mobile phone that this application will be installed to is Android with minimum Application Program Interface (API) level 16 or also called as Android 4.1 (Jelly Bean).
2. There are words or characters on the images that Tess2Speech is converting.
3. The words being processed are in English Dictionary, is/are readable, not in cursive form, and as much as possible has minimal noises.
4. The input image contains black texts on a white background. If this condition is not met, the image should at least be binarizeable (see chapter 3 C..2) – which are images that can be converted to black and white.
5. The user has a Text-to-Speech Engine installed on their mobile phone.
6. The users have an SD card inserted on their mobile phones.
7. The user has a desktop computer (aside from their mobile phone) and has Java Installed.

II. Review of Related Literature

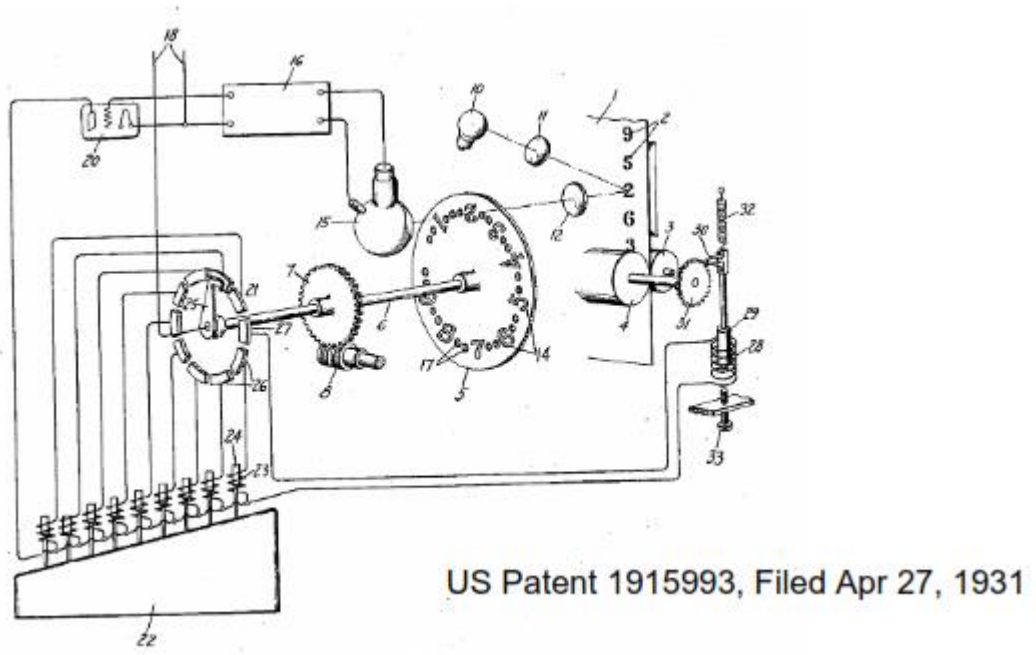
Since this application can be divided into two – Optical Character Recognition and Text-to-Speech – some researches only exists for OCRs/ICRs and some only exists for TTS.

A. Image-to-Text

A..1 Optical Character Recognition/Intelligent Character Recognition

The origin of character recognition can be found way back in 1870 and predates electronic computers (see Figure 1). During the first decades of the 19th century, several attempts were made to help develop devices to aid the blind. However, the modern version of OCR did not appear until the middle of 1940s with the development of digital computer [17].

Figure 1: Example of OCR in the 1930s when there are still no electronic computers. [18]



Modern OCRs have character recognition rates of up to 99% on high quality documents. If we have an average word length of 5 characters, this still means that one out of 20 words is defect. Thus giving us at least 5% of all processed words will have an OCR error [19]. Realistically, this OCR error will be even higher because aside from this results are not from handwritten texts; there are many factors such as the quality of the document, quality of the image, noise present in the image, and print quality.

One research paper entitled Optical Character Recognition for Handwritten Cursive English Characters [17] used a four-step method for their character recognition:

First Step: Image Acquisition

Second Step: Image Pre-processing step where they used different image pre-processing to enhance the image and convert it to a more usable form.

Third Step: Segmentation step wherein they separate the characters in a word.

Fourth Step: Feature Extraction which is the extraction of the texture and feature of the handwritten characters.

A research paper entitled Tesseract vs Gocr A Comparative Study [20] provided error analysis of most common errors in Tesseract. Some of them are:

1. 'F' is recognized as 'T'.
2. 'q' is recognized as '0'.
3. 'i' is recognized as 'l'.
4. 'o' is recognized as '0'.
5. 'p' is recognized as '0'.
6. 'd' is recognized as 'cl' or '0'.
7. 'g' is recognized as '0'.

8. '!' is recognized as 't'.

The research also proved that Tesseract indeed performs better than Gocr which is another OCR tool.

There are also countless researches about training Tesseract, one example is training Tesseract to recognize Odia Language [21] (see Figure 2). Odia language is used in the state of Odisha (formerly known as Orissa), India.

Figure 2: Consonants of Odia Alphabet. [21]

The alphabet can be considered as a very complicated alphabet but they still managed to train Tesseract to recognize its alphabets. This further proves that it is possible that Tesseract can be taught to recognize handwritten characters.

The research paper by Rakshit, S. and Basu, S. [3] trained Tesseract to recognize handwritings. They concentrated on training Tesseract to recognize only lower case characters of Roman Script. Though their training data set is small, they still managed to produce accuracy around 78.4%. They also noticed that Tesseract rejects around 9.24% characters in the data set mainly due to multi-skewed handwritten text lines in the test documents. Completely cursive words were also rejected completely in many cases during the experimentation. They also concluded that the performance of the designed system can be improved by incorporating more training data and inclusion of word-level dictionary matching techniques.

A..2 Image Pre-Processing Techniques

Another research paper entitled Intelligent Character Recognition (ICR): A Novel Algorithm to Extract Text from a Scanned Form Based Image [2] proposed a system which performs a series of image pre-processing to aid the OCR. These series of image pre-processing are Binarization using different types of thresholding, Skew Detection and De-Skewing, Slant Detection, Smoothing, Cleaning, Enhancing, Pre-Segmentation, Character Recognition, Separation of Characters, Normalization of Characters, Thinning, Grid Formation, Singular Point Determination, Line-Detection, Character Matching, and Post-processing. They have developed a system which produces adequate accuracy in reading scanned form based images.

The paper entitled Optical Character Recognition by Open Source OCR Tool Tesseract: A Case Study [22] proved that using grayscale images enhances the accuracy of Tesseract (see Table 1).

Table 1: Tesseract OCR Result Analysis after Grayscaling images. [22]

Image No	Image Type	Number of character in image	No of characters extracted	Accuracy of OCR of color images (in Percentages)	Time taken for OCR (in Seconds)	Image Type	No of characters extracted after converting color to gray scale image	Accuracy of OCR of gray scale images (in Percentages)	Time taken for OCR (in Seconds)	Change in Accuracy (In percentages)
1	color	**12**	5	42	0.4	gray scale	5	42	0.397	
2	color	**12**	12	100	0.202	color	12	100	0.202	
3	color	**12**	8	67	0.301	gray scale	8	67	0.601	
4	color	**9**	9	100	0.5	color	9	100	0.5	
5	color	**8**	8	100	0.505	color	8	100	0.505	
6	color	**9**	7	78	0.909	gray scale	7	78	0.909	
7	color	**8**	8	100	0.805	color	8	100	0.805	
8	color	**9**	7	78	1.01	gray scale	7	78	1.01	
9	color	**10**	7	70	0.85	gray scale	7	70	0.798	
10	color	**9**	4	44	0.907	gray scale	5	56	0.402	20
11	color	**10**	1	10	1.007	gray scale	4	40	0.548	75
12	color	**10**	4	40	0.699	gray scale	7	70	0.402	42.86
13	color	**10**	3	30	1.51	gray scale	4	40	0.701	25
14	color	**9**	0	0	1.008	gray scale	4	44	0.705	100
15	color	**9**	0	0	1.815	gray scale	2	22	0.7	100
16	color	**11**	6	55	1.619	gray scale	8	73	1.717	25
17	color	**9**	5	56	0.99	gray scale	6	67	0.806	16.67
18	color	**11**	5	45	0.907	gray scale	6	55	0.596	16.67
19	color	**9**	9	100	3.048	color	9	100	3.048	
20	color	**9**	9	100	1.007	color	9	100	1.007	
			Average Accuracy	**61**			**Average Accuracy**	**70**		

A research by Ntirogiannis, K. entitled Performance Evaluation Methodology for Historical Document Image Binarization [23] further proved that Binarization of image is of great importance in document image analysis and recognition pipeline since it affects further stages of the OCRs recognition process. He concluded that converting a document into a binary image can help in document image analysis. A binary image is an image wherein there can only be two possible values for each pixel (Black or White).

B. Text-to-Speech

There are also many researchers who ventured and discussed Text-to-Speech. A study done by Chandra E. and Sasirekha, D. [24] summarized published literatures on TTS and discussed the efforts taken in each paper. They concluded that there are many free TTS libraries and applications in our market today that continues to improve. They also discussed the 5 fundamental components of TTS systems namely:

1. Text analysis and detection
2. Text normalization and linearization
3. Phonetic Analysis
4. Prosodic Modeling and Intonation
5. Acoustic Processing

C. Image-to-Speech

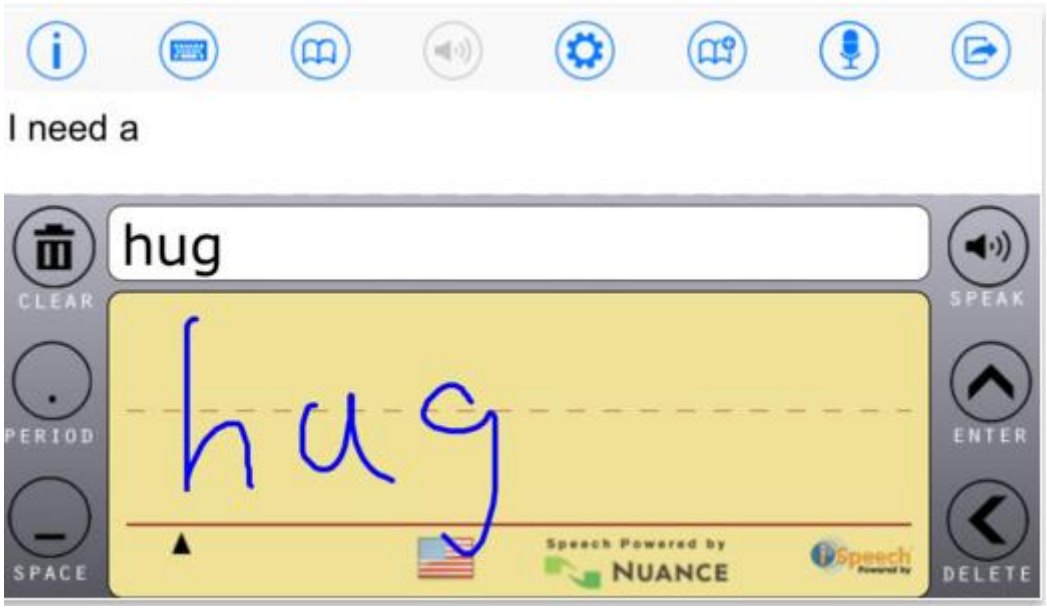
There are also researches that discussed and ventured on making a similar application of performing image-to-speech application. One such paper is Optical Character Recognition by Mithe R., Indalkar, N., and Divekar N. [25] in which they also used Tesseract as their OCR but did not train it to recognize handwritten texts. They use Speech Synthesizer to convert text into audible speech. They have successfully created a mobile application that converts an image of computer-typed text into editable text. They also stated limitations of their application which are:

1. Accuracy of an OCR system is directly dependent on the quality of input document.
2. The output from OCR systems is often quite noisy and garbled. In order to correct this, the application will perform some post processing on the text after it has received a response from the OCR package.

D. HandySpeech

HandySpeech is the only handwriting-to-speech application for iOS devices. HandySpeech lets users write on the screen of the tablet using a stylus or ones finger and this application will convert it to speech. The voices that this device can use can be a male or a female. It can learn the users handwriting style, understands cursive, print and mixed handwriting styles, and contains dictionary of unusual words. Recently, HandySpeech also implemented a voice-recognition capability which is powered by Nuance. This application was created by a 12-year old boy who was inspired by hear autistic sister which has speech disabilities [15]. Though this application can recognize text, it doesnt have the capability to read text from images captured by the phones camera. The texts that this application can read is limited to the texts written on the tablets screen which limits the number of words that it can convert to speech.

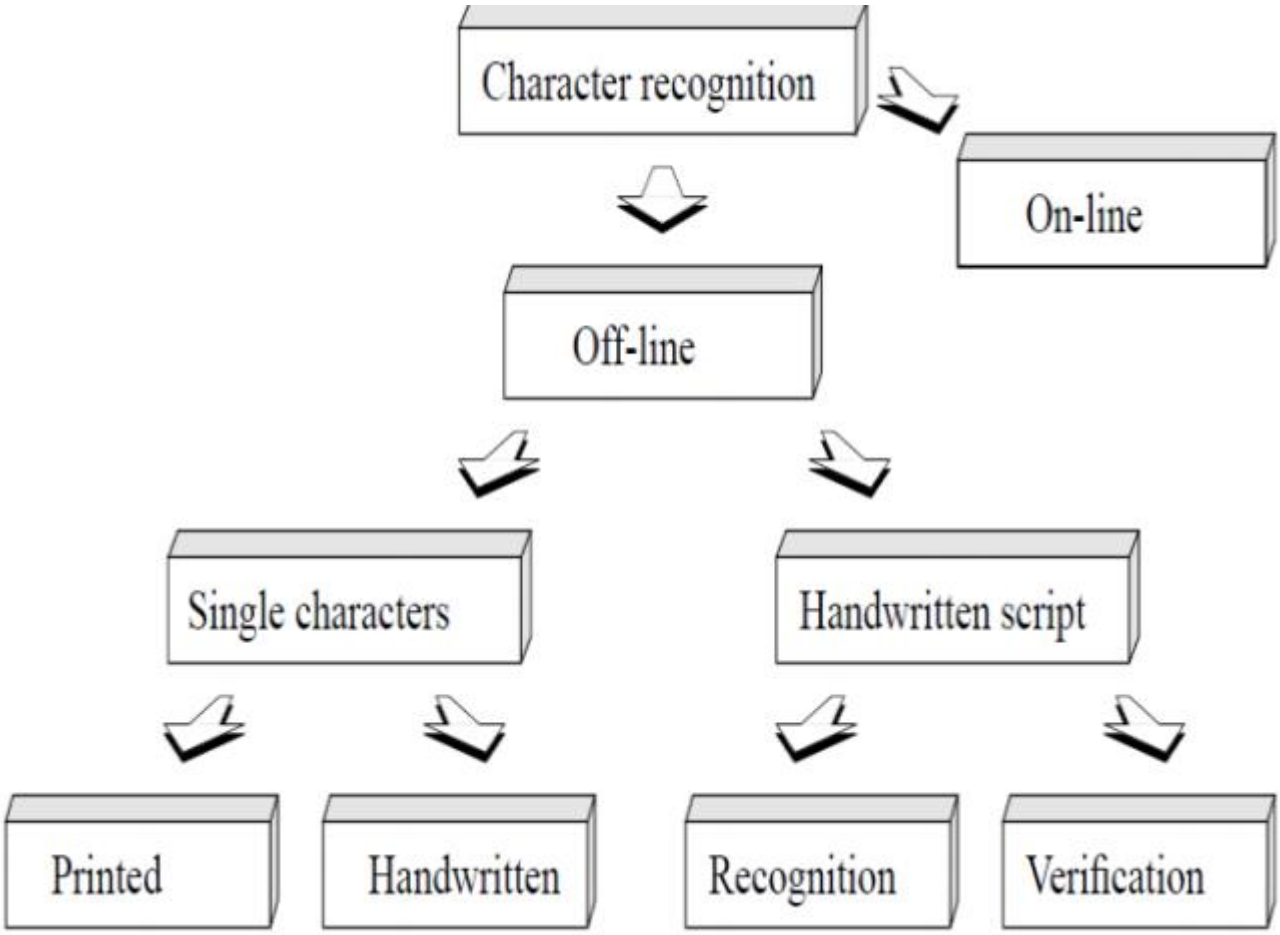
Figure 3: Screen Capture of HandySpeech Interface. [15].
Image from:(https://itunes.apple.com/us/app/handyspeech/id563600464?mt=8)



III. Theoretical Framework

There are many areas in Character Recognition:

Figure 4: Areas of Character Recognition. [26]



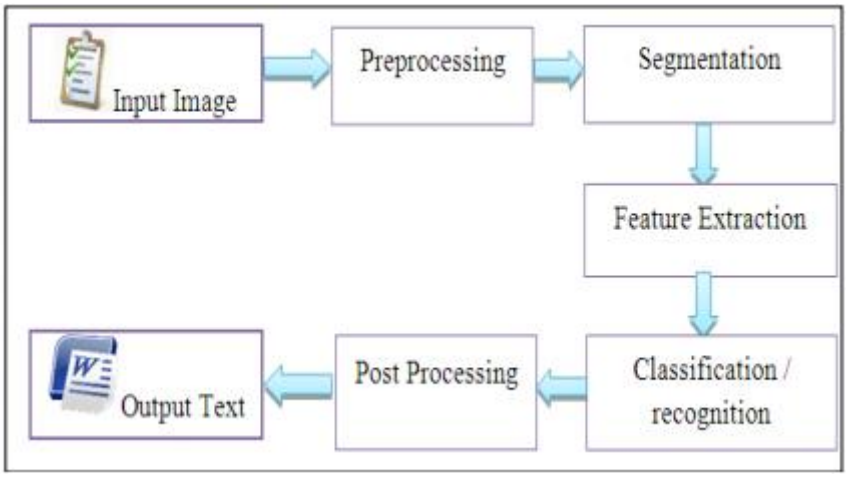
In this special project, I will venture through Off-line Handwritten Script Recognition and Verification.

A. Optical Character Recognition

Optical Character Recognition technology automatically recognizes texts from images with formats like JPG, PNG, BMP, GIF, TIFF and multi-page PDF files. OCR involves analysis of the captured or scanned images and then translate character images into character codes, so that it can be edited, searched, stored more efficiently, displayed on-line, and used in machine processes [27].

While we view text documents as lines of text, computers actually see them as binary data, or a series of ones and zeros. Therefore, the characters within a text document must be represented by numeric codes. In order to accomplish this, the text is saved using one of several types of character encoding such as ASCII and UTF-8 [1].

Figure 5: Most OCR Tool's Process would look like this. [27]



OCR tools processes are step-by-step and defined as: [27]

A..1 Input Image

These are digitalized images like a scanned or captured text images. It could be of different formats, i.e. PDF, TIFF, JPEG, GIF, PNG, and BMP image.

A..2 Preprocessing

This is a step essential for OCR systems for image handling. This is the step where the image is enhanced through different image processing techniques such

as adding or removing of noises, maintaining the correct contrast of the image, background removal, etc.

A..3 Segmentation

The accuracy of the OCR depends on the Segmentation algorithm used. Segmentation consists of different steps. First is Page Segmentation which is the separation of graphics from text. Next is Word Segmentation which is the problem of dividing a string of written language into its component words. Finally, Character Segmentation separates characters from the others.

A..4 Feature Extraction

This step refers to the extraction of the most relevant information from the text image which helps to recognize the characters in the text.

A..5 Classification/Recognition

It is the step wherein optical patters are classified into alphanumeric and other characters. The information should be readable to both human and machine.

A..6 Post Processing

The goal of this step is to increase recognition by checking grammatical errors and misspellings in the OCR output text after the input image has been scanned and completely processed.

A..7 Output Text

The editable text that is extracted from the input image is displayed in the output text.

B. Intelligent Character Recognition, Learning Classifier Systems, and Neural Networks

Intelligent Character Recognition (ICR) is an advanced OCR which allows different fonts to be learned by the computer through training. These fonts include different styles of human handwritings [2]. According to Ray Smith [18], who is from Google, Tesseract uses Adaptive Classifier as its default training mechanism. But after reading the wiki on Tesseract-OCR, there is an option called "Cube Mode" where Tesseract will switch to using Neural Network for the learning instead of the adaptive classifier. Unfortunately, Tesseract will now deprecate Cube Mode since using it is slower, and will now introduce a better adaptive classification algorithm [5].

B..1 Artificial Neural Network (ANN)

An Artificial Neural Network (ANN) or sometimes called Neural Network (NN) is a mathematical or computational model that is inspired by the way biological neural networks, like our brains, process information. A neural network consists of highly interconnected group of artificial neurons that works together to solve specific problems using a connectionist approach. ANNs, like people, are an adaptive system and can learn by examples. ANN changes its structure based on external or internal information that flows through the network during the learning phase [26].

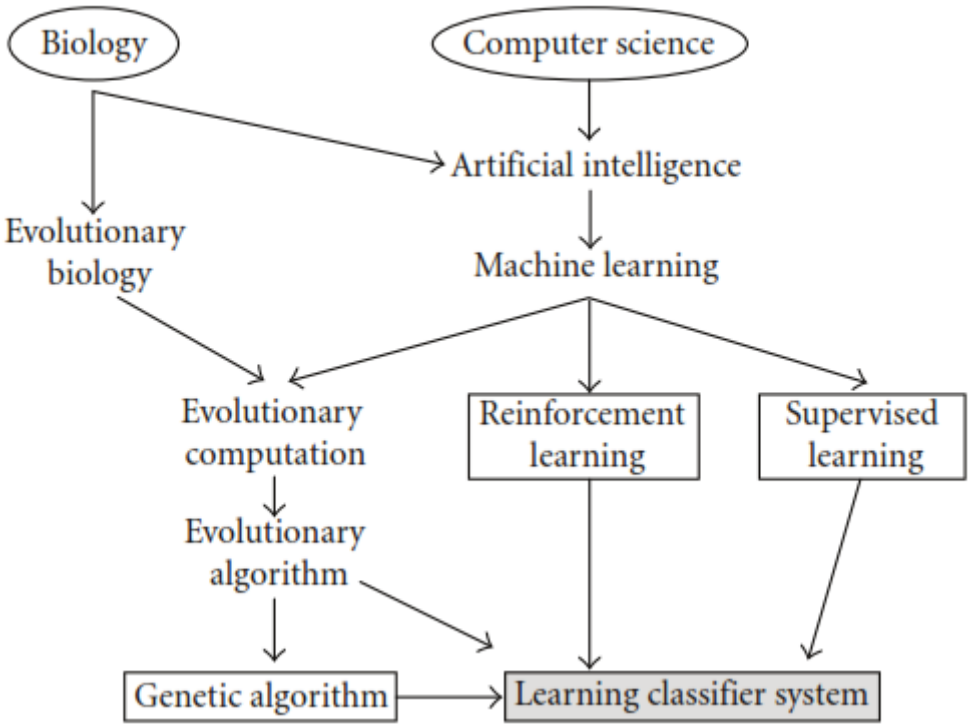
Since Tesseract's Cube Mode is now deprecated, Tesseract will not use this as their Training Mechanism anymore.

B..2 Learning Classifier Systems

Learning Classifier Systems (LCS) is one of the many Nature-Inspired Algorithms. LCS is a combination of Evolutionary algorithm from the Biology Field, and Reinforcement Learning and Supervised Learning from the Machine Learning Field [28] (see Figure 6).

Learning Classifier System algorithm optimizes payoff or reward based on exposure to stimuli from a problem-specific environment by managing credit assignment for those rules that prove useful and searching for new rules and variations on existing rules using an evolutionary process [29].

Figure 6: Field tree – foundation of the LCS Community. [28]



An LCS has four practically universal components [28]:

1. A finite population of classifiers that represents the current knowledge of the system.
2. A Performance Component, which regulates interaction between the environment.
3. A Reinforced Component, which is also called credit assignment component, which distributes reward received from the environment to the classifier.
4. A Discovery Component which uses different operators to discover better rules and improve existing one.

According to Smith [18], It has been suggested and demonstrated that using Adaptive Classifier can improve OCR engine's accuracy. It will be further discussed later on (see D.).

C. Image Processing/Preprocessing

Here are some image processing techniques that are usually used in OCRs [2]:

C..1 Grayscaling

GrayScaling is an image processing technique that converts the images color into a range of gray shades from white to black.

Basic grayscale algorithms utilize three basic steps[30]:

1. Get the Red, Blue, and Green values of a pixel (RGB Color Channels).
2. Use math to turn the combined values of Red, Blue, and Green pixel into a single gray value.
3. Replace the original Red, Blue, and Green pixel with the new gray value.

Another approach is the **desaturation** of the image. In this method, instead of using the RGB Space (Red, Green, Blue), the HSL color space is used (Hue, Saturation, lightness). Hue could be considered as the name of the color (Red, Blue, Green etc.). Mathematically, hue is described as an angular dimension on the color wheel ranging from 0°to 360°, where pure Red occurs at 0°and 360°, pure Green at 120°, and pure Blue at 240°. Saturation describes on how pure or vivid a color is. A pure color has 100% saturation while gray has 0% or no saturation. Lightness describes the brightness of a color; white has full lightness, while black has zero lightness.

This implies that by setting the saturation of an image to 0, we can easily create a grayscale image.

C..2 Thresholding & Binarization

Thresholding is an image segmentation method that converts a gray scale image into a binary image using different kinds of Thresholding algorithms. Binary image is a black and white image. Each pixel could only have 0 or 1 as values (black or white).

Tesseract is using a Binarization method called Otsu Binarization before it process an image. The method involves iterating through all the possible threshold values and calculating a measure of spread for the pixel levels each side of the threshold, i.e. the pixels that either falls in foreground or background. The aim is to find the threshold value where the sum of foreground and background spreads is at its minimum [31].

C..3 De-Skew

De-Skewing is an image processing technique that aligns tilted and skewed documents by tilting the image a few degrees clockwise or counter-clockwise. I will use an alternative for automatic de-skewing since this image processing technique is very broad and studying its implementation is another topic.

C..4 Separation of Characters/Segmentation

The idea behind separation of characters is to search the blank spaces between the individual characters to identify which are the cut-off points of the scanned word. Tesseract is already implementing this inside its function and will be further discussed at the next section (see D.).

C..5 Character Normalization

Since handwritten characters vary greatly in different styles, we normalize the characters. Normalization means that a character is made to fit into a standard size square. This means that characters that are close to the normalized character

will be considered as that normalized character. Tesseract is already implementing this inside its function.

C..6 Adjusting Brightness and Contrast

Adjusting Brightness and Contrast can help process an image with poor illumination and can also be an alternative for Thinning since increasing brightness and contrast can result to a pseudo-thinning to a character. Illumination is the amount of light present in an image.

Thinning is applied to all normalized characters. It refers to the reduction of the thickness of the normalized character to create a more general and non-strict normalized character. Thinning only retains the most significant feature or identifiers of a character.

Android can use a Matrix to manipulate a bitmap's color [6]. This Matrix is called a Color Matrix which is a 5 x 4 matrix for transforming the color and alpha components of a Bitmap. The matrix can be passed as single array, and is treated as follows:

```
[ a, b, c, d, e,
  f, g, h, i, j,
  k, l, m, n, o,
  p, q, r, s, t ]
```

Then, if we apply this matrix to a color [R, G, B, A] (where R = Red, G = Green, B = Blue, A = Alpha – transparency of bitmap), the resulting color is computed as:

```
R = a*R + b*G + c*B + d*A + e;
G = f*R + g*G + h*B + i*A + j;
B = k*R + l*G + m*B + n*A + o;
A = p*R + q*G + r*B + s*A + t;
```

That resulting color [R, G, B, A] then has each channel restricted to the 0 to 255 range.

With this knowledge, we can adjust the brightness and the contrast of an image by applying this matrix to the image:

```
contrast, 0,        0,        0, brightness,
0,        contrast, 0,        0, brightness,
0,        0,        contrast, 0, brightness,
0,        0,        0,        1, 0
```

Where the default value of contrast is 1 and can have any value from [1,10], and the default value for brightness is 0 and can have any value from [-255, 255].

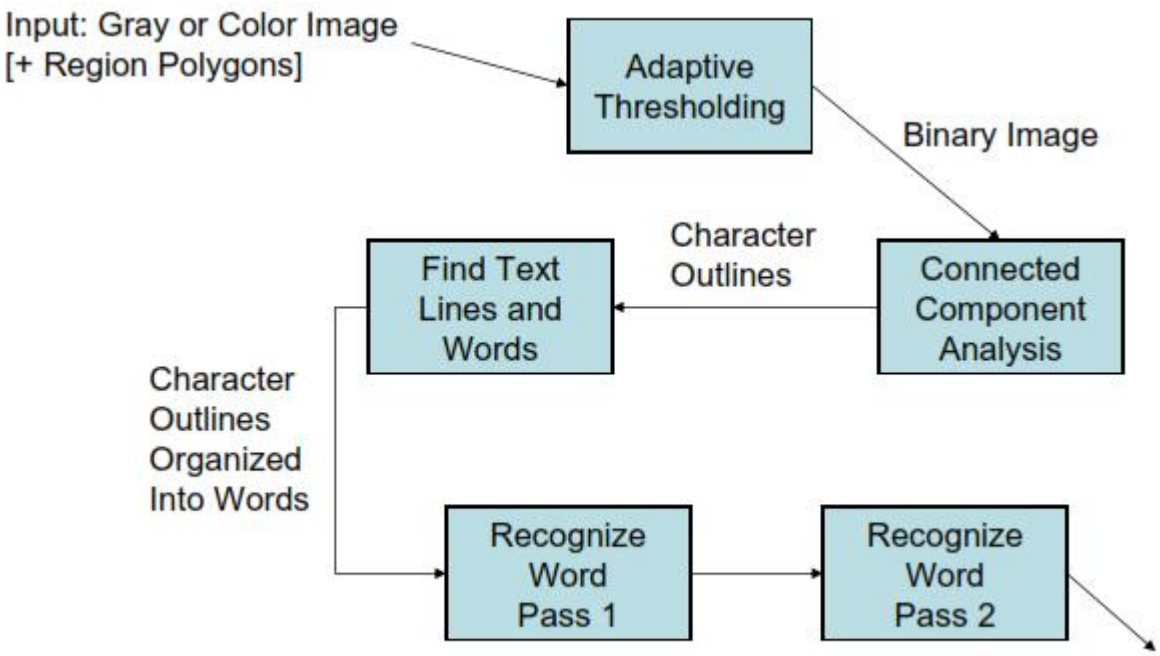
D. Tesseract

Tesseract is Free Open-source OCR engine. It was first developed between 1984 and 1994 at HP. In 1995, it was one of the top 3 engines in the 1995 UNLV Accuracy test after the joint project between HP Labs Bristol and HPs Scanner Division in Colorado. Finally in 2005, Tesseract was released as open source by HP and is now being extensively developed by Google which greatly increased its accuracy [25]. Tesseract supports over 60 languages and is combined with the Leptonica Image Processing Library which can read a wide variety of image formats and perform some image processing for Tesseract [5]. It is available at http://code.google.com/p/tesseract-ocr.

D..1 Tesseract Architecture

Tesseract OCR works in a step by step manner. These steps are shown in figure 7 and are also defined below according to the research done by Patel, C., Patel, D., and Patel, A. [22].

Figure 7: Tesseract-OCR Architecture. [18]



1. **Adaptive Thresholding**

   This is the step wherein the input image is converted into a binary image using Otsu's Binarization Method.

2. **Connected Component Analysis**

   This is the step which is used to extract the character outlines. It does the recognition of character in the image with white text and black background. These outlines are then converted into Blobs (Binary Large Objects), which a collection of binary data stored as a single entity.

3. **Find Text Lines and Words**

   This is the step wherein the Blobs are organized into text lines. Lines and regions are analyzed for some fixed area or equivalent text sizes. Texts are then divided into words. Tesseract created a line finding algorithm that allows it to recognize a skewed page without de-skewing it.

4. **Recognize Word Pass 1**

The first pass is an attempt to recognize each word from the text. Each word passed as satisfactory is passed to an adaptive classifier as a training data.

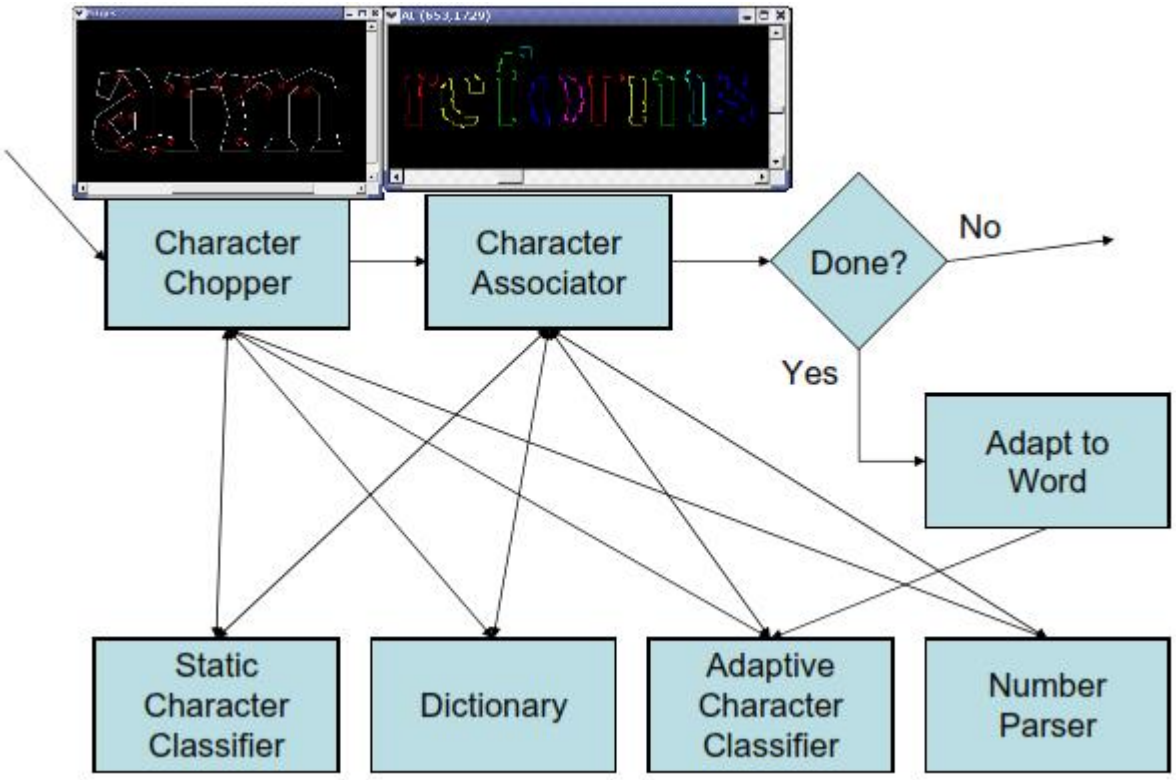
5. **Recognize Word Pass 2**

   The second pass is an attempt to recognize each word after it has learned something new using the adaptive classifier (will be explained later on). With these, various recognition issues are resolved and the texts from the images are then extracted.

D..2 Tesseract Word Recognition

Tesseract uses Classifiers to train. Before we could analyze how Tesseract trains its data, we need to know the architecture of Tesseract in recognizing words. This architecture (see Figure 8), including each steps and the Classifiers, were discussed by Smith, R. from Google Inc. as follows: [18]

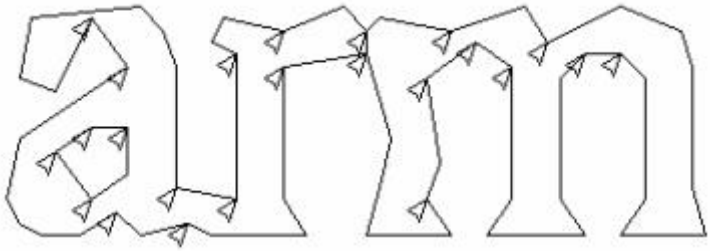
Figure 8: Tesseract-OCR Architecture in Recognizing a word. [18]



1. **Character Chopper** Tesseract attempts to improve the result by chopping the Blob with the worst confidence from the character classifier (will be discussed later on in 4). Confidence is the measure on how confident or sure the Tesseract is that the extracted word will be correct.

   These chops (see Figure 9) are executed in priority order. If the chops do not improve the confidence, the chops are undone but not completely discarded because chops can be re-used by the associator if needed.

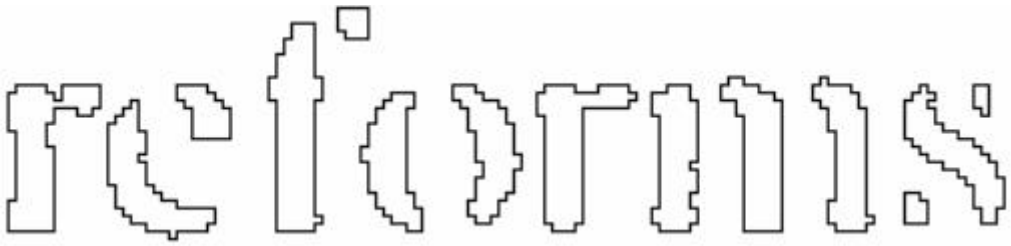
Figure 9: A set of candidate chop points with arrows and the selected chop as a line across the outline where the 'r' touches the 'm' [18]



2. **Character Associator**

   If all the potential chops have been exhausted but the words still have low confidences, it is given to the associator and performs an A* (best first) search of possible combinations of the maximally chopped blobs into candidate characters.

Figure 10: Example of easily recognized word by Character Associator [18]



3. **Dictionary and Number Parser**

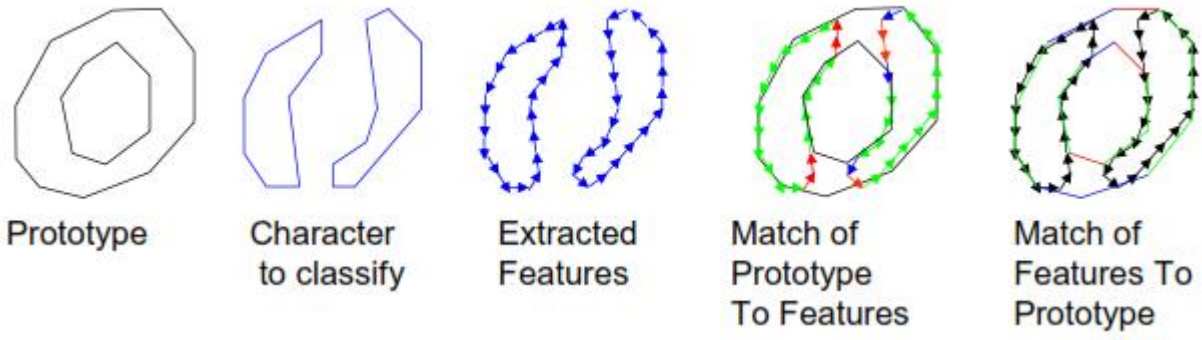
   Tesseract can use a dictionary that helps in recognizing words. A word that is in the dictionary will increase its confidence. Tesseract's built-in

dictionary can support different languages such as English, Tagalog, Indian, etc. Dictionaries can also help Tesseract identify whether the text are a number or a letter by using Punctuation and Number dictionary, aside from the Wordlist dictionary.

4. **Static Character Classifier**

   Static Character Classifier uses outline fragments extracted from the outline as features and matches it to the features of the prototype training data. There is a process called small to large matching in the classifier that can easily recognize a broken character. Performing Static Character Classification is computationally heavy. This shows that Tesseract uses polygonal approximation as input instead of the raw outlines, which is believed to be a main weakness of Tesseract.

Figure 11: Static Character Classifier way of identifying the character using the training prototype data [18]



5. **Adaptive Character Classifier**

   Since Static Character classifier focuses on generalizing any kind of font, its ability to discriminate between different characters is weakened. Due to this, a more font-sensitive adaptive classifier is used to obtain greater discrimination within each document, where the fonts are limited. Adaptive Character Classifier is trained through the output of Static Character Classifier. Adaptive Character Classifier uses the same technique as the Static

Character Classifier. The only difference is that they use different normalization techniques.

D..3 Training Tesseract

Here are the steps, commands, and required files in order to train Tesseract [5] (see block diagram of training in Figure 19):

1. To train Tesseract, **training tools for Tesseract should be installed on your computer**. These binary files should be present on your Tesseract directory:
   (a) ambiguous_words.exe
   (b) classifier_tester.exe
   (c) cntraining.exe
   (d) combine_tessdata.exe
   (e) dawg2wordlist.exe
   (f) mftraining.exe
   (g) set_unicharset_properties.exe
   (h) shapeclustering.exe
   (i) tesseract.exe
   (j) unicharset_extractor.exe
   (k) wordlist2dawg.exe

2. **Identify the required files** that we need to be able to create our trained data. Suppose that we want to name our trained data as 'engh'. Then the required files are:
   (a) engh.unicharset
   (b) engh.inttemp
   (c) engh.pffmtable

(d) engh.normproto

(e) engh.shapetable

(f) engh.config (Optional)

(g) engh.punc-dawg (Optional)

(h) engh.word-dawg (Optional)

(i) engh.unicharambigs (Optional)

(j) engh.number-dawg (Optional)

(k) engh.freq-dawg (Optional)

(l) engh.bigram-dawg (Optional)

(m) engh.unambig-dawg (Optional)

(n) engh.params-model (Optional)

(o) engh.fixed-length-dawgs (Deprecated)

(p) engh.cube-unicharset (Deprecated)

(q) engh.cube-word-dawg (Deprecated)

The combination of these files will be 'engh.traineddata', which will be the final product of our training. Tesseract can now use this file to read the font that we have trained it to.

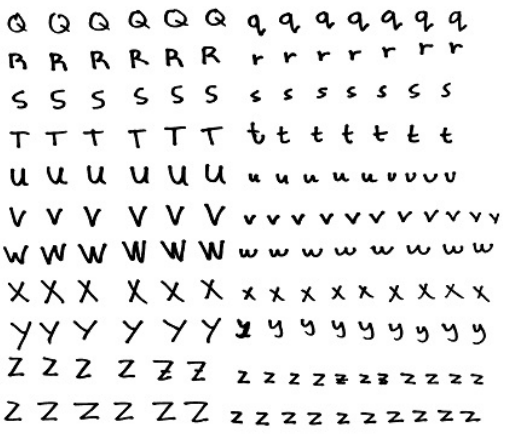
3. **Generating the training Image and creating a .box file.**

Training image is the image that contains the characters you want Tesseract to learn (see Figure 12). The image format can be anything that Tesseract accepts, however the recommended format is TIFF because it can contain multiple image in one image file which can minimize the number of image files in our directory. The Training image should be readable by Tesseract and should contain one type of handwriting per image file if possible, (which is also the reason why TIFF is recommended) for more accurate results.

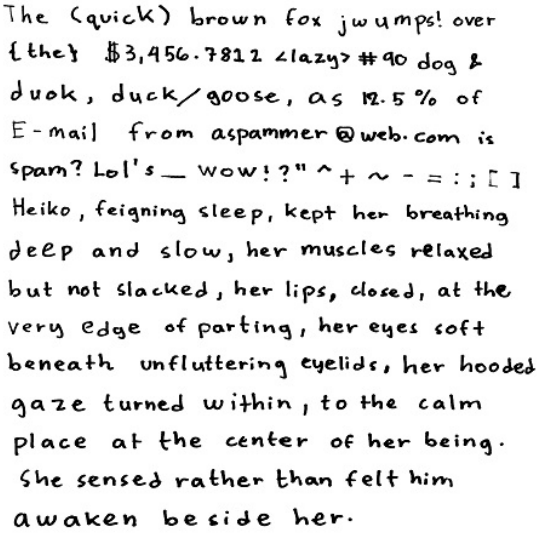
Figure 12: Training Images Sample

(a) Page 1

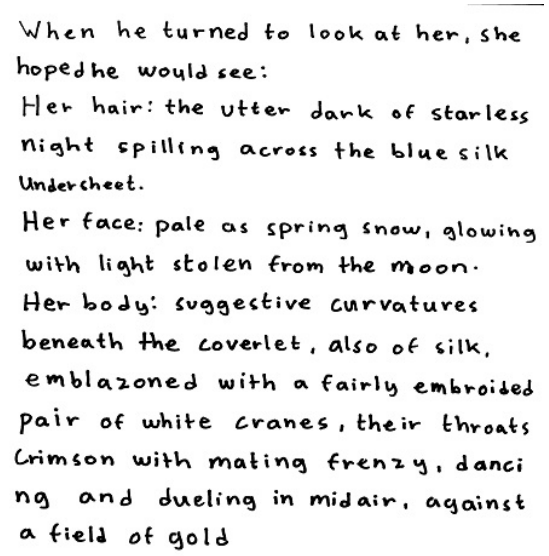
(b) Page 2



(c) Page 3



(d) Page 4



The **box** file is a text file which has a .box extension. It contains the characters in the training image, in order, one per line, with the coordinates of the bounding box around the image. This is a sample content of a box file (First letter 'A' in page 1 of training images in Figure 12):

A 283 3208 372 3329 0

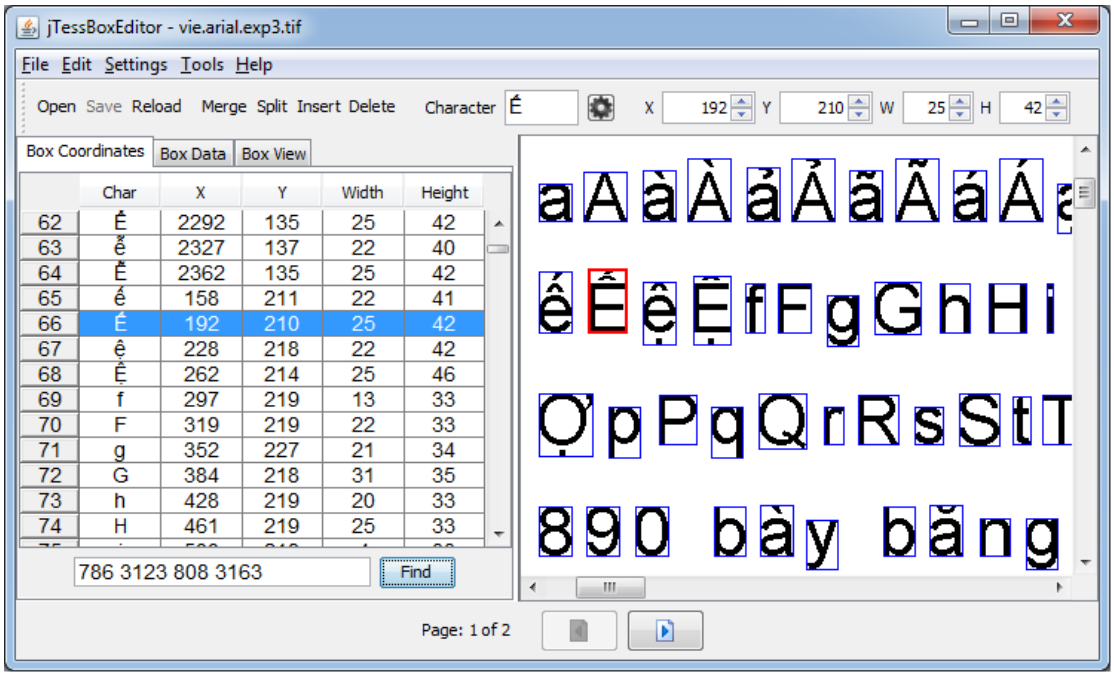
The format is:

Character X Y Width Height Tiff−Page

See Appendix A. for the whole content of the box file for the sample training image in Figure 12.

As you can see from the example, creating a box file for numerous training images is very tedious and time consuming. For this very reason, there exists a free and open-source box editor and training tool called jTessBoxEditor which is created by vietOCR [32].

Figure 13: Screenshot of jTessBoxEditor [32].



The training images' filename format should be 'fontname.imagename.exp0.extension' and the box file's format should have the same name as its training image

counterpart except for the extension part. So in our case, if the training image's file name is 'engh.aaronfont.exp0.tiff', its box file counterpart should have a file name 'engh.aaronfont.exp0.box'.

Note that we can create any number of image-box file matching pairs depending on how many fonts you want to train Tesseract.

To make things a little easier, you can use Tesseract to automatically create a box file for the training image. It will automatically detect the bounding boxes of the characters for you, although if Tesseract is not yet trained to that language, it is guaranteed to label all, if not most, of the characters wrong. This is where jTessBoxEditor will come in handy. You will manually relabel and correct the bounding box of the produced box file of Tesseract.

The command for Tesseract to make a box file is: (Assuming that you open CMD on the same directory as the binary files, and the images and box files are also in the same directory)

'tesseract engh.aaronfont.exp0.tif engh.aaronfont.exp0 batch.nochop makebox'

This will output an 'engh.aaronfont.exp0.box' file.

4. **Generating (.tr) training file**

   For each of your training image, boxfile pairs, run Tesseract in training mode:

   'tesseract engh.aaronfont.exp0.tif engh.aaronfont.exp0 box.train'

   This will output an 'engh.aaronfont.exp0.tr' file which contains the different features for each character being trained.

5. **Computing the Character Set**

   Tesseract needs to know the set of possible characters it can output or in other words, its unicharset. The unicharset_extractor program is used on the box files to generate the unicharset data file:

   'unicharset_extractor engh.aaronfont.exp0.box'

   This will output a single unicharset data file. Unicharset data file contains character properties isalpha, isdigit, isupper, islower, ispunctuation. Each line of this file corresponds to one character. The character in UTF-8 is followed by a hexadecimal number representing a binary mask that encodes the properties. Each bit corresponds to a property. If the bit is set to 1, it means that the property is true. The bit ordering is (from least significant bit to most significant bit): isalpha, islower, isupper, isdigit.

6. **Creating a font_properties file**

   The purpose of this file is to provide font style information that will appear in the output when the font is recognized. For windows, it is recommended to save the font_properties text file as front_properties.sh.

   Each line of the font_properties file is formatted as follows:

   <fontname><italic><bold><fixed><serif><fraktur>

   In our case, since we are training for handwritten texts and do not need any font properties, our font_properties file contains:

   aaronfont 0 0 0 0 0

7. **Clustering**

   After all the character features have been extracted, we need to cluster them to create prototypes. This can be done by using shapeclustering, mftraining, and cntraining programs.

   Although according to Google, shapeclustering should only be used on

Indic Languages because this program is still experimental[5]. For this reason, shapeclustering program will not be used in training handwritten texts.

To perform mftraining:

'mftraining -F font_properties.sh -U unicharset -O engh.unicharset engh.aaronfont.exp0.tr'

where 'unicharset' is the unicharset file we extracted using unicharset_extractor, 'font_properties.sh' file is the file created earlier, and engh.unicharset is the output modified unicharset.

This will output three additional files which are:

(a) **shapetable** file, which contains the master shape table.

(b) **inttemp** file which contains the shape prototypes.

(c) **pffmtable** file which contains the number of expected features for each character.

Next is to perform cntraining:

'cntraining engh.aaronfont.exp0.tr'

This will output a **normproto** file which contains the character normalization sensitivity prototypes.

8. **Optional Files**

A Directed Acyclic Word Graph (DAWG), is a data structure that permits extremely fast word searches. Each nodes of the graph represents a letter, and you can traverse from the node to two other nodes, until the letter matches the one you are searching for. Note that since it's Acyclic, meaning that there are no cycles, You can move from node A to Node B, but not the other way around. [33]

(a) **config file (engh.config)** which is a text file that contains different

configurations that Tesseract will use when using this trained data.

(b) **word-dawg** which is a DAWG made from dictionary words from the language. It is produced by creating a list of words (dictionary) in a text file which are separated by new line and using wordlist2dawg program to convert the text file to a DAWG file.

wordlist2dawg wordlist.txt

(c) **punc-dawg** which is also a DAWG file similar to word-dawg. But instead of listing words, we list pattern of punctuations. The "word" part is replaced by a single space i.e. ! ? indicates that words are frequently in between exclamation point and question mark. It is also created using the wordlist2dawg program.

wordlist2dawg punctuations.txt

(d) **unicharambigs** which is not a DAWG file. It is a text file which describes possible ambiguities between characters or sets of characters. It is manually generated and has a format in (Tesseract version 3.03):

<characters for match source><tab><characters for match target><tab><type indicator>

i.e

v2

" " 1 ->for all ' ' encountered, change it to "

m rn 0 ->if m is encountered, consider checking rn.

iii m 0 ->if iii is encountered, consider checking m.

(e) **number-dawg**'s format is the same as punc-dawg, but instead of using symbols, numbers are used.

wordlist2dawg numbers.txt

(f) **freq-dawg**'s format is the same as word-dawg, but instead of listing the whole dictionary, we only list the most frequent words in the language.

(g) **bigram-dawg** is a DAWG of word bigrams – a pair of consecutive written units such as letters, syllables, or words – which is also created

from text file using wordlist2dawg, where the words are separated by a space and each digit is replaced by a ?.

wordlist2dawg bigrams.txt

(h) **unambig-dawg** is still currently undefined in Tesseract's page.

(i) **params-model** is still currently undefined in Tesseract's page.

(j) **fixed-length-dawgs** is deprecated.

(k) **cube-unicharset** is deprecated.

(l) **cube-word-dawg** is deprecated.

9. **Combining Components to Produce Trained Data**

This is the last step to produce a .traineddata for our language. To combine all the components, we first need to add a prefix 'lang.' (in our case 'engh.') in the beginning of all the generated components from the previous steps. Then we will use the combine_tessdata program to combine all the files with the prefix 'engh.' to produce the final training data.

This will produce a .traineddata file (in our case 'engh.traineddata'), which is a data file for Tesseract. In order to use this data file, we need to copy this file to the 'tessdata' folder of Tesseract. After copying, Tesseract can now detect and use your language for recognizing texts.

tesseract -l engh sampleimage.tif textoutput

D..4 Criteria for Finishing Tesseract's Training

The research done by Rakshit S. and Basu, S. [3] only used a three user model, which means that they only collected handwriting samples from three individuals. There are two data sets for each individual which are the Isolated Characters and Free-Flow Texts. The distribution of training test and samples can be seen in Table 2:

Isolated characters are from test samples that has only letters from a-z in each and every row of the paper. They should not form a word but letters only. Free

Table 2: Distribution of training and test samples of different users [3]

	Isolated Characters	Free flow Text		Total Characters
		Characters	Words	
Train set for User-1	1185	659	137	1844
Test set for User-1	442	691	120	1133
Train set for User-2	1006	529	130	1535
Test set for user-2	468	718	128	1186
Training set for user-3	588	525	169	1113
Test Set for user-3	260	944	161	1204

flow texts are test samples that contain words and sentences i.e. handwritten journal article, book, etc.

Table 3: Result of the Rakshit and Basu's experiment. [3]

	Characters of Dataset-1	Characters of Dataset-2	Overall performance
Successful Recognition	95.42	83.2	87.92
Misclassification	4.1	16.19	11.52
Segmentation Failure	0.48	0.61	0.56
Rejection	6.10	4.34	5.03

	Characters of Dataset-1	Characters of Dataset-2	Overall performance
Successful Recognition	91.62	76.45	81.53
Misclassification	8.38	18.31	15.00
Segmentation Failure	0.00	5.24	3.47
Rejection	26.07	4.18	12.82

	Characters of Dataset-1	Characters of Dataset-2	Overall performance
Successful Recognition	92.34	58.35	65.71
Misclassification	6.81	6.24	6.36
Segmentation Failure	0.85	35.41	27.93
Rejection	9.61	9.96	9.88

OCR's recognition accuracy is computed using the formula:

$$RecognitionAccuracy = \frac{C_t - C_e}{C_t} * 100$$

Where C_t = the total number of characters and C_e = the number of misclassified character.

As shown in the tables (Table 3), the overall character-level recognition accuracy of their trained Tesseract OCR is around 78.4%. The overall character misclassification rate is observed around 11%. Segmentation failures in the document pages account for around 10.6% error cases. The trained OCR also rejects around 9.24% characters in the test data test due to the presence of multi-skewed

handwritten texts and cursive words.

These results can still improve by incorporating more training samples for each users and implementing different training technique/s to at least reach 80% accuracy on handwritten texts.

E. Text-to-Speech

The focus of this project is on Optical Character Recognition training, so the installed Text-to-Speech (TTS) system will be used and no further improvement of the TTS system will be done. Users are allowed to download any TTS Engines that they prefer in Google Play Store.

IV. Design and Implementation

A. Diagrams

A..1 Block Diagram of Tess2Speech Mobile Application

Tess2Speech combines Google's Tesseract [5], which converts the input images into text, and then convert it to speech. The block diagram is shown in Figure 14

Figure 14: Block Diagram of Tess2Speech

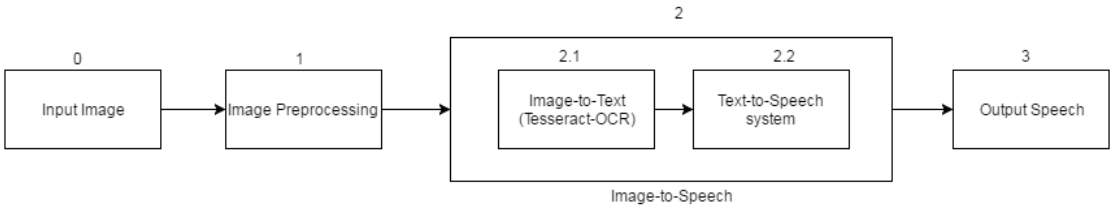
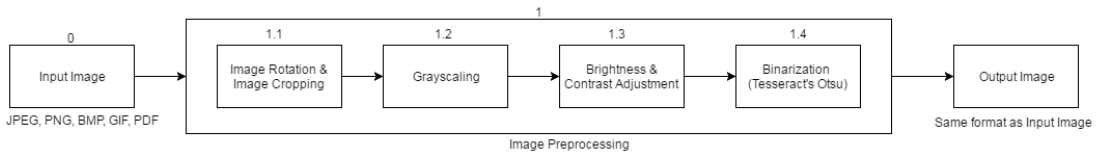


Figure 15: Sub-Explosion of Image Preprocessing in Fig. 14



In figure 15, you will notice that there is an Image Rotation and Image Cropping. Image rotation and image cropping are functionalities of Tess2Speech, which means that they are manual and the users are the ones that will indicate on how will they rotate or crop an image. Image rotation serves as an alternative for image de-skewing, and image cropping can eliminate the border problems (discussed in Scope and Limitations #7) for some images.

Figure 16: Sub-Explosion of Tesseract in Fig. 14

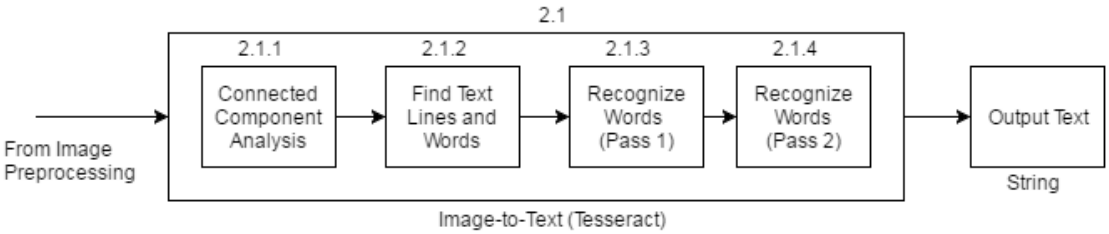
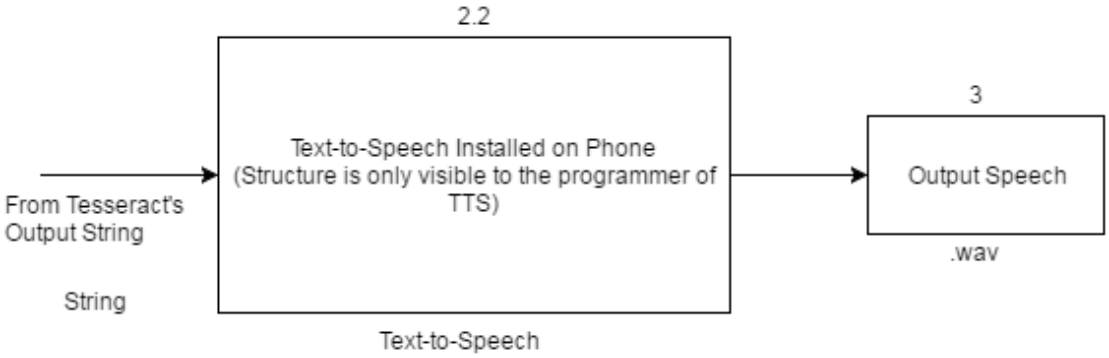


Figure 17: Sub-Explosion of Text-to-Speech in Fig. 14

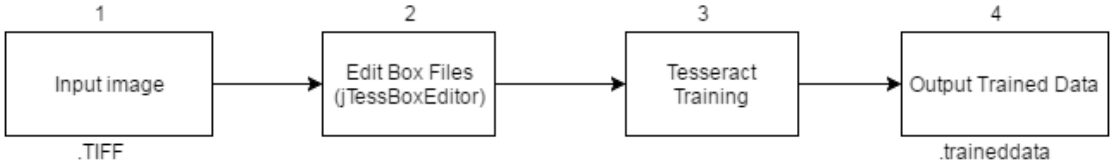


A..2 Block Diagram of Tess2Speech Trainer Desktop Application

Tess2Speech Trainer is a user-friendy desktop application that allows users to train or re-train Tess2Speech to recognize the desired font or handwriting. It automates all of the required steps in training Tesseract except for the box file editing. Its Block diagram is shown in figure 18. Tess2Speech Trainer needs an input image (see figure 12 for sample training image) which is in TIFF format. After the user inputted the image, the application will automatically produce a box file. A Box Editor, which is a modified jTessBoxEditor [32], will then pop-up to allow users to edit their box files. After editing the box files, the training will proceed automatically. The application will produce a .traineddata which can be used by Tess2Speech mobile application.

Since it needs to be user-friendly, Tesseract's training is abstracted from the user but its sub-explosion block diagram is shown in figure 19.

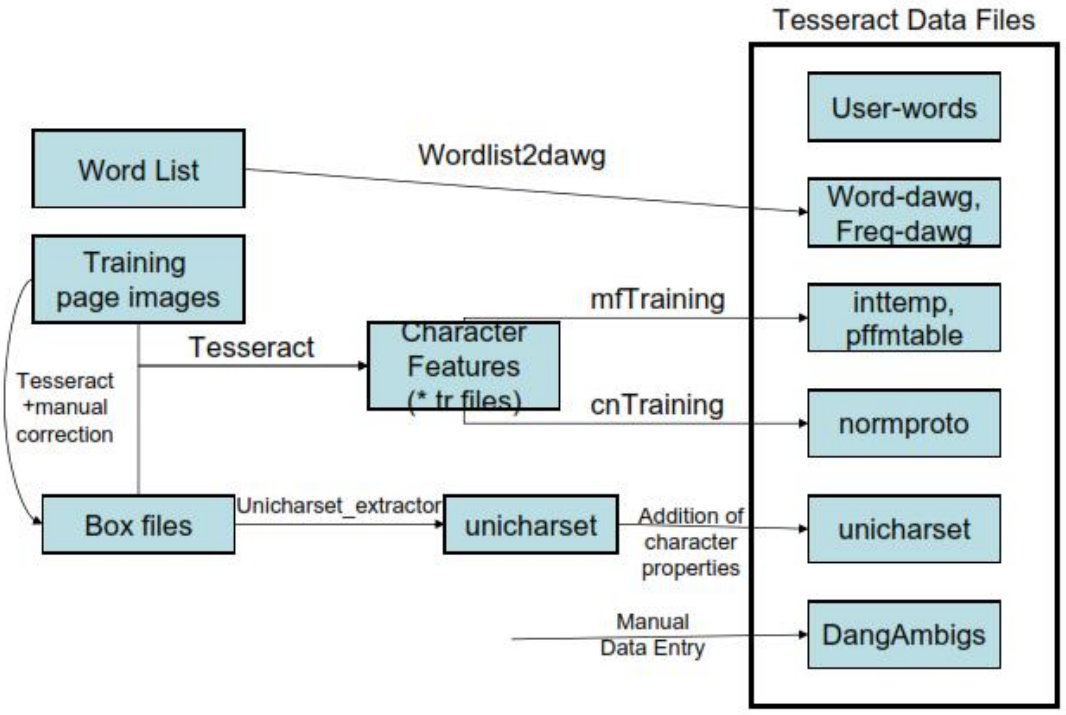
Figure 18: Block Diagram of Tesseract Trainer



A..3 Block Diagram of Tesseract's Training

All of these components are discussed in Chapter 3 D..3.

Figure 19: Block Diagram of Tesseract's Training[18]



A..4 Flowchart of Tess2Speech Mobile Application

Input file refers to the image/PDF/Ebook that contains the texts the user wants to convert into an editable text and speech. Input files can be PNG, JPEG, BMP, GIF, PDF, and EPUB for Ebooks.

Tess2Speech gives the users choices on what input they want to process. The users can choose to browse their phone for an existing input file, use their camera to capture an input image, or use the canvas to write on the screen.

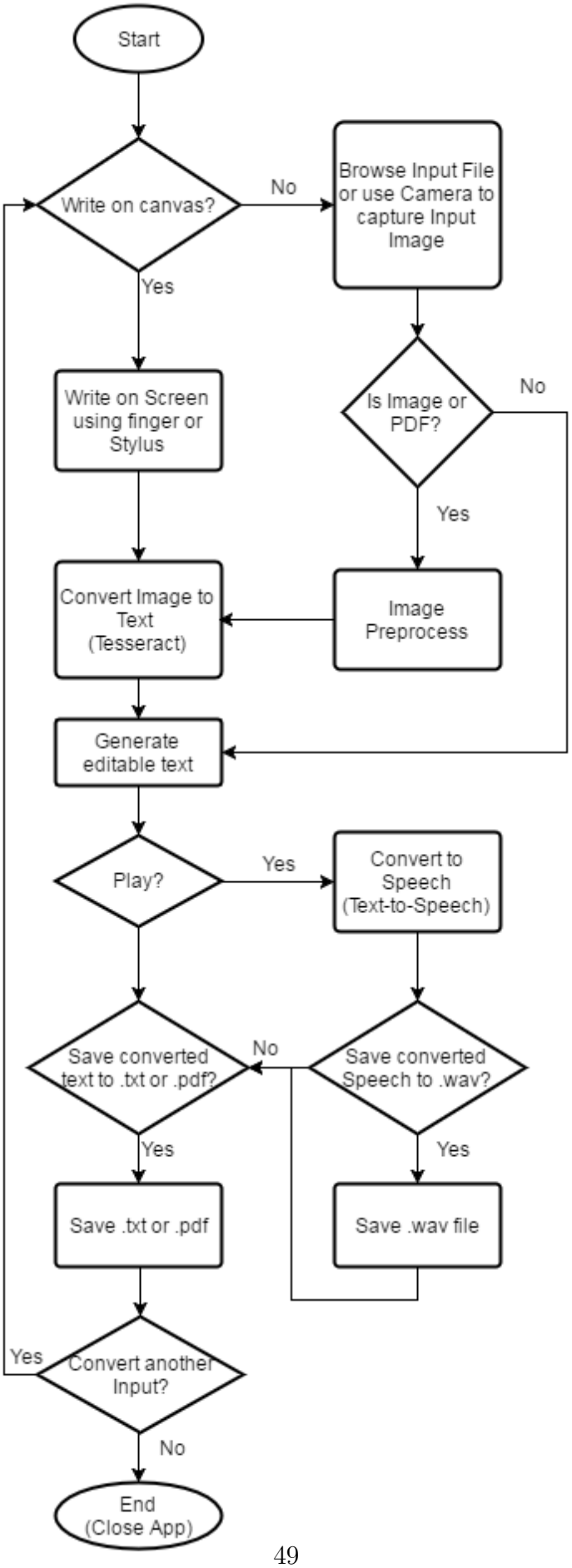
After the input has been created or selected, the users can manually rotate or crop the input image before passing it to other preprocessing techniques, specifically Grayscaling, Adjusting brightness and contrast, and Otsu's Binarization.

After performing series of image preprocessing, the processed input will be fed to Tesseract OCR and convert it to text that is displayed in an editable textbox.

The users can edit, modify, and save the generated editable text into .txt or .pdf. Users are also notified words that are not in the dictionary by using Android's built-in spell-checker.

The generated editable texts can now be converted into speech by pressing play button. The user can replay the speech by pressing the play button again. The user also has a choice to save the converted speech in to an audio file with .wav format. The flow chart of this application is shown in Figure 20.

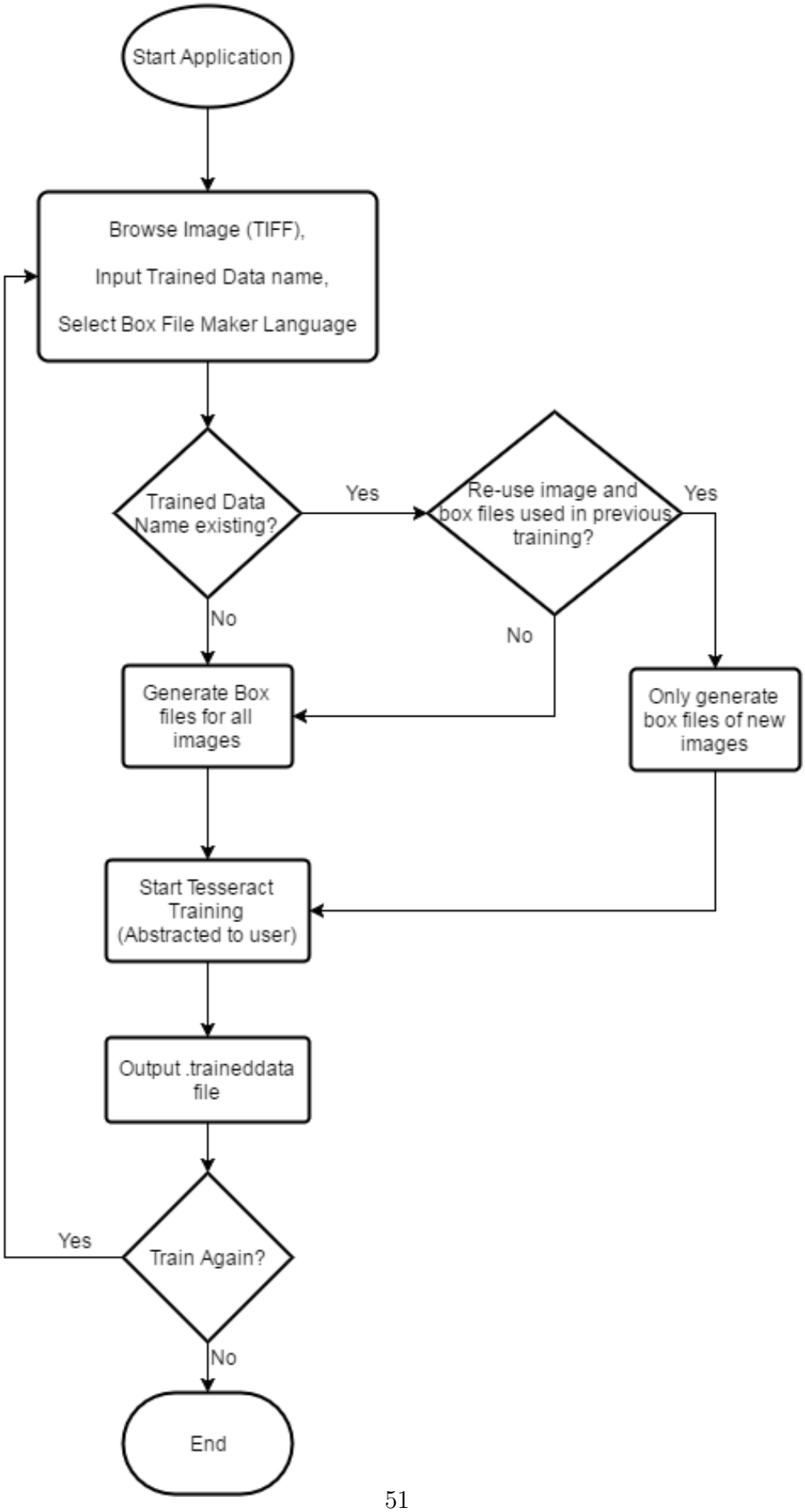
Figure 20: Flowchart of Tess2Speech Mobile Application



A..5 Flowchart of Tess2Speech Trainer

The user browses any number of TIFF files, and then specify the name of the output trained data. If the name of the trained data already exists, the old training images that is used to train that trained data can be re-used. The user can specify whether to overwrite the old box files of the old images or retain it to avoid re-editing the same box file for the old images. jTessBoxEditor [32] will pop-up and allows the user edit the box files of the images. After the user has finished editing the box files, Tess2Speech trainer will automatically train Tesseract and will automatically output a .traineddata file with the name that the user specified. This .traineddata file can now be used in Tess2Speech mobile application by copying the .traineddata to the Tessdata folder of Tess2Speech (see section B..5 to find Tess2Speech tessdata folder). The flow chart of Tess2Speech trainer can be seen in figure 21.

Figure 21: Flowchart of Tess2Speech Trainer



A..6 Use-Case Diagram

Tess2Speech features two types of users: The general users and the application developers. The general users are the application users that uses the main functionalities of Tess2Speech, while the application developers are those that perform training on Tesseract and updates on the application.

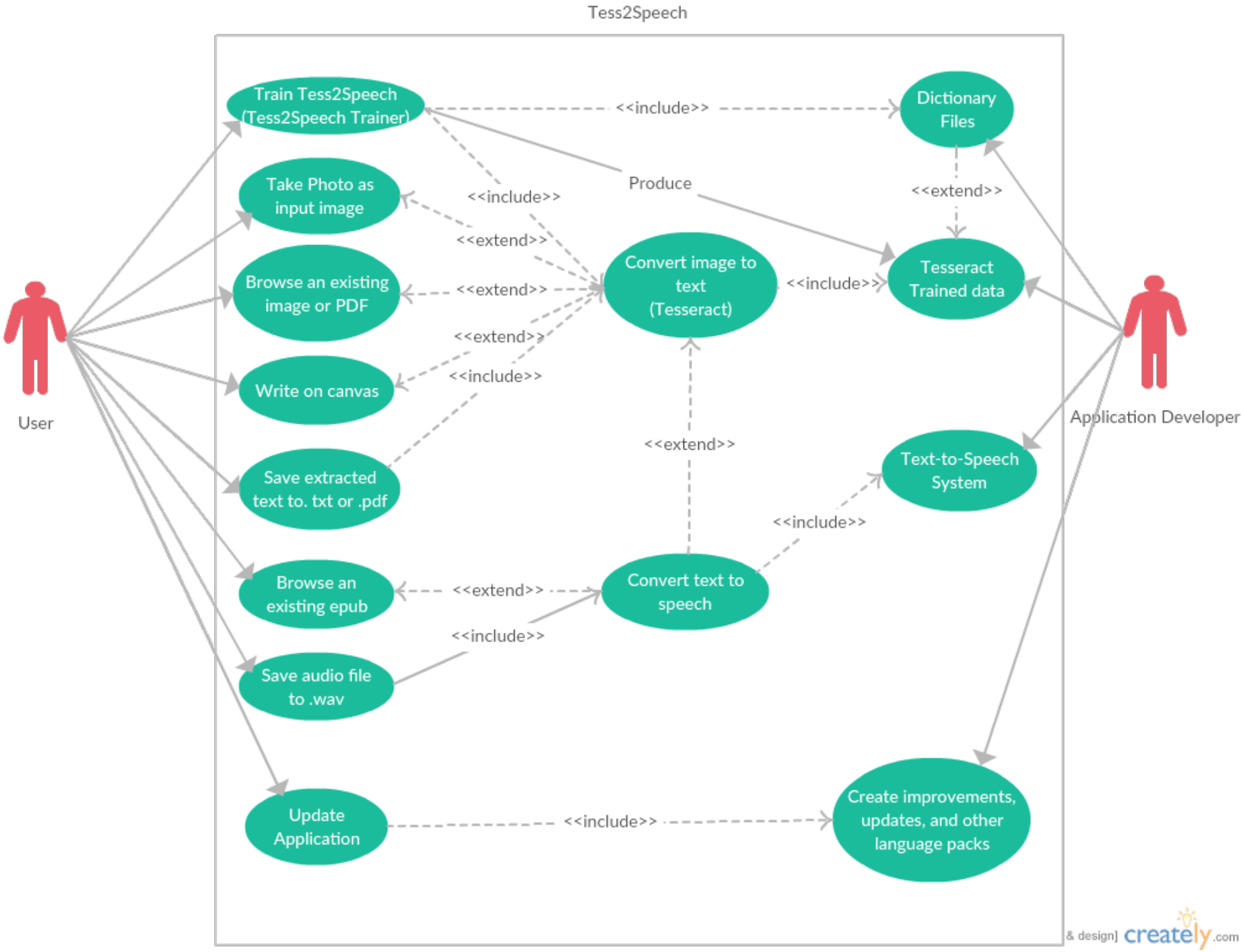
Tess2Speech can use any language packs that the application developers created. Language packs are trained data that were created by application developers. Example of a language pack is for handwritten English texts which I created (engh.traineddata). Since Tesseract is still trainable even after the deployment of this application, application developers can still create handwritten language packs for Chinese, Korean, Arabic etc. language packs in the future and add those to Tess2Speech.

It is also important to note that since Tess2Speech can use any .traineddata files on the internet, users can download them and copy it to Tess2Speech' tessdata folder. This will make Tess2Speech recognize that language (can be anything that is currently existing i.e. Computer typed texts for Chinese, Korean, Arabic, etc.).

These added language packs can be implemented through future updates or by manually downloading the file from a host server.

The use-case diagram of Tess2Speech is can be seen on Figure 22.

Figure 22: Use-Case diagram of Tess2Speech



B. System Architecture

B..1 Android OS

Android is a mobile operating system (OS) which was developed and currently being developed by Google [6]. Its operating system is based on the Linux kernel and is primarily designed for smart phones and tablets. Android applications are developed using Java Programming Language (see B..2) and Android Software Development Kit (SDK). Android OS will be used as the platform for this application because aside from the fact that Tesseract is now being developed by google [5]; Tesseract can now be used as a library for Android by using Tess-Two (see B..4).

B..2 Java Programming Language

Java Programming Language is a general-purpose, concurrent, class-based, object-oriented language. Its syntax is somewhat related to C and C++ but is organized differently. Java language is strongly typed which means that it has a strict set of rules when it comes to programming syntax. Java programming language also distinguishes run-time errors and compile-time errors. It is also a relatively high-level programming language since it includes automatic storage management and garbage collectors which automatically performs deallocation unlike in C and C++ which uses explicit deallocation [34]. Java needs Java Virtual Machine (JVM) installed on the computer to work. Java Programming Language is also used in Android application development alongside other Android SDKs. Java version 1.8.0_66 is used in the development of both Tess2Speech mobile application and Tess2Speech Trainer desktop application.

B..3 Android Studio

Android Studio is the official Integrated Development Environment (IDE) for Android Application Development. Android Studio version 1.5.1 is used in the development of Tess2Speech mobile application.

B..4 Tess-Two

Since Tesseract OCR is compiled using C++, Tess-Two is a free open-source Android library project that provides a Java API for accessing natively-compiled Tesseract and Leptonica APIs [35].

B..5 Tess2Speech' Tessdata folder

Data files (.traineddata) for Tesseract OCR should be placed in Tesseract's 'tessdata' folder in order for it to detect the trained data. It is the same for Tess2Speech. In order for a trained data be used in Tess2Speech, it should be located in Tess2Speech' 'tessdata' folder. It is usually located on
'Phone Storage/Android/Data/anteraaron.tess2speech/files' directory.

B..6 Text-to-Speech System

Any Text-to-Speech system installed can be used on Tess2Speech. This means that the performance of Tess2Speech' TTS is based on the TTS being used on the phone. Text-to-Speech systems can be downloaded on Google Play Store. TTS settings can be changed on Android's phone settings.

B..7 Tess2Speech' Canvas

As we stated on Background of the Study, stylus-based applications are mostly proprietary and expensive. As a solution to that, I developed a pseudo stylus-based application. In the canvas part of Tess2Speech, I created a Canvas object on the phone's screen, which is similar to how MS Paint works, and then convert the content of that canvas to image every time I want to convert that handwriting to text or speech. The performance of this canvas is dependent on the hardware of the phone (touchscreen capability).

B..8 Tess2Speech' Image Viewer

The reason why Tess2Speech is only accepting input images of format PNG, JPEG, GIF, and BMP is because Android's ImageView Object accepts these same image formats. In order to support pinch-zoom, double-tap zoom, and pan functionalities in the ImageView, I used an open-source Android View class called GestureImageView by jasonpolites [36].

B..9 Tess2Speech' Camera

Almost all smart phones have a camera, and it is accessible by all of the applications installed on the phone. The problem is that not all camera resolutions are equal. It varies from phone-to-phone. Since Tess2Speech relies mostly on the quality of images, an image taken from a high-resolution camera will most likely yield a better accuracy.

B..10 Image Cropping and Rotation

It is stated in Scope and Limitations no.7 that the border of the paper should not be seen as it adds noise to the image and may be interpreted as a character such as 'l' (small L) or '1' (one). As a solution, I implemented a choice for the user to crop the image and only select which part of the image is relevant. I used an open-source image cropper library by ArthurHub [37], and edited it a bit to fit my personal preferences.

Since Image De-Skewing is hard to implement without the use of large libraries, I also provided an alternative solution by allowing the user to manually rotate the image allowing them to correctly align the texts on the image.

B..11 Tess2Speech' PDF Viewer

Android supports viewing of PDF starting from API level 21. Since the minimum API level of Tess2Speech will be API level 16, I used an open-source PDF Viewer to support lower API levels. Android-pdfview is a library created by Joan Zapata [38]

which provides fast PDFView component for Android, with animations, gestures, and Zoom. It is based on VuDroid [39], which is a PDF decoder made by google.

B..12 PDF to Image and Vice-Versa

It is stated in B..11 that the PDF Viewer uses VuDroid [39]. Since VuDroid is a PDF Decoder, it should have a functionality of converting a PDF to an Image. After studying the source code of VuDroid, I found a method on how to convert PDF to Image and vice-versa. To maximize the use of the library, I will incorporate these functionalities in Tess2Speech.

B..13 Tess2Speech' Ebook Viewer

Ebooks with .epub extensions are essentialy a zip of html files. Since Android does not have an innate capability to display Ebook, I used an open-source library called EPubLib by Paul Siegman [40]. EPubLib is a Java Library for reading, writing, and manipulating epub files. The problem is that the formatting of epub is not acquired when using EPubLib. By creating a workaround, I eventually displayed the proper formatting of the Ebook.

B..14 Tess2Speech's Built-in File Picker

I also incorporated a built-in File Picker for Tess2Speech in-case that the user does not have a compatible File Picker installed on their phone. The file picker I used is an open-source file picker library by Anders Kaloer [41].

B..15 Saving Files

Converted texts can be saved as a text file (.txt), or a PDF file (.pdf) by using VuDroid [39]. An SD Card is required in order to save Tess2Speech files. If an SD Card is not present, Tess2Speech will prompt the user and automatically close the application.

B..16 Settings

The Preference Screen of Tess2Speech contains different settings for Tess2Speech such as the option to whether turn-on Image Preprocessing, Automatic Image Resizing (for faster conversion), Tesseract language, help, and Licenses.

B..17 Tess2Speech Trainer

Tess2Speech Trainer is a user-friendly desktop application which allows the users to personalize and train Tess2Speech to recognize a new font or handwriting in just a click of a button. It produces a .traineddata file that can be used by Tess2Speech.

The Box Editor of Tess2Speech trainer is an edited jTessBoxEditor by VietOCR [32]. I added an indicator on which box files have already been visited to help users know if they missed editing some characters. I removed other functionalities of jTessBoxEditor that are not needed and integrated it to Tess2Speech Trainer's Graphical User Interface.

Tess2Speech Trainer is programmed using Java and is a .jar executable. Tess2Speech Trainer is also a stand-alone application, which means that there is no installation required for the application itself. The user just needs a Java installed to run the .jar executable inside the Tess2Speech Trainer. It is important to know that 'Tess2Speech Trainer.jar' file cannot be moved outside the folder or it will not run. Only 'create shortcut' method is allowed.

C. Technical Architecture

C..1 Development Environment

These applications were created in these following environments:

1. Windows 7 Ultimate SP1
   (a) Intel® Core™ i3 CPU
   (b) RAM: 4 GB (2.80GB usable)
   (c) 32-bit Operating System
2. Android Studio 1.5.1 for the Mobile Application
3. Eclipse IDE 4.5 (Mars) for the Desktop Applicaiton
4. Java version 1.8.0_60
   (a) Java™ SE Runtime Environment (build 1.8.0_66-b18)
   (b) Java HotSpot™ Client VM (build 25.66-b18, mixed mode)
5. Trained using Tesseract OCR [5].
   (a) Tesseract version 3.05.00dev
   (b) Leptonica version 1.73
   (c) jTessBoxEditor version 1.5 [32]
6. Tested using:
   (a) Lenovo S820 (Android 4.4.2/Kitkat/API level 19)
   (b) Genymotion Android Emulator (Google Nexus 5X/Android 6.0.0/Marshmallow/API level 23)
   (c) Android Studio Emulater (Google Nexus 5X/Android 6.0.0/Marshmallow/API level 23)

C..2 Minimum System Requirements

The minimum system requirements for Tess2Speech mobile application:

1. Android 4.1/Jelly Bean/API level 16
2. 120 MB free space (50+ MB for application itself and 50+ MB for built-in trained data and other files)
3. Wi-Fi 802.11 connectivity (optional for updates)
4. 2MP primary camera
5. Mounted 1GB SD card
6. 100 MB free RAM
7. An existing .traineddata in tessdata folder
8. A Text-to-Speech Engine is installed

The minimum requirements Tess2Speech Trainer desktop application are:

1. Windows 7
   (a) Intel® Core™ i3 CPU
   (b) RAM: 2GB
   (c) 32-bit Operating System (can also be run on 64-bit)
2. Java version 1.7
3. 110MB free space (excluding the size of generated trained data)

Tesseract and jTessBoxEditor does not need to be installed because it is already included in Tess2Speech trainer.

D. Training Methodology

In training Tesseract, there are no required format or template for the training datasets, although the performance of the resulting trained data will be dependent on the training datasets that you used in training.

In creating a trained data for handwritten texts, I considered using Basu and Rakshit's training method which they used in training Tesseract for handwritten lower case Roman script [3]. They used an n-user model where n is the number of persons they collected handwriting from. The training dataset templates for each user/individual is divided into two, namely isolated characters and free-flow text. See figure 40 for sample training dataset.

Isolated Characters are datasets that contains one repeated character per line, both the upper case and lower case of the character. The characters should not be very close to each other. Figure 40a and 40b are samples of Isolated Characters dataset.

Free-flow texts are datasets that contain a paragraph/s that has all of the desired characters you want to recognize. This is important because sometimes a person's handwriting in free-flow text is different from their handwriting in isolated characters. This is also to simulate the realistic frequency of each characters. Figure 40c and 23d is a sample of free-flow texts dataset.

Using these training datasets, training is then done either manually or by using Tess2Speech Trainer which automates the training process of Tesseract.

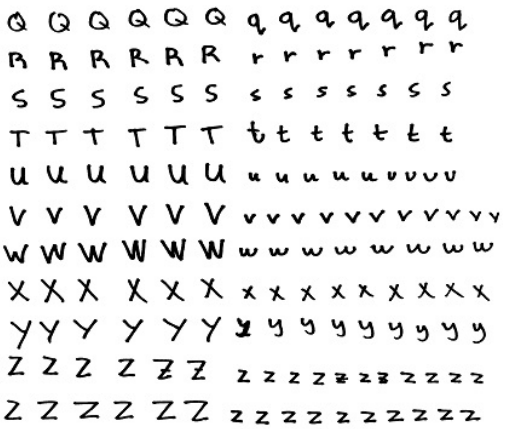
The test set for the trained data is also a free-flow text which is also written by at least on of the n individuals in the n-user model.

In Basu and Rakshit's experiment [3], they only used a three-user model but still managed to get at least 78% accuracy. By increasing the number of user and training sets, the accuracy can further increase.

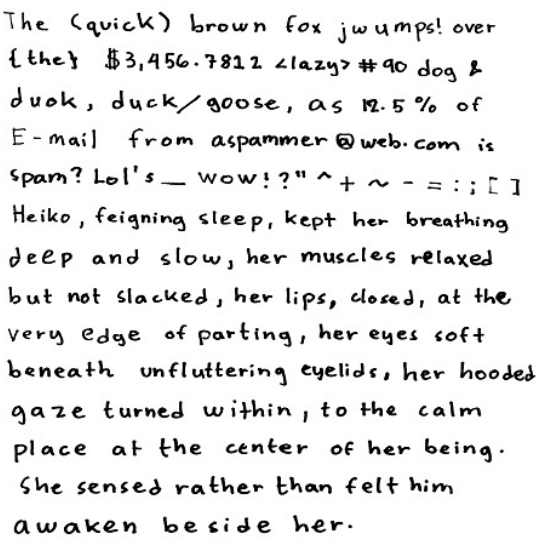
Figure 23: Training Images Sample

(a) Page 1 (Isolated Characters)

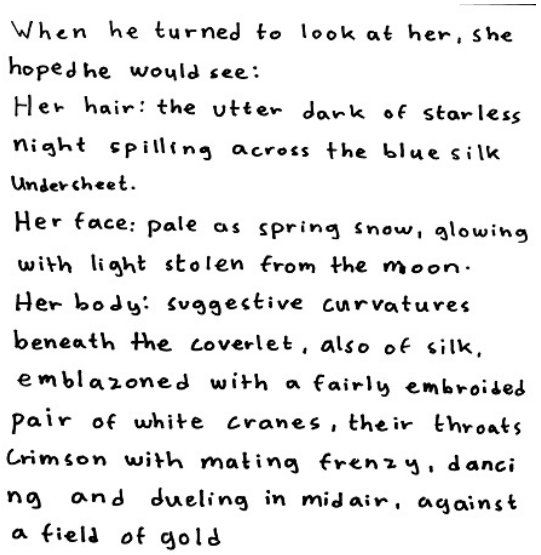
(b) Page 2 (Isolated Characters)



(c) Page 3 (Free-flow texts)



(d) Page 4 (Free-flow texts)



V. Results

A. Tess2Speech Mobile Application

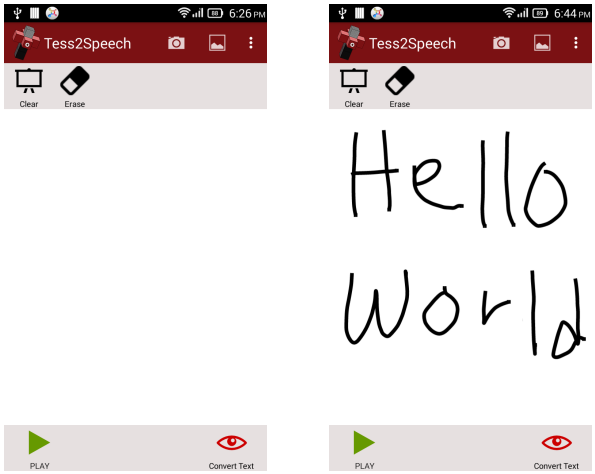
A..1 Splash Screen

Figure 24: Splash Screen of Tess2Speech

The splash screen shown in figure 24 is shown to the user while Tess2Speech is loading required resources.

A..2 Home Screen/Canvas

Figure 25: Home Screen/Canvas of Tess2Speech



The home screen of Tess2Speech is shown in figure 25. It is also the canvas functionality of Tess2Speech. The user can write anything on the Canvas and

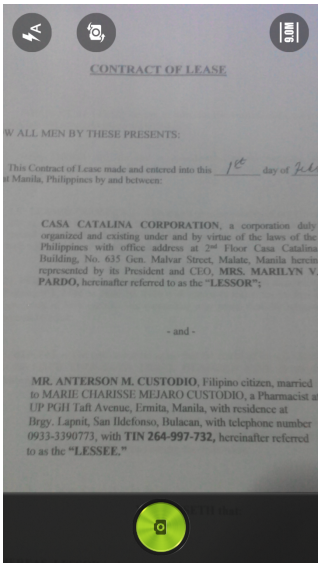
convert it to text or speech. If the 'Clear' button in the upper-left corner of the screen is pressed, it will clear the whole canvas into a white blank page. The button next to it, which is 'Erase' replaces the touch functionality of the user from writing to erasing. The button on the lower left corner of the screen is the 'Play' button which converts the words written on the screen into speech (see figure 30). The button on the lower right corner of the screen is the 'Convert Text' button which converts the words written on the screen into an editable text (see figure 29).

A..3 Camera

Figure 26: Camera Functionality of Tess2Speech

(a) Camera Button in Toolbar

(b) After Camera Button is Pressed



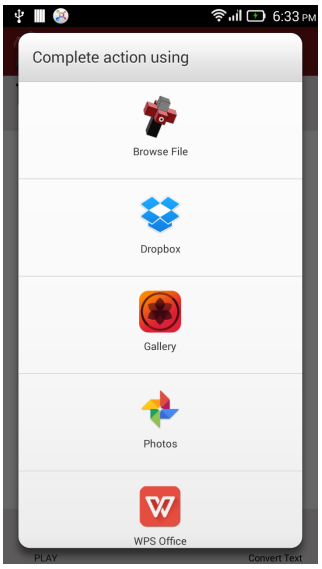
The camera button is the first button in Tess2Speech' toolbar (figure 26a). It allows the user to use the phone's camera to take pictures of the desired input image that will be displayed on the screen (see figure 28a). The camera interface (figure 27b) is based according to your phone's camera interface.

A..4 Browsing Image

Figure 27: Browse Image Functionality of Tess2Speech

(a) Browse Image Button in Toolbar

(b) File Chooser Dialog



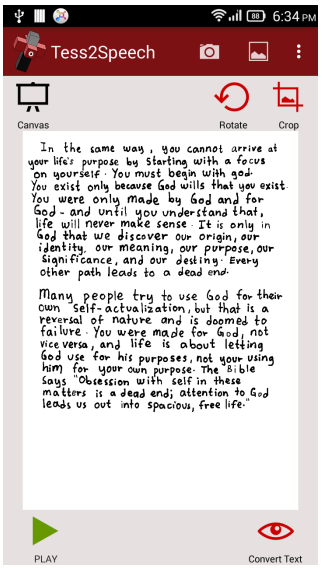
(c) Selecting File

'Browse Image' button is the button next to 'Camera' (figure 27a). After pressing 'Browse Image' button, a Chooser Dialog (figure 27b) will pop-up and gives the user choices of available applications installed on the phone that can open images. Figure 27b is the result of picking Gallery as a file picker. The user can now pick their desired input image and will be displayed on the phone's screen (see figure 28a).

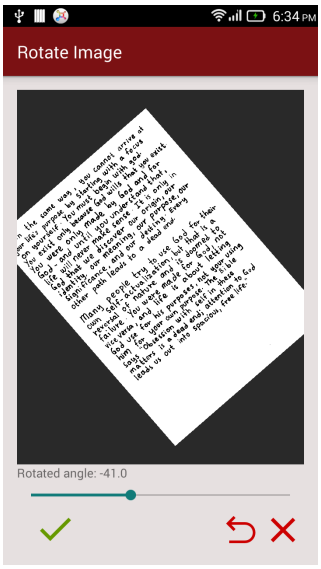
A..5 Displaying, Rotating, and Cropping Images

Figure 28: Displaying an image

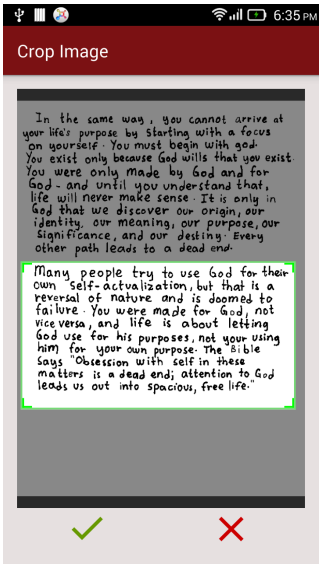
(a) Image being displayed



(b) Image Rotation



(c) Image Cropping



After browsing an image or taking a picture of an image, it will be displayed on the screen just like in figure 28a. New buttons in the upper-right part of the screen will be available if an image is being displayed. These are the 'Crop' and 'Rotate' Buttons. 'Rotate' button allows the user to rotate an image in any angle by adjusting the seek bar below the image (figure 28b). This serves as an alternative for de-skewing an image. The 'Crop' button allows the user to crop

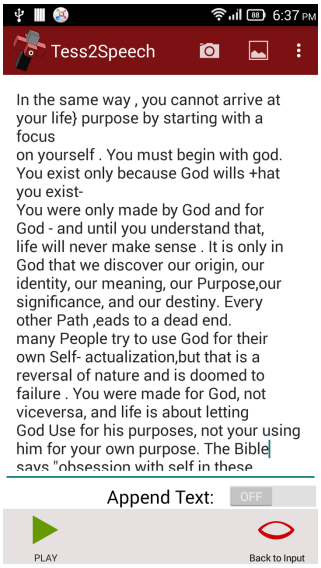
the unnecessary part of the image that Tess2Speech may read (figure 28c). The image displayed can now be converted to text or to speech (see figure 29).

A..6 Converting to Text

Figure 29: Converting to Text

(a) Loading Screen while Converting Text

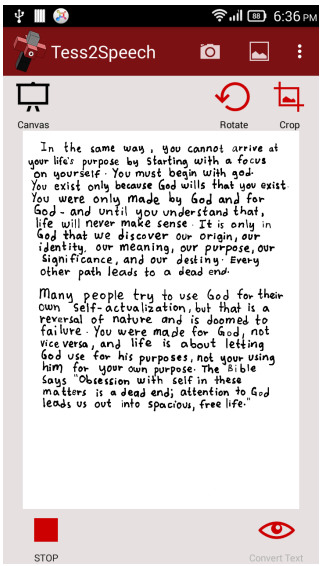
(b) Converted text of the image in 28a



The button in the lower-right corner of the screen is the 'Convert Text' button which converts the displayed image, text, canvas, pdf, or epub to text. After pressing the 'Convert Text' button, a loading screen (figure 29a) will appear while converting the displayed image to text. The converted text (displayed on figure 29b) is editable and can be copied. If you press the lower-right button again, it will bring you back to the displayed image. That's why after pressing 'Convert Text' button, its label change to 'Back to Input'. If the 'Append Text' switch in figure 29b is enabled, the current text displayed will not be erased and the next converted texts will be appended to it as long as it is enabled.

A..7 Converting to Speech

Figure 30: Converting the image in 28a to Speech



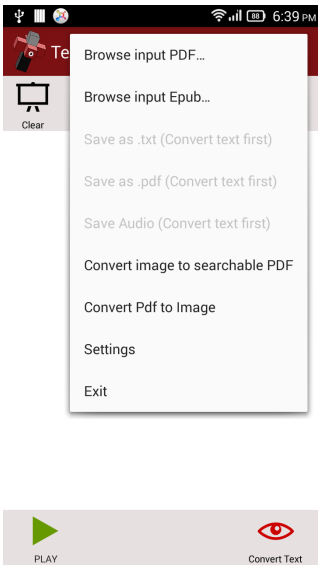
The button in the lower-left corner of the screen is the 'Convert to Speech' button which converts the displayed image (editable text, pdf, or epub depending on which is being displayed) into speech. If you will notice, the button in the lower-left corner in figure 30 became 'Stop' after pressing the 'Play' button. This is to allow the users to stop Tess2Speech from speaking. Note that converting an image to text will also produce a loading screen just like in figure 29a.

A..8 Browse an Input PDF

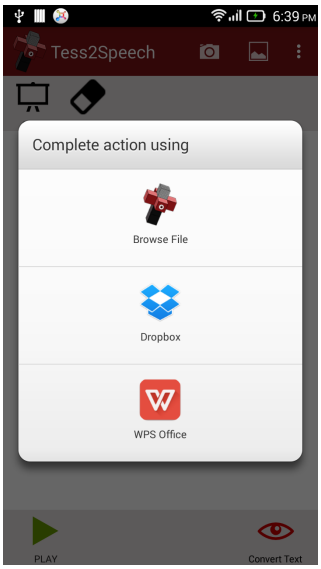
The button after the browse image in the tool bar is the overflow menu which contains other functionalities of Tess2Speech (31a). The first item in the overflow menu is the 'Browse input PDF...' which allows the user to use PDF as their input image. After pressing that option, a File Chooser Dialog will pop-up (figure 31b) which gives the user choices on which installed application to use in browsing PDF. In figure 31c, WPS office is used in browsing PDF. After browsing for the desired PDF, it will be displayed on the screen just like in figure 31d. When converting a PDF to text or speech, a pop-up dialog will appear (figure 31e) which gives the user choice on what pages of the PDF Tess2Speech convert.

Figure 31: Browsing a PDF

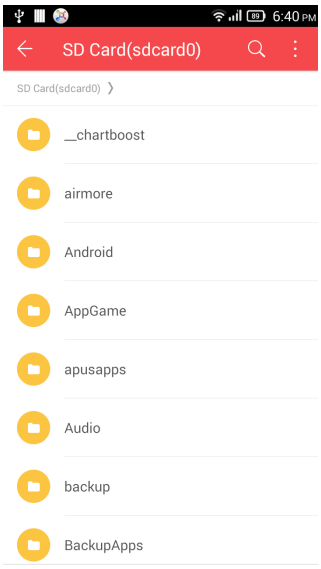
(a) Overflow Menu



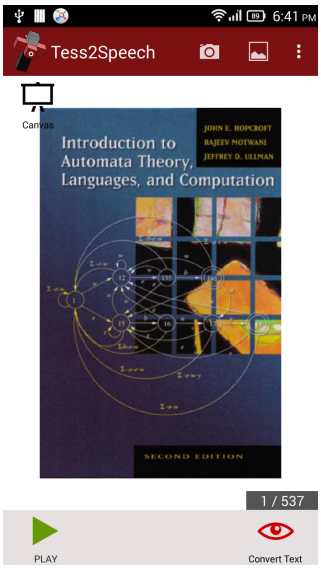
(b) File Chooser Dialog



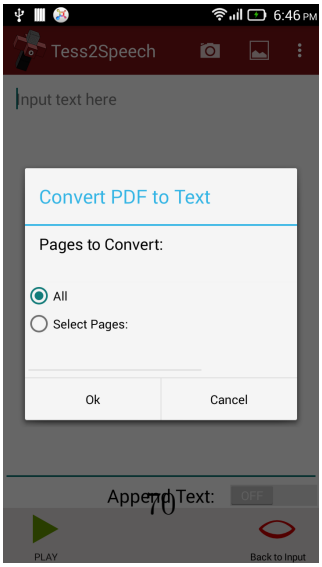
(c) Browsing PDF using WPS Office



(d) PDF Being Displayed



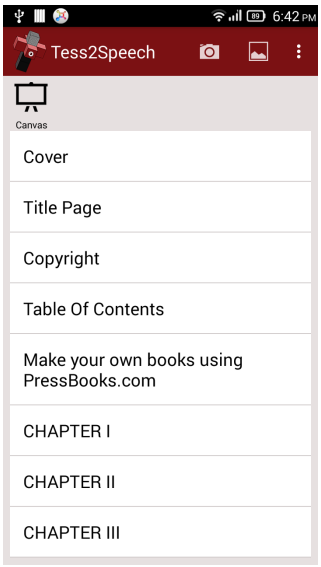
(e) Converting PDF to Text



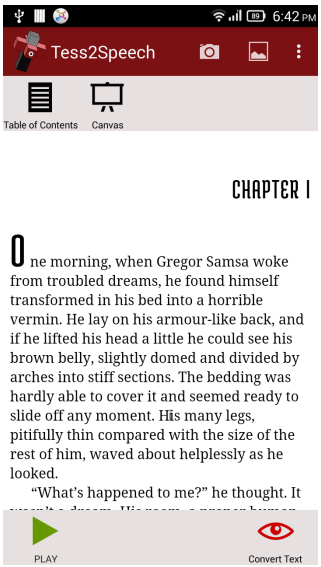
A..9 Browse an Input Epub

Figure 32: Displaying Epub

(a) Table of Contents



(b) Epub content

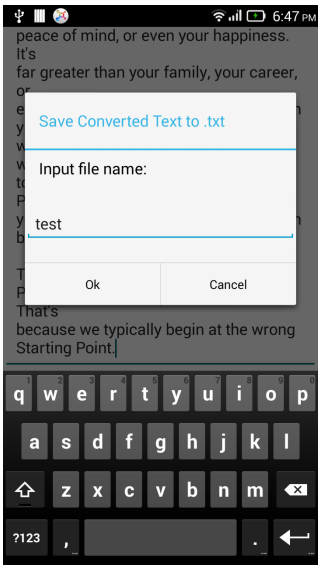


Browsing an epub is the same as browsing a PDF. First step is selecting 'Browse an input Epub...' from figure 31a. A pop-up menu will also open just like in figure 31b. Applications that can open epub will be displayed instead of displaying applications that can open PDF. After browsing for the desired epub, a table of contents for the selected epub will be displayed (figure 32a). Here, the user can choose which part or chapter of the epub they like to view. It will be displayed just like in figure 32b. The user can go back to the table of contents by pressing the upper-left button 'Table of Contents'. The user has a choice to convert the displayed chapter to text or speech by pressing the corresponding button.

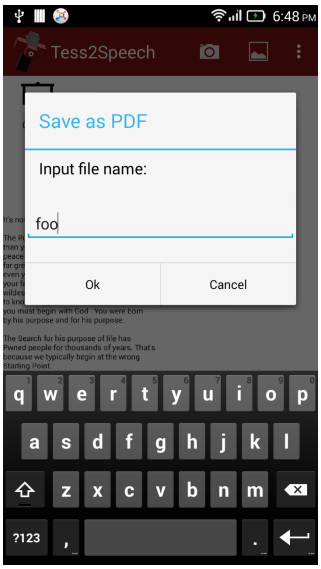
A..10 Saving

Figure 33: Saving Files to .txt, .pdf, or .wav

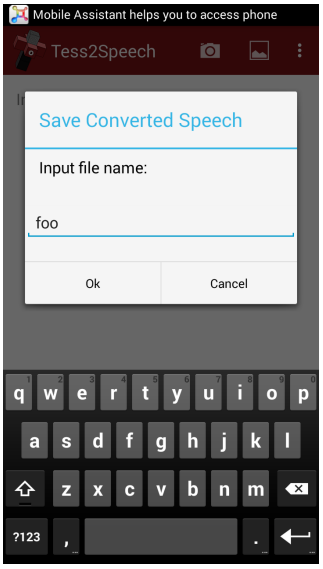
(a) Saving as .txt



(b) Saving as .pdf



(c) Saving as .wav

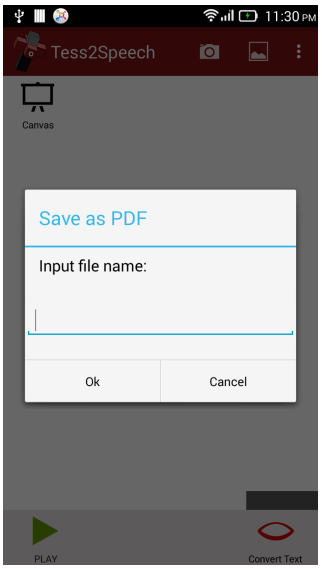


Saving can be done by selecting the save options in the overflow menu (figure 31a). The figures 33a, 33b, and 33c shows the pop-up dialog that asks for a file name for saving the .txt, .pdf, or .wav respectively.

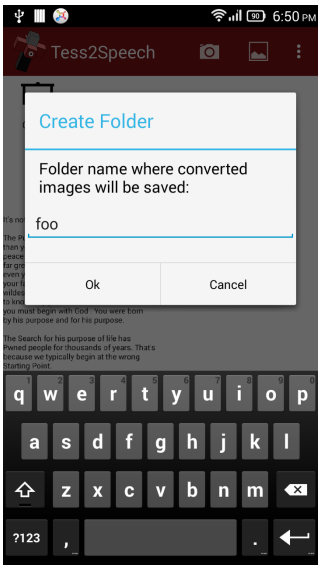
A..11 Converting Image to PDF and Vice-versa

Figure 34: Converting Image to PDF and Vice-versa

(a) Converting Image to PDf



(b) Converting PDF to Image



Conversion can be done by selecting the 'Convert Image to searchable PDF' or 'Convert PDF to Image' in the overflow menu in figure 31a. Figure 34a shows the pop-up dialog when the user chooses to convert an image to PDF. It asks for a file name for the resulting PDF. Figure 34b shows the pop-up dialog when the user chooses to convert a PDF to image. It asks for a folder name that the extracted images will be saved since one page from a PDF file corresponds to one image. If there are 50 pages in a PDF, 50 images will be produced. The user will also be prompted on what pages will be saved just like in figure 31e.

A..12 Settings

When 'Settings' is selected from the overflow menu in 31a, the preference screen in figure 35a will be shown. This preference screen contains different settings for Tess2Speech.

The first item is 'Image Pre-processing' which gives the user an option to turn-on Grayscaling and illumination fixing of images before converting to text or speech.

The next item is 'Re-scale image' which reduces the size of the image to convert to speed-up conversion time. It is not recommended for images with small texts sizes since Tess2Speech might not be able to recognize very small texts.

The next item is 'Tesseract Trained Data' which allows the user to select which trained data Tess2Speech will use. All trained data files in the 'tessdata' directory will be shown here (figure 35b The user can choose more than one trained data.

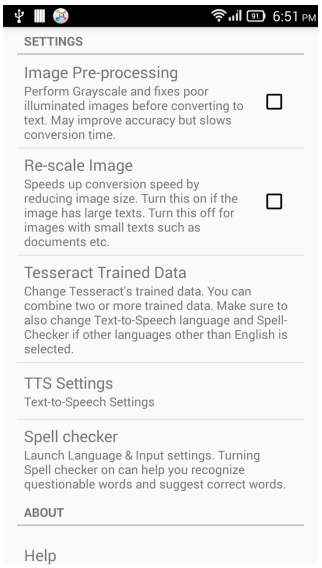
The next item is 'TTS Settings' in which the user will be redirected to TTS settings of the phone (figure 35c).

The next item is 'Spell-Checker Settings' in which the user will be redirected to 'Language and Input' settings of their phone (figure 35d). Here they can change the Spell-Checker settings of their phone.

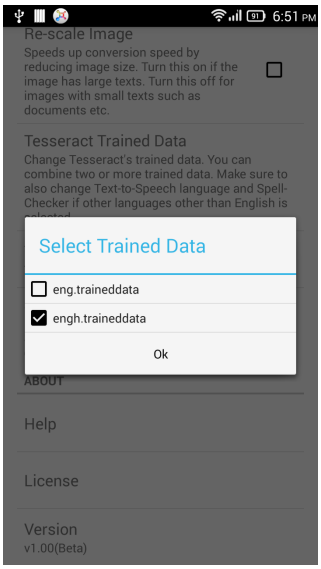
The last two items are Help and Licenses. Help displays different tips and helps the user become familiar with the application (figure 35e). Licenses shows different licenses and credits for the Open-Source libraries used in Tess2Speech (figure 35f).

Figure 35: Settings

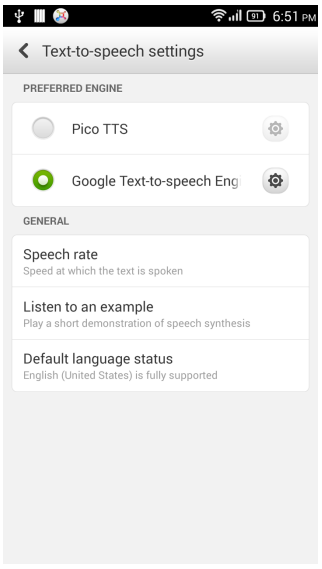
(a) Preference Screen



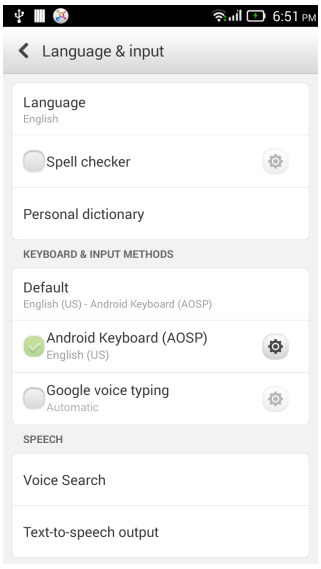
(b) Selecting Trained Data



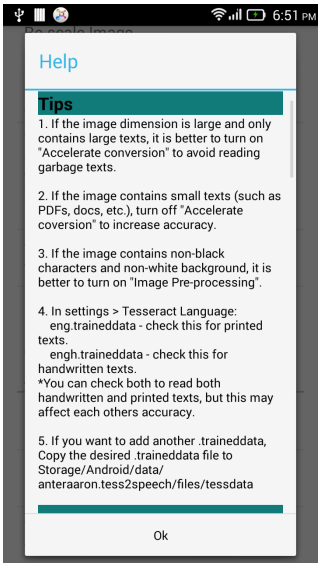
(c) TTS Settings



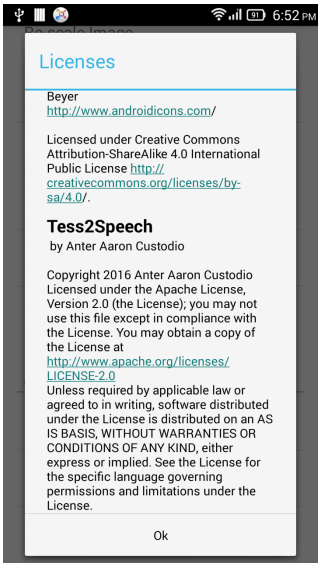
(d) Spell-Checker Settings



(e) View Help



(f) View Licenses



B. Tess2Speech Trainer

Since Tess2Speech Trainer needs to be user-friendly, its UI is very straight forward and contains minimal functionality.

B..1 Graphical User Interface

Figure 36: GUI of Tess2Speech Trainer

The GUI of Tess2Speech Trainer can be seen in figure 36. The 'Browse TIFF/s' button opens a File Chooser (see 37) that allows the user to browse for existing TIFF files in the computer which will be your training set. The user can either type the absolute path of the TIFF file (separated by ; if there are multiple TIFF files) or just use the Browse button. The user can choose as many TIFF files as they want.

The 'Trained Data Name' field will contain the name of the output Trained Data. If the user specifies an existing trained data, the old training files that were used in developing that trained data will be re-used.

The 'Box Maker' field is the trained data that Tess2Speech trainer will use in making the initial boxes of the input TIFF files. Choosing a trained data that is familiar with your input training images will minimize the errors in the initial box files and minimize the number editing in box files later.

B..2 File Picker Dialog

Figure 37: File Picker Dialog of Tess2Speech Trainer

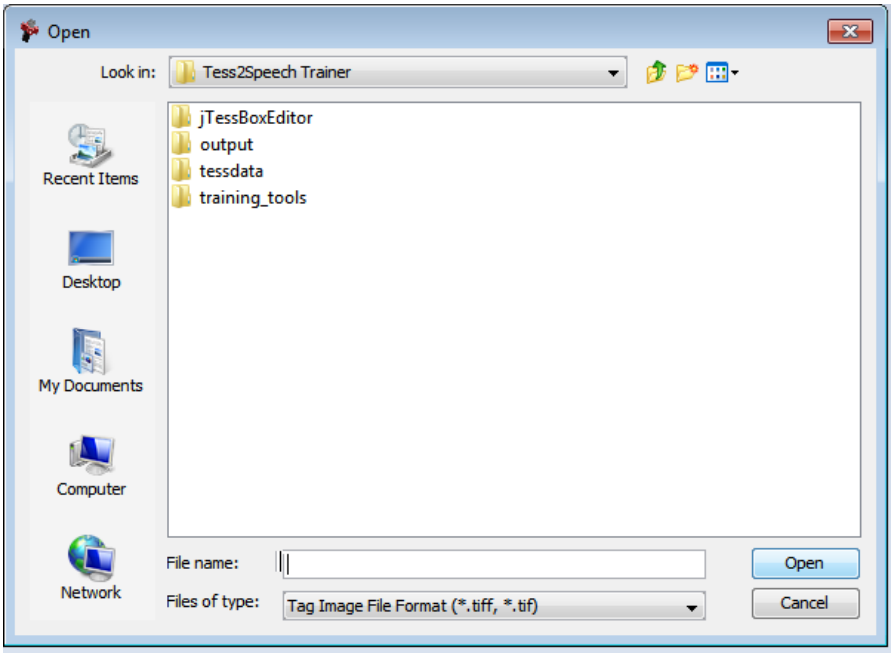
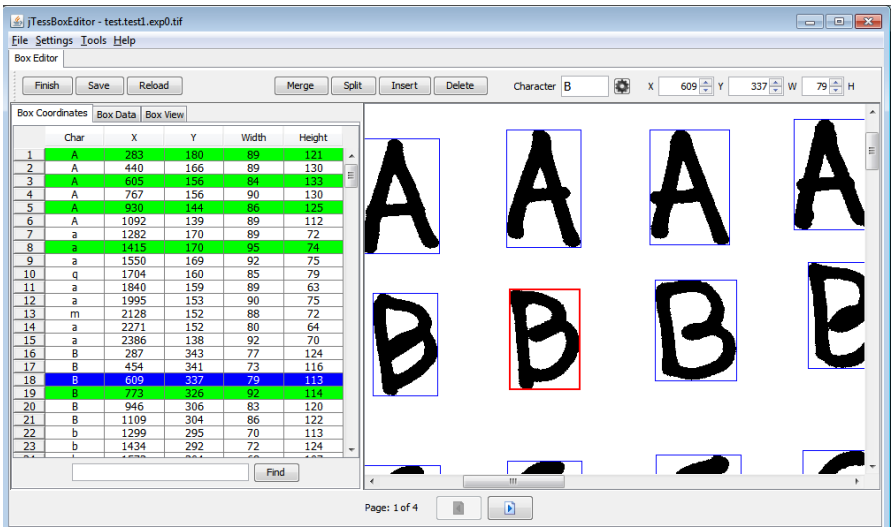


Figure 37 shows the File Picker dialog when 'Browse TIFF/s' is clicked. Multiple TIFF files can be selected at once.

B..3 Box Editor

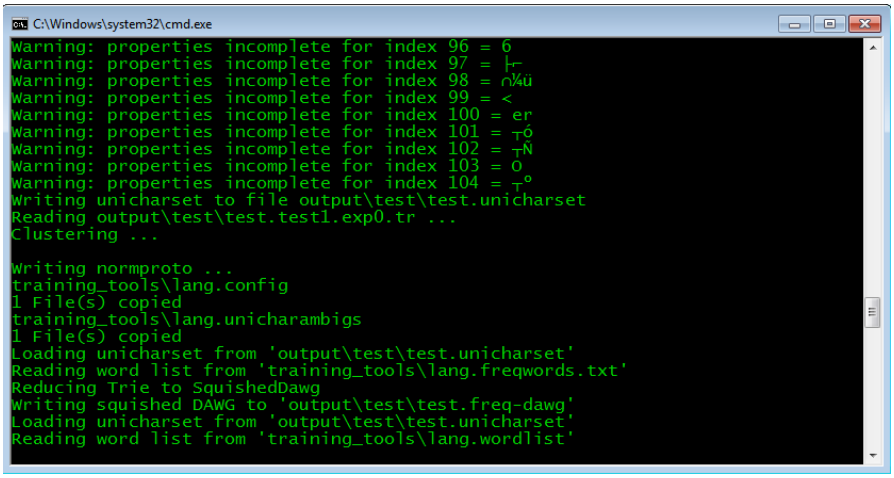
Figure 38: Box Editor (edited jTessBoxEditor [32])



After clicking the 'Start Training' button in figure 36, Tess2Speech will generate the initial box file using the trained data you selected in the 'Box Maker' field. After creating the box files automatically, a box editor will pop-up which allows the user to edit and correct the bounding boxes in the box file. It is an edited jTessBoxEditor by VietOCR [32].

B..4 Automatic Training

Figure 39: Automated Training



After the user finished editing the box files using the editor (figure 39), a

command prompt will automatically pop-up and will perform the training automatically. The user will only need to wait until the command prompt closes and finishes training. After finishing the training, Tess2Speech Trainer will ask the user whether to open the tessdata folder since the output file will be located in Tess2Speech Trainer's tessdata folder.

VI. Discussions

A. Tesseract Training

A..1 Training Results

Using the training methodology discussed in Chapter 4 D., I used a 6-user model training dataset for developing my experimental trained data. These training dataset's character distribution is shown in the table 4. White spaces and new lines are not included in counting the number of characters.

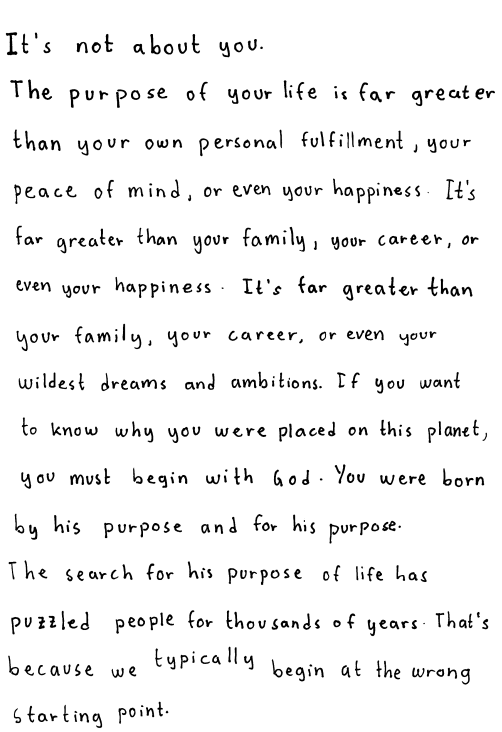
Table 4: Character Distribution of Training Dataset

n - Users	Isolated Characters	Free-flow Text		Total Characters
		Characters	Words	
Train Set for User 1	408	756	153	1164
Train Set for User 2	469	748	153	1217
Train Set for User 3	483	809	162	1292
Train Set for User 4	1575	1001	211	2576
Train Set for User 5	130	200	39	330
Train Set for User 6	400	741	153	1141

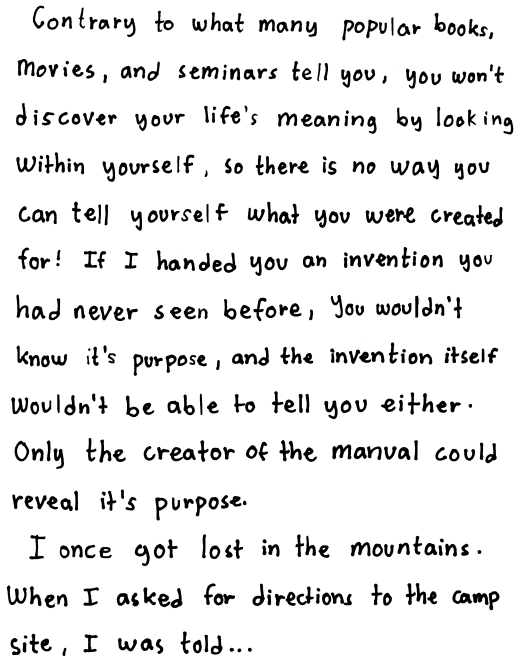
After training Tesseract using these training datasets, I tested the performance of the resulting experimental trained data using the test datasets in figure 40. These test datasets are from the handwriting of user 1, 2, and 3.

Figure 40: Test Data

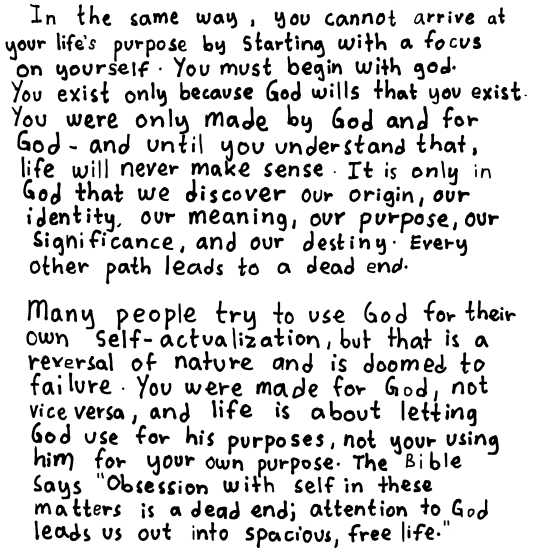
(a) Test Data 1



(b) Test Data 2



(c) Test Data 3



The summary of the test results of the experimental trained data is shown in table 6. I compared the experimental trained data that I created to the default trained data of Tesseract (table 5 shows summary of the default Tesseract).

$$CharacterAccuracy = \frac{C_t - C_e}{C_t} * 100$$

Where C_t = the total number of characters and C_e = the number of misclassified character.

$$WordAccuracy = \frac{C_w - C_e}{C_w} * 100$$

Where C_w = the total number of words and C_e = the number of misclassified words.

Table 5: Summary of the results of the default Tesseract trained data.

Test Data	# of Characters	Recognized Characters	Character Accuracy	# of Words	Recognized Words	Word Accuracy
Test Data 1	472	257	54.45%	103	31	30.10%
Test Data 2	411	253	61.56%	91	27	29.67%
Test Data 3	810	430	53.09%	151	62	41.06%

Table 6: Summary of the results of the experimental trained data that I created.

Test Data	# of Characters	Recognized Characters	Character Accuracy	# of Words	Recognized Words	Word Accuracy
Test Data 1	472	472	100%	103	103	100%
Test Data 2	411	406	98.78%	91	88	96.70%
Test Data 3	810	800	98.77%	151	145	96.03%

As you can see in tables 5 and 6, there is a huge improvement on the accuracy in recognizing handwritten texts in figure 40 after training. The following are the information that we gathered after training:

1. In Test Data 1, there is a **45.5%** improvement in the character recognition accuracy and a **69.9%** improvement on word recognition accuracy.
2. In Test Data 2, there is a **37.22%** improvement in the character recognition accuracy and a **67.03%** improvement on word recognition accuracy.

3. In Test Data 3, there is a **45.68%** improvement in the character recognition accuracy and a **54.97%** improvement on word recognition accuracy.

4. The average character and word accuracy of the default Tesseract trained data are **54.45%** and **33.61%** respectively. On the other hand, the average character and word accuracy of the experimental trained data are **99.18%** and **97.58%** respectively. These show an average increase on the the character and word accuracy by **44.73%** and **63.97%** respectively.

A..2 Problems Encountered in Training

1. If the training image is noisy i.e. taken from the phone's camera, Tesseract can't recognize some of the characters in the training image. This leads to 'APPLY_BOXES FAIL!' error on Tesseract during training. The solution is to use a scanner and save the image file as a 300dpi black and white image in TIFF format for the most optimal training image format.

2. If you are training manually, Tesseract's accuracy may drop if you include symbols and numbers to the training dataset. The solution to this is to create the optional files punc-dawg and number-dawg to give Tesseract a guideline on when to use numbers or symbols in a word. It also helps Tesseract to avoid confusing symbols from letters such as 't' form '+', etc. It is discussed in chapter 3 D..3.

3. Tesseract frequently mis-classify these characters:

   (a) 'i' is recognized as 'l.' or 'l"

   (b) 'l l' is recognized as 'H'

   (c) 't' is recognized as '+'

   (d) 'l' is recognized as '/', '\' or ','

   (e) 'O' is recognized as '0'

   (f) 'e' is recognized as '6'

(g) Characters that have similar capital and small letters such as handwritten 'A', 'C', 'J', 'K', 'M', 'N', 'O', 'P', 'S', 'U', 'V', 'W', 'X', 'Y', and 'Z'.

The workaround for this is to use the set_unicharset_properties.exe and create wordlist-dawg, bigram-dawg, and freq-dawg when you are training manually. Another workaround is to increase the frequencies of these frequent mis-classified characters in the training images.

4. Although the average character and word accuracy that I got is 99.18% and 97.58% respectively, there is still no conclusion that the trained data really achieved those accuracy FOR ALL handwritings since the handwritings used for the training images are only 6. The solution for this is to create an automated Tesseract trainer in which the users can easily train and personalize Tesseract for their own handwritings. With this, anyone can train Tesseract and if proper training is done for the user's handwriting, this will fulfill the objective to reach at least 80% accuracy for all handwritings.

B. Developing Tess2Speech Trainer

B..1 Tess2Speech Trainer Results

As stated before, in order for Tess2Speech to become useful for everyone, it still needs to have the training capability of Tesseract even after the application has been deployed. It's because there is still no guarantee that the trained data that I created will also be usable to other users. Because of this, I developed an automatic Tesseract trainer which is user-friendly and can be used by any user having minimal knowledge about computers. This allows the users to personalize their Tess2Speech and allows it to learn their handwritings if the trained data I created does not produce satisfactory results for their handwritings. This solves the problem of generalizing the handwritings and fulfills my objective to create a user-friendly application that automates Tesseract's training.

The training results using this application will be the same as the manual training method. The only disadvantage of using this application is that the user can only edit the box files. I provided the default values for the dawg files, config file, and other optional files since it will be too complicated for the user to do. The default values is based on my initial experimental trained data that yield the best results.

B..2 Problems Encountered in Developing Tess2Speech Trainer

1. Tesseract cannot run the training tools on the phone. As a workaround, I made Tess2Speech Trainer as a desktop application that is bundled with Tess2Speech mobile application.

2. There are box editors that are already existing. The problem is that I need to integrate it to my Tess2Speech Trainer application. In order to do this, I used an open-source box editor by vietOCR which is the jTessBoxEditor [32]. Since it is open-source, I managed to edit its source code and integrate it into my application.

C. Tess2Speech Mobile Application

C..1 Tess2Speech Results

Since Tess2Speech uses Tesseract as its OCR Engine, I can use the .traineddata files that are produced from Tesseract's training in desktop computers using Tess2Speech Trainer. Because of this, I managed to create an Android Application that fulfills the objectives I specified. The user can upload an image that contains the handwriting (can also be computer printed texts) or write the handwritten text itself on the mobile's screen through a canvas, and convert it to text or speech. The user can also personalize Tess2Speech by training their own handwriting using Tess2Speech Trainer and used the output .traineddata files to their application.

The user can also download other .traineddata files from Tesseract's home page, through updates, or by downloading from a host site.

The user can also save the converted text into a .txt or .pdf file. The user can also save the converted speech into a .wav file. Since there are a lot of other functionalities that can be done, I tried maximizing the capability of Tess2Speech by adding more functionalities such as PDF-to-Image and vice-versa, Ebook-to-Speech, and PDF-to-Speech.

C..2 Problems Encountered in Tess2Speech Application

1. Tesseract-OCR [5] is programmed in C++ which means it is not compatible with the programming language that I will use which is Java. As a solution, I used tess-two [35] which contains tools for compiling Tesseract and Leptonica for use on the Android Platform. It provides a Java API for accessing natively-compiled Tesseract and Leptonica APIs.

2. Text-to-Speech systems on Android only supports up to 4000 Characters at a time. As a solution, if a paragraph is more than 4000 characters long, I chopped the paragraph into strings of 4000 length and saved it into an ArrayList. I then used this ArrayList as input since Text-to-Speech system supports an ArrayList of strings as input. As a consequence, when saving a paragraph with greater than 4000 characters into a .wav file, the number of .wav files that will be created will be the same as the number of elements in the ArrayList.

3. Tesseract does not support PDF as an input and Android cannot display PDF files for API below 21. As a solution, I used PDFViewer library by Joan Zapata [38] in order to view PDFs for APIs below 21. I also examined the library and found a workaround for using PDF as a Tesseract input. The workaround is that I converted PDF to images first by using VuDroid [39], which is included in the PDFViewer library, before feeding it to Tesseract.

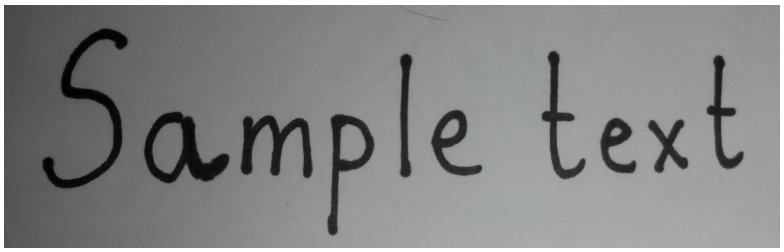
4. Since Android stylus functionalities like those in Galaxy Notes are propri-

etary, I need to create my own canvas for supporting handwritings that are written in the mobile's screen. As a workaround, I managed to create a canvas by imitating how MS Paint works, although writing is not that smooth and versatile unlike in Galaxy Note. The performance of the canvas is also dependent on the hardware (screen capability).

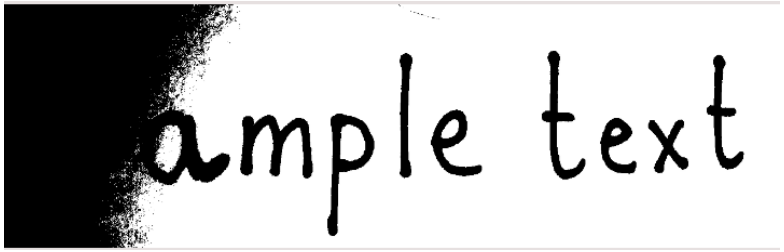
5. Tess2Speech is very hardware dependent. The quality of the image depends on the phone's camera (unless it was scanned and then sent to phone). Since images taken from the low quality camera is usually noisy, it is very hard to get an image that Tess2Speech can read. As a solution, I implemented the capability on Tess2Spech to perform image pre-processing to an image before converting the image to text or speech.

Figure 41: Image Pre-processing

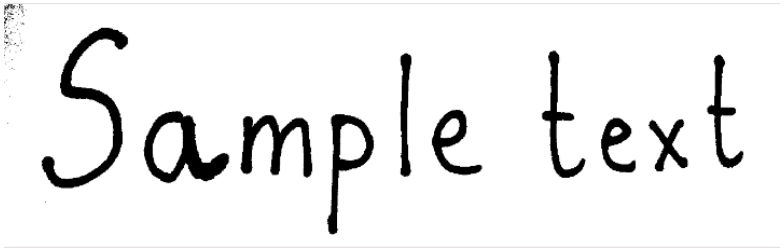
(a) Raw Image



(b) Image processed only by Tesseract's Otsu Binarization



(c) Image processed by GrayScaling, Brightness and Contrast adjustment, and Tesseract's Otsu Binarization



6. If the image is too large, processing time can take a very long time. Large images with large texts can also confuse Tesseract since it will treat the letters as a background and it will try look for texts inside the letters. As a solution, I implemented the capability to rescale the image into a smaller dimension before converting the image to text or speech.

7. Since De-skewing is hard to implement, I implemented an alternative which allows the users to manually rotate an image. I also implemented the crop functionality to allow the users to remove unnecessary parts of the image that can produce garbage texts.

8. The trained data for handwritten texts yields low accuracy when used to recognize computer typed texts. As a solution, since Tesseract can use multiple trained data at the same time, the user can choose any combination of .traineddata files including 'eng.traineddata' in the Tesseract Language settings of Tess2Speech.

D. Significance of Tess2Speech

Since Tess2Speech is free and open-source, it provides users with a free, portable, non-proprietary, and trainable Intelligent Character Recognition Engine. This also means that Tess2Speech can still improve over time by re-training over and over again.

Tess2Speech can also read images of computer-typed texts aside from the ability to read handwritten texts which makes it both and OCR and ICR engine.

Since the users can write on the mobile phone's screen, it can help people with speech impairments and help them communicate. Tess2Speech is cheap, easy to use, and inexpensive unlike other speech synthesizers and handwritten-to-text applications. It can also help pre-school students to know the proper pronunciation of words that they write on the mobile phone's screen. Tess2Speech can also help users to proof-read their written works by listening to the voice equivalent of their written works.

Tess2Speech also has many functionalities like saving the converted text or speech into a .txt, .pdf, or .wav file, Ebook-to-Speech (Audio book), Converting Image-to-PDF and vice-versa, and PDF-to-Speech functionalities.

VII. Conclusions

Tess2Speech is an Intelligent Character Recognition Android application which can recognize both computer-typed and non-cursive handwritten texts and convert it to either text or speech. With the help of Tess2Speech Trainer, which is a desktop application, Tess2Speech is re-trainable and can be personalized by the users with minimal effort.

Tess2Speech is developed to provide the speech impaired a way to communicate, and also to provide researchers a free, open-source, non-proprietary ICR which is trainable.

This application also proves that Tesseract is indeed trainable for handwritten texts. Although, it is only limited to non-cursive handwritten texts since Tesseract relies on segmentation of characters. Because of this, it can be said that it is almost impossible to recognize cursive texts using Tesseract.

The development of this application also helped to determine the limitations and capabilities of Tesseract, and what training methods and configurations are effective for achieving high accuracy with Tesseract.

This application is currently at its Beta stage. Once it gains popularity, different users and researchers may collaborate to improve the trained data for handwriting until such time that they create a generalized trained data for most, if not all, handwritten texts.

VIII. Recommendations

It is suggested that in order to further improve the accuracy of Tess2Speech' handwriting recognition, a higher number of sample handwritings (n-users) should be used. A higher number of characters and words are also suggested when creating training datasets.

Improving the canvas functionality of Tess2Speech that can rival Galaxy Note's can greatly improve the accuracy of Tess2Speech when recognizing handwritten texts on the phone's screen.

One of the main reason why the accuracy of Tesseract drops is because of image noises. Using open-source libraries for image pre-processing or developing an algorithm that can greatly reduce image noises can drastically improve Tesseract's accuracy. Such algorithms are adaptive thresholding and applying different binarization techniques.

Another improvement would be finding a way to train Tesseract in a smart phone so that Tess2Speech Trainer will not be needed. It can open a possibilty for Tesseract to learn handwritings dynamically without user intervention.

Adding functionality like language translator can also add usability to Tess2Speech. Lastly, if there is an open-source OCR Engine which is better than Tesseract, you can completely replace the OCR engine of this application.

IX. Bibliography

[1] P. Christensson, "Character Encoding Definition." http://techterms.com, September 2010.

[2] K. Kumar.K, M. V. S. Muvvala, P. S. D. Sruthi, and P. Dinesh, "Intelligent Character Recognition (ICR): A Novel Algorithm to Extract Text from a Scanned Form Based Image," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 2, no. 6, pp. 47–55, 2013.

[3] S. Rakshit and S. Basu, "Development of a Multi-User Handwriting Recognition System Using Tesseract Open Source OCR Engine," *PROC. International Conference on C3IT*, pp. 240–247, 2009.

[4] S. Ch, S. Mahna, and N. Kashyap, "Optical Character Recognition on Handheld Devices," *International Journal of Computer Applications IJCA*, vol. 115, no. 22, pp. 10–13, 2015.

[5] Google, "Tesseract-OCR." https://code.google.com/p/tesseract-ocr/, 2006 - 2016.

[6] Google, "Android OS." http://www.android.com/ or http://developer.android.com/, 2008.

[7] American Speech-Language-Hearing Association, "Aphasia." http://www.asha.org/public/speech/disorders/Aphasia/.

[8] Stroke Association, "Communication Aids and Computer-Based Therapy after Stroke." http://www.stroke.org.uk/, 2013.

[9] Free Press Journal, "2.5 bn smartphone users globally by 2015," *The Free Press Journal*, July 2014.

[10] Stefan, "Samsung just bought 5stylus!," *Android Authority*, January 2013.

[11] S. Vijiyarani and A. Sakila, "Performance Comparison of OCR Tools," *International Journal of UbiComp (IJU)*, vol. 6, pp. 19 – 30, July 2015.

[12] B. Verma, M. Blumenstein, and S. Kulkarni, "Recent Achievements in Offline Handwriting Recognition Systems,"

[13] A2iA, "A2iA document reader ICR for handwritten text." http://www.a2ia.com/sites/default/files/product_files/a2ia_documentreader_datasheet_en.pdf/, 2015.

[14] Y. Williams, "What is Speech impairment? - Defenitions, Causes and Characteristics." http://www.a2ia.com/sites/default/files/product_files/a2ia_documentreader_datasheet_en.pdf/.

[15] iSpeak4U, "Handyspeech." https://itunes.apple.com/us/app/handyspeech/id563600464?mt=8, 2014.

[16] A. Christie, "Multiple Benefits of Text to Speech Applications." http://www.lc2.ca/item/85-multiple-benefits-of-text-to-speech-applications, November 2012.

[17] A. Aparna and I. Muthumani, "Optical Character Recognition for Handwritten Cursive English characters," *International Journal of Computer Science and Information Technologies (IJCSIT)*, vol. 5, no. 1, pp. 847 – 848, 2014.

[18] R. Smith, "An Overview of the Tesseract OCR Engine." http://static.googleusercontent.com/media/research.google.com/en//pubs/archive/33418.pdf.

[19] S. Singh, "Optical Character Recognition Techniques: A Survey," *Journal of Emerging Trends in Computing and Information Sciences*, vol. 4, pp. 545 – 550, June 2013.

[20] S. Dhiman and A. Singh, "Tesseract Vs Gocr A Comparative Study," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 2, pp. 80 – 83, September 2013.

[21] M. Nayak and A. K. Nayak, "Odia Characters Recognition by Training Tesseract OCR Engine," *International Journal of Computer Applications (IJCA)*, pp. 25 – 30, December 2013.

[22] C. Patel, A. Patel, and D. Patel, "Optical Character Recognition by Open Source OCR Tool Tesseract: A Case Study," *International Journal of Computer Applications*, vol. 55, pp. 50–56, October 2012.

[23] K. Ntirogiannis, "Performance Evaluation Methodology for Historical Document Image Binarization," *Institute of Electrical and Electronics Engineers (IEEE)*, vol. 22, pp. 595 – 609, September 2012.

[24] D. sasirekha and E. Chandra, "Text-to-Speech: A Simple Tutorial," *International Journal of Soft Computing and Engineering (IJSCE)*, vol. 2, pp. 275 – 278, March 2012.

[25] R. Mithe and N. D. Supriya Indalkar, "Optical Character Recognition," *International Journal of Recent Technology and Engineering (IJRTE))*, vol. 2, pp. 72 – 75, March 2013.

[26] S. Barve, "Optical Character Recognition Using Artificial Neural Network," *International Journal of Advanced Research in Computer Engineering & Technology)*, vol. 1, pp. 131 – 133, June 2012.

[27] S. Vijayarani and A. Sakila, "Performance Comparison of OCR Tools," *International Journal of UbiComp (IJU))*, vol. 6, pp. 19 – 29, July 2015.

[28] R. Urbanowicz and J. Moore, "Learning Classier Systems: A Complete Introduction, Review, and Roadmap," *Journal of Articial Evolution and Applications*, vol. 2009, June 2009.

[29] J. Brownlee, "Clever Algorithms: Nature-Inspired Programming Recipes." http://www.cleveralgorithms.com/nature-inspired/evolution/learning_classifier_system.html, 2015.

[30] T. Helland, "Seven grayscale conversion." http://www.tannerhelland.com/3643/grayscale-image-algorithm-vb6/.

[31] B. Cigan, "Java image binarization using Otsu's algorithm." http://developer.bostjan-cigan.com/java-image-binarization/, December 2012.

[32] VietOCR, "jTessBoxEditor." http://vietocr.sourceforge.net/training.html, 2008 - 2016.

[33] Wutka, "Directed Acyclic Word Graphs." http://www.wutka.com/dawg.html.

[34] J. Gosling and et al., *Java Language Specification*. Java SE 8 Edition, Addison-Wesley Professional, 2014.

[35] R. Theis, "Tess-Two [Eclipse Android Library for Tesseract]." https://github.com/rmtheis/tess-two, 2011.

[36] J. Polites, "GestureImageView." https://github.com/jasonpolites/gesture-imageview, 2012.

[37] ArthurHub, "Android-Image-Cropper." https://github.com/ArthurHub/Android-Image-Cropper/wiki, 2013.

[38] J. Zapata, "Android-pdfview." https://github.com/JoanZapata/android-pdfview, 2014.

[39] Google, "VuDroid." https://code.google.com/archive/p/vudroid/.

[40] P. Siegman, "EPubLib." https://github.com/psiegman/epublib.

[41] A. Kaloer, "Android-File-Picker-Activity." https://github.com/AndersKaloer/Android-File-Picker-Activity/tree/master/src/com/kaloer/filepicker, 2011.

X. Appendix

A. Box File


```
A 283 3208 372 3329 0
A 440 3213 529 3343 0
A 605 3220 689 3353 0
A 767 3223 857 3353 0
A 930 3240 1016 3365 0
A 1092 3258 1181 3370 0
a 1282 3267 1371 3339 0
a 1415 3265 1510 3339 0
a 1550 3265 1642 3340 0
a 1704 3270 1789 3349 0
a 1840 3287 1929 3350 0
a 1995 3281 2085 3356 0
a 2128 3285 2216 3357 0
a 2271 3293 2351 3357 0
a 2386 3301 2478 3371 0
B 287 3042 364 3166 0
B 454 3052 527 3168 0
B 609 3059 688 3172 0
B 773 3069 865 3183 0
B 946 3083 1029 3203 0
B 1109 3083 1195 3205 0
b 1299 3101 1369 3214 0
b 1434 3093 1506 3217 0
b 1573 3098 1641 3205 0
b 1711 3094 1780 3211 0
b 1851 3096 1925 3210 0
b 2001 3086 2072 3206 0
b 2137 3072 2213 3200 0
b 2282 3067 2354 3181 0
b 2407 3048 2487 3177 0
C 272 2852 360 2967 0
C 443 2857 530 2973 0
C 606 2858 690 2978 0
C 774 2865 870 2985 0
C 940 2867 1035 2990 0
C 1106 2878 1205 2993 0
c 1303 2896 1370 2967 0
c 1432 2904 1498 2975 0
c 1568 2911 1637 2980 0
c 1699 2909 1772 2983 0
c 1838 2899 1908 2976 0
c 1986 2907 2054 2974 0
c 2131 2917 2201 2988 0
c 2271 2921 2341 2986 0
c 2402 2918 2483 2984 0
D 285 2675 354 2800 0
D 454 2674 523 2802 0
D 614 2680 688 2804 0
D 779 2687 851 2821 0
D 937 2690 1007 2818 0
D 1113 2686 1181 2822 0
d 1287 2702 1354 2824 0
d 1409 2705 1476 2824 0
d 1545 2695 1619 2813 0
d 1678 2701 1751 2832 0
d 1836 2708 1907 2835 0
d 1972 2705 2049 2836 0
d 2115 2694 2192 2839 0
d 2240 2714 2310 2845 0
d 2354 2708 2430 2840 0
E 275 2504 352 2617 0
E 449 2503 534 2621 0
E 621 2512 710 2622 0
E 793 2515 875 2629 0
E 949 2512 1033 2625 0
E 1110 2516 1189 2640 0
e 1279 2531 1354 2606 0
e 1404 2537 1479 2610 0
e 1548 2543 1622 2634 0
e 1690 2548 1761 2620 0
e 1827 2545 1898 2630 0
e 1951 2548 2026 2642 0
e 2073 2550 2136 2637 0
e 2191 2546 2270 2624 0
e 2300 2549 2379 2624 0
e 2421 2561 2499 2634 0
F 293 2312 359 2440 0
F 445 2324 538 2423 0
F 627 2316 703 2414 0
F 812 2319 884 2431 0
F 964 2329 1036 2443 0
F 1109 2312 1176 2443 0
f 1282 2346 1338 2457 0
f 1408 2353 1473 2464 0
f 1522 2347 1595 2461 0
f 1651 2352 1723 2460 0
f 1785 2355 1846 2468 0
f 1917 2357 1986 2466 0
f 2064 2345 2118 2464 0
f 2205 2357 2259 2468 0
f 2320 2349 2382 2472 0
f 2420 2357 2498 2466 0
G 273 2129 361 2267 0
G 458 2130 534 2264 0
G 620 2136 697 2267 0
G 797 2135 873 2268 0
G 948 2128 1019 2271 0
G 1085 2122 1164 2259 0
h 1294 2160 1368 2268 0
h 1424 2147 1483 2277 0
h 1542 2158 1617 2280 0
h 1672 2157 1727 2284 0
h 1796 2149 1862 2287 0
h 1924 2156 1986 2283 0
h 2056 2145 2124 2287 0
h 2191 2146 2270 2277 0
h 2315 2152 2383 2268 0
h 2425 2146 2508 2283 0
H 282 1947 366 2090 0
H 453 1949 536 2077 0
H 610 1953 729 2081 0
H 796 1947 894 2086 0
H 948 1933 1036 2082 0
H 1090 1952 1174 2086 0
g 1285 1913 1358 2030 0
g 1407 1920 1475 2042 0
g 1548 1909 1622 2035 0
g 1669 1913 1749 2043 0
g 1800 1932 1869 2045 0
g 1931 1922 2000 2039 0
g 2056 1930 2129 2044 0
g 2195 1930 2257 2046 0
g 2304 1917 2382 2045 0
g 2430 1922 2497 2048 0
I 280 1783 355 1892 0
I 444 1779 530 1889 0
I 610 1798 704 1891 0
I 789 1794 874 1885 0
I 942 1788 1030 1886 0
I 1077 1771 1160 1887 0
i 1307 1799 1329 1884 0
i 1404 1792 1424 1885 0
i 1494 1789 1523 1892 0
i 1577 1774 1628 1880 0
i 1672 1773 1698 1889 0
i 1773 1789 1800 1895 0
i 1863 1796 1885 1901 0
i 1948 1803 1988 1897 0
i 2058 1795 2099 1888 0
i 2146 1806 2178 1895 0
i 2223 1811 2257 1899 0
i 2309 1806 2355 1892 0
i 2415 1800 2444 1902 0
J 290 1643 346 1740 0
J 461 1633 524 1744 0
J 618 1640 693 1739 0
J 803 1640 861 1733 0
J 953 1611 1009 1736 0
J 1091 1628 1146 1736 0
j 1292 1622 1324 1721 0
j 1370 1609 1410 1721 0
j 1469 1595 1512 1729 0
j 1583 1588 1627 1714 0
j 1688 1583 1721 1721 0
j 1758 1586 1816 1718 0
j 1897 1569 1944 1726 0
j 2012 1572 2048 1723 0
j 2115 1577 2160 1737 0
j 2214 1590 2256 1697 0
j 2313 1583 2353 1728 0
j 2395 1583 2436 1733 0
K 286 1435 360 1565 0
K 454 1441 543 1554 0
K 624 1442 712 1552 0
K 788 1431 890 1552 0
K 950 1430 1037 1552 0
K 1084 1435 1166 1553 0
k 1260 1426 1320 1512 0
k 1359 1430 1416 1495 0
k 1454 1420 1507 1492 0
k 1553 1426 1608 1498 0
k 1649 1426 1716 1502 0
k 1764 1432 1834 1506 0
k 1885 1427 1969 1507 0
k 2038 1424 2112 1507 0
k 2192 1424 2261 1499 0
k 2322 1406 2391 1491 0
L 279 1273 353 1403 0
L 443 1285 516 1396 0
L 594 1279 679 1395 0
L 756 1282 834 1389 0
L 902 1282 977 1397 0
L 1027 1277 1091 1395 0
L 1142 1269 1199 1392 0
l 1295 1274 1318 1376 0
l 1380 1265 1404 1373 0
l 1480 1272 1522 1361 0
l 1585 1279 1606 1366 0
l 1660 1265 1719 1361 0
l 1787 1267 1806 1368 0
l 1878 1269 1902 1369 0
l 1988 1270 2013 1373 0
l 2110 1283 2132 1370 0
l 2236 1275 2268 1378 0
l 2362 1274 2381 1369 0
M 273 1095 370 1225 0
M 439 1114 539 1231 0
M 603 1106 702 1238 0
M 776 1098 886 1220 0
M 942 1115 1048 1225 0
M 1095 1107 1197 1232 0
m 1286 1109 1379 1188 0
m 1428 1099 1526 1178 0
m 1582 1090 1689 1178 0
m 1749 1089 1856 1172 0
m 1909 1086 2020 1169 0
m 2090 1075 2193 1157 0
m 2256 1071 2361 1158 0
N 263 907 350 1035 0
N 436 911 541 1029 0
N 609 911 685 1031 0
N 775 901 858 1044 0
N 944 923 1022 1071 0
N 1083 917 1170 1066 0
n 1269 947 1343 1023 0
n 1425 950 1516 1031 0
n 1588 943 1669 1028 0
n 1755 950 1845 1031 0
n 1929 947 2002 1039 0
n 2093 946 2176 1037 0
n 2269 934 2361 1026 0
O 260 725 358 870 0
O 424 723 535 863 0
O 591 718 704 858 0
O 765 721 885 861 0
O 938 719 1056 853 0
O 1085 734 1207 857 0
o 1274 734 1346 812 0
o 1439 737 1507 806 0
o 1596 727 1668 799 0
o 1765 727 1837 798 0
o 1922 725 1998 790 0
o 2087 713 2160 783 0
o 2289 708 2359 774 0
P 279 501 343 662 0
P 435 496 502 670 0
P 599 477 671 663 0
P 779 475 850 668 0
P 959 493 1033 674 0
P 1120 504 1198 668 0
p 1292 467 1365 602 0
P 1445 466 1517 610 0
P 1605 490 1681 604 0
P 1772 465 1845 597 0
P 1926 471 2000 594 0
P 2104 473 2171 592 0
P 2283 471 2350 601 0
Q 80 3248 171 3362 1
Q 270 3242 367 3361 1
Q 455 3258 552 3366 1
Q 637 3270 737 3378 1
Q 801 3277 914 3382 1
Q 990 3283 1087 3386 1
q 1186 3228 1268 3352 1
q 1330 3243 1428 3363 1
q 1500 3269 1587 3370 1
q 1646 3254 1746 3367 1
q 1800 3261 1900 3378 1
q 1956 3252 2042 3374 1
q 2112 3253 2196 3369 1
R 85 3035 170 3130 1
```

R 273 3026 356 3133 1
R 462 3038 551 3143 1
R 641 3042 732 3152 1
R 807 3048 891 3157 1
R 974 3065 1060 3169 1
r 1207 3061 1258 3124 1
r 1352 3074 1411 3137 1
r 1505 3084 1567 3144 1
r 1650 3087 1710 3147 1
r 1804 3103 1867 3166 1
r 1975 3120 2041 3189 1
r 2113 3126 2176 3193 1
S 101 2811 172 2917 1
S 279 2808 348 2921 1
S 468 2809 529 2916 1
S 661 2834 728 2933 1
S 834 2840 891 2945 1
S 991 2843 1047 2945 1
s 1198 2849 1247 2918 1
s 1352 2859 1392 2927 1
s 1502 2870 1554 2934 1
s 1641 2872 1687 2940 1
s 1780 2874 1825 2946 1
s 1926 2871 1979 2955 1
s 2076 2877 2121 2960 1
T 90 2586 182 2681 1
T 254 2598 355 2685 1
T 427 2597 524 2694 1
T 634 2585 720 2693 1
T 801 2599 902 2711 1
T 969 2602 1072 2708 1
t 1183 2608 1266 2735 1
t 1320 2611 1381 2711 1
t 1477 2620 1531 2711 1
t 1619 2609 1673 2722 1
t 1763 2623 1831 2718 1
t 1922 2616 1977 2720 1
t 2071 2614 2128 2704 1
U 94 2367 178 2471 1
U 261 2378 358 2480 1
U 447 2378 535 2487 1
U 643 2381 732 2481 1
U 819 2368 913 2487 1
U 991 2379 1085 2496 1
u 1190 2388 1245 2441 1
u 1316 2378 1376 2439 1
u 1444 2384 1513 2436 1
u 1578 2392 1653 2448 1
u 1710 2388 1785 2444 1
v 1830 2393 1872 2445 1
v 1911 2389 1962 2446 1
u 1992 2387 2061 2439 1
u 2091 2388 2145 2451 1
V 89 2133 168 2247 1
V 259 2144 335 2240 1
V 430 2134 511 2253 1
V 637 2136 724 2254 1
V 808 2142 895 2261 1
V 985 2141 1084 2267 1
v 1168 2145 1226 2204 1
v 1273 2144 1334 2203 1
v 1387 2143 1450 2209 1
v 1517 2147 1582 2213 1
v 1633 2149 1706 2218 1
v 1744 2154 1818 2226 1
v 1876 2159 1944 2224 1
v 1997 2160 2065 2233 1
v 2097 2164 2164 2227 1
v 2206 2161 2261 2220 1
v 2297 2151 2345 2212 1
W 61 1919 178 2028 1
W 231 1915 352 2039 1
W 403 1923 538 2040 1
W 606 1929 720 2058 1
W 783 1937 904 2061 1
W 958 1941 1081 2058 1
w 1146 1956 1233 2009 1
w 1283 1958 1375 2009 1
w 1436 1959 1533 2011 1
w 1585 1969 1686 2019 1
w 1750 1974 1859 2027 1
w 1917 1974 2028 2032 1
w 2084 1974 2179 2035 1
w 2223 1984 2316 2044 1
X 82 1703 181 1819 1
X 233 1693 342 1829 1
X 403 1698 498 1829 1
X 634 1691 721 1833 1
X 785 1702 895 1825 1
X 969 1718 1072 1833 1
x 1161 1713 1223 1790 1
x 1287 1704 1357 1790 1
x 1432 1702 1497 1791 1
x 1573 1717 1635 1798 1
x 1696 1723 1767 1799 1
x 1836 1701 1904 1801 1
x 1969 1720 2049 1810 1
x 2094 1722 2181 1807 1
x 2223 1702 2301 1797 1
Y 93 1427 191 1591 1
Y 227 1461 320 1594 1
Y 382 1484 494 1598 1
Y 610 1464 705 1597 1
Y 795 1472 906 1602 1
Y 973 1466 1064 1612 1
y 1130 1494 1200 1607 1
y 1280 1498 1349 1602 1
y 1429 1510 1496 1607 1
y 1571 1498 1637 1598 1
y 1694 1487 1768 1593 1
y 1834 1482 1901 1580 1
y 1961 1470 2019 1563 1
y 2078 1475 2154 1581 1
y 2220 1460 2282 1576 1
Z 82 1274 159 1387 1
Z 242 1279 315 1388 1
Z 395 1263 466 1378 1
Z 591 1268 669 1372 1
Z 749 1252 845 1370 1
Z 906 1249 994 1362 1
z 1139 1235 1197 1311 1
z 1260 1237 1312 1303 1
z 1375 1228 1425 1297 1
z 1487 1220 1543 1294 1
z 1587 1231 1649 1291 1
z 1700 1234 1754 1291 1
z 1783 1228 1834 1295 1
z 1888 1231 1961 1297 1
z 2003 1232 2066 1291 1
z 2113 1233 2170 1292 1
z 2212 1228 2271 1292 1
Z 65 1021 147 1135 1
Z 217 1022 313 1130 1
Z 383 1029 504 1126 1
Z 569 1023 662 1128 1
Z 746 1021 856 1129 1
Z 892 1012 1006 1132 1
z 1108 1005 1165 1077 1
z 1222 994 1285 1073 1
z 1342 1003 1410 1072 1
z 1460 998 1528 1070 1
z 1595 1002 1656 1069 1
z 1705 1005 1779 1071 1
z 1825 1010 1906 1076 1
z 1943 1015 2014 1082 1
z 2060 1018 2133 1085 1
z 2163 1014 2258 1078 1
T 90 3320 141 3418 2
h 172 3311 223 3418 2
e 237 3311 295 3367 2
(399 3314 448 3408 2
q 472 3273 536 3363 2
u 533 3313 584 3363 2
i 597 3318 615 3382 2
c 638 3306 695 3361 2
k 703 3316 781 3414 2
) 802 3301 853 3417 2
b 948 3299 1009 3408 2
r 1020 3306 1067 3352 2
o 1075 3303 1122 3352 2
w 1141 3305 1213 3354 2
n 1239 3302 1295 3360 2
f 1388 3301 1436 3400 2
o 1443 3303 1494 3357 2
x 1507 3294 1547 3352 2
j 1628 3236 1664 3357 2
w 1693 3291 1760 3341 2
u 1797 3277 1859 3357 2
m 1885 3301 1961 3360 2
p 1979 3248 2032 3359 2
s 2033 3294 2068 3353 2
! 2080 3290 2102 3383 2
o 2156 3296 2202 3349 2
v 2209 3299 2254 3350 2
e 2253 3293 2294 3349 2
r 2310 3300 2349 3347 2
{ 115 3095 171 3209 2
t 201 3115 243 3201 2
h 255 3112 308 3192 2
e 316 3113 372 3160 2
} 386 3089 429 3194 2
$ 555 3086 623 3212 2
3 652 3097 697 3190 2
, 715 3079 730 3123 2
4 752 3089 805 3181 2
5 832 3093 875 3184 2
6 882 3098 940 3177 2
. 967 3109 984 3128 2
7 1015 3093 1068 3177 2
8 1088 3092 1136 3175 2
1 1146 3088 1197 3171 2
2 1224 3091 1276 3164 2
< 1339 3094 1391 3152 2
l 1413 3088 1432 3165 2
a 1448 3082 1508 3138 2
z 1519 3082 1569 3139 2
y 1576 3049 1629 3133 2
> 1638 3088 1681 3145 2
1721 3072 1807 3148 2
9 1833 3068 1887 3150 2
0 1887 3070 1944 3136 2
d 1994 3049 2050 3146 2
o 2058 3047 2108 3099 2
g 2116 3014 2170 3099 2
& 2230 3048 2288 3141 2
d 124 2870 176 2987 2
u 213 2872 260 2932 2
o 285 2873 336 2928 2
k 355 2866 426 2962 2
, 474 2848 498 2891 2
d 589 2860 643 2968 2
u 669 2851 725 2919 2
c 748 2868 797 2917 2
k 822 2870 889 2948 2
/ 885 2833 995 2941 2
g 1024 2841 1086 2925 2
o 1091 2861 1142 2927 2
o 1156 2861 1218 2919 2
s 1241 2865 1283 2925 2
e 1288 2857 1357 2916 2
, 1394 2837 1413 2876 2
a 1493 2851 1573 2918 2
s 1599 2836 1645 2910 2
l 1740 2842 1760 2920 2
2 1761 2842 1816 2920 2
. 1823 2857 1841 2873 2
5 1879 2840 1934 2917 2
% 1980 2839 2079 2933 2
o 2173 2848 2225 2900 2
f 2242 2848 2289 2930 2
E 130 2654 192 2756 2
− 227 2688 263 2700 2
m 302 2652 371 2720 2
a 391 2645 440 2700 2
i 449 2646 468 2723 2
l 497 2635 513 2743 2
f 665 2638 732 2750 2
r 741 2650 794 2693 2
o 815 2647 863 2701 2
m 884 2656 960 2702 2
a 1046 2644 1111 2711 2
s 1119 2638 1155 2708 2
p 1161 2613 1216 2708 2
a 1216 2648 1277 2702 2
m 1279 2652 1352 2712 2
m 1356 2651 1442 2708 2
e 1457 2646 1514 2706 2
r 1519 2652 1578 2702 2
@ 1617 2628 1702 2720 2
w 1731 2635 1805 2691 2
e 1803 2642 1849 2693 2
b 1858 2633 1915 2723 2
. 1926 2648 1944 2667 2
c 1978 2631 2028 2687 2
o 2031 2625 2077 2674 2
m 2084 2623 2185 2676 2
i 2271 2621 2300 2693 2
s 2305 2619 2342 2669 2
s 121 2447 160 2517 2
p 175 2410 228 2496 2
a 229 2431 291 2487 2
m 296 2432 379 2496 2
? 397 2437 450 2530 2
L 506 2431 556 2520 2
o 560 2425 616 2473 2
l 642 2426 662 2527 2
’ 688 2497 706 2539 2
s 739 2439 778 2497 2
_ 815 2424 926 2441 2
W 983 2432 1069 2500 2
o 1087 2433 1145 2483 2
w 1166 2432 1253 2488 2
! 1287 2406 1309 2498 2
? 1355 2402 1400 2499 2
” 1431 2466 1477 2510 2
^ 1532 2456 1596 2502 2
+ 1635 2388 1702 2476 2
˜ 1766 2414 1859 2457 2
− 1927 2432 1971 2446 2
= 2038 2398 2102 2442 2
: 2163 2389 2180 2454 2
; 2234 2369 2257 2445 2
[2319 2374 2361 2472 2
] 2419 2365 2462 2461 2
H 134 2207 187 2313 2
e 196 2213 243 2259 2
i 261 2213 280 2284 2
k 294 2213 339 2294 2
o 353 2207 395 2260 2
, 440 2176 458 2216 2
f 512 2207 561 2300 2
e 566 2212 611 2257 2
i 625 2207 645 2275 2
g 662 2170 711 2257 2
n 721 2207 765 2255 2
i 776 2206 796 2267 2
n 811 2207 865 2251 2

g 867 2159 926 2247 2
s 988 2200 1031 2262 2
l 1051 2202 1069 2277 2
e 1087 2199 1134 2254 2
e 1150 2195 1204 2252 2
p 1230 2157 1279 2247 2
, 1307 2172 1324 2210 2
k 1384 2200 1436 2281 2
e 1454 2186 1501 2247 2
p 1507 2165 1558 2241 2
t 1565 2185 1617 2267 2
h 1696 2190 1751 2273 2
e 1759 2188 1806 2232 2
r 1811 2194 1857 2231 2
b 1942 2188 1994 2274 2
r 2002 2191 2062 2239 2
e 2058 2188 2108 2239 2
a 2109 2184 2156 2234 2
t 2160 2181 2206 2260 2
h 2204 2193 2253 2272 2
i 2259 2188 2277 2243 2
n 2292 2182 2343 2231 2
g 2352 2151 2408 2227 2
d 118 1954 168 2068 2
e 199 1972 242 2033 2
e 267 1973 328 2054 2
p 356 1950 405 2038 2
a 499 1970 558 2025 2
n 576 1972 631 2028 2
d 656 1970 710 2057 2
s 812 1965 856 2033 2
l 884 1977 902 2068 2
o 920 1965 978 2024 2
w 1005 1969 1088 2027 2
, 1120 1941 1143 1996 2
h 1207 1973 1259 2063 2
e 1268 1975 1317 2027 2
r 1331 1972 1391 2027 2
m 1456 1981 1547 2044 2
u 1559 1976 1613 2039 2
s 1617 1975 1656 2035 2
c 1667 1982 1721 2035 2
l 1736 1985 1754 2069 2
e 1774 1992 1832 2046 2
s 1846 1976 1889 2042 2
r 1963 1970 2014 2026 2
e 2015 1971 2063 2048 2
l 2081 1975 2100 2051 2
a 2114 1969 2164 2017 2
x 2174 1954 2225 2028 2
e 2233 1964 2284 2028 2
d 2284 1963 2335 2054 2
b 113 1742 169 1843 2
u 197 1741 255 1795 2
t 273 1747 313 1831 2
n 387 1754 442 1809 2
o 443 1746 484 1805 2
t 500 1743 534 1834 2
S 612 1750 650 1820 2
l 661 1746 678 1834 2
a 694 1744 748 1795 2
c 774 1750 820 1795 2
k 831 1756 892 1825 2
e 905 1753 957 1809 2
d 971 1740 1027 1833 2
, 1068 1724 1093 1771 2
h 1152 1753 1204 1839 2
e 1222 1748 1271 1802 2
r 1279 1754 1325 1801 2
l 1389 1746 1407 1837 2
i 1424 1741 1442 1815 2
p 1464 1706 1522 1806 2
s 1529 1742 1570 1805 2
, 1587 1712 1617 1764 2
c 1686 1754 1731 1806 2
l 1733 1747 1754 1832 2
o 1764 1745 1805 1793 2
s 1821 1745 1859 1797 2
e 1857 1741 1909 1790 2
d 1928 1735 1974 1824 2
, 2002 1721 2020 1766 2
a 2087 1744 2143 1801 2
t 2155 1743 2189 1830 2
t 2253 1751 2292 1837 2
h 2296 1745 2349 1832 2
e 2351 1740 2421 1815 2
v 110 1527 160 1602 2
e 177 1541 220 1593 2
r 242 1542 282 1591 2
y 302 1506 365 1591 2
e 451 1534 501 1605 2
d 520 1516 574 1604 2
g 592 1501 650 1572 2
e 664 1532 719 1593 2
o 832 1542 876 1589 2
f 884 1539 932 1623 2
P 986 1506 1041 1594 2
a 1050 1536 1107 1586 2
r 1113 1546 1154 1592 2
t 1176 1537 1220 1638 2
i 1243 1538 1262 1606 2
n 1282 1536 1338 1586 2
g 1355 1497 1412 1586 2
, 1455 1512 1475 1563 2
h 1539 1540 1594 1630 2
e 1605 1538 1652 1595 2
r 1665 1538 1717 1590 2
e 1771 1540 1817 1597 2
y 1829 1507 1878 1590 2
e 1896 1538 1944 1591 2
s 1956 1535 1992 1594 2
s 2072 1528 2105 1589 2
o 2120 1530 2174 1585 2
f 2186 1522 2229 1613 2
t 2247 1522 2298 1606 2
b 107 1319 170 1423 2
e 187 1328 232 1380 2
n 254 1323 307 1382 2
e 323 1318 371 1380 2
a 385 1323 449 1375 2
t 467 1328 512 1413 2
h 525 1323 597 1420 2
u 720 1323 771 1380 2
n 776 1323 828 1379 2
f 848 1321 894 1408 2
l 910 1322 930 1411 2
u 957 1315 1014 1372 2
t 1014 1321 1050 1408 2
t 1062 1325 1097 1410 2
e 1111 1325 1161 1381 2
r 1171 1328 1213 1376 2
i 1222 1325 1240 1397 2
n 1257 1327 1315 1378 2
g 1338 1274 1391 1379 2
e 1457 1333 1505 1396 2
y 1508 1304 1560 1386 2
e 1567 1325 1622 1386 2
l 1632 1324 1652 1404 2
i 1659 1330 1677 1401 2
d 1696 1322 1748 1405 2
s 1772 1319 1804 1376 2
, 1838 1304 1863 1352 2
h 1926 1302 1990 1410 2
e 2005 1307 2054 1365 2
r 2066 1314 2116 1363 2
h 2189 1312 2244 1413 2
o 2256 1308 2309 1364 2
o 2320 1311 2366 1368 2
d 2383 1307 2433 1394 2
e 2434 1302 2483 1366 2
d 2485 1300 2533 1394 2
g 128 1051 191 1161 2
a 215 1089 280 1154 2
z 309 1091 376 1157 2
e 403 1087 466 1156 2
t 549 1086 598 1186 2
u 623 1091 676 1149 2
r 687 1094 729 1146 2
n 739 1092 789 1150 2
e 798 1090 851 1151 2
d 861 1087 917 1186 2
w 1008 1096 1100 1157 2
i 1137 1095 1160 1193 2
t 1175 1081 1229 1182 2
h 1229 1095 1288 1194 2
i 1306 1099 1324 1171 2
n 1349 1102 1411 1156 2
, 1468 1064 1486 1130 2
t 1548 1094 1590 1182 2
o 1618 1097 1667 1155 2
t 1730 1101 1777 1179 2
h 1781 1100 1832 1186 2
e 1849 1103 1896 1161 2
c 1998 1100 2040 1160 2
a 2064 1099 2119 1154 2
l 2138 1095 2163 1185 2
m 2185 1098 2284 1150 2
p 139 833 198 939 2
l 212 876 230 963 2
a 264 873 325 931 2
c 346 873 394 932 2
e 416 876 469 942 2
a 590 874 654 933 2
t 681 872 725 962 2
t 805 864 860 975 2
h 869 882 927 975 2
e 946 879 999 941 2
c 1111 886 1166 942 2
e 1168 882 1211 946 2
n 1230 884 1284 942 2
t 1303 888 1348 973 2
e 1355 883 1414 942 2
r 1439 884 1487 940 2
o 1594 878 1646 937 2
f 1657 876 1697 955 2
h 1763 879 1823 963 2
e 1835 875 1889 930 2
r 1901 882 1958 931 2
b 2027 884 2081 973 2
e 2094 882 2138 935 2
i 2163 884 2181 955 2
n 2202 884 2259 934 2
g 2279 835 2332 929 2
. 2370 889 2388 908 2
S 167 652 211 755 2
h 236 661 297 754 2
e 307 651 360 713 2
s 430 650 469 725 2
e 476 655 535 718 2
n 548 660 606 715 2
s 628 657 662 725 2
e 682 660 745 722 2
d 759 647 815 740 2
r 880 650 942 707 2
a 953 650 1016 708 2
t 1035 661 1078 758 2
h 1087 653 1150 750 2
e 1163 652 1223 715 2
r 1236 666 1283 710 2
t 1344 648 1396 756 2
h 1404 661 1462 753 2
a 1476 662 1539 713 2
n 1539 654 1597 710 2
f 1661 655 1715 752 2
e 1736 658 1785 722 2
l 1804 658 1826 751 2
t 1850 655 1903 750 2
h 1955 669 2011 752 2
i 2021 666 2041 736 2
m 2060 653 2154 716 2
a 138 424 210 505 2
w 242 429 334 492 2
a 367 424 443 491 2
k 463 430 535 510 2
e 548 432 616 503 2
n 639 426 701 497 2
b 822 430 892 524 2
e 920 432 979 496 2
s 1032 431 1076 497 2
i 1104 436 1126 517 2
d 1164 432 1223 519 2
e 1255 429 1317 502 2
h 1420 436 1491 532 2
e 1504 432 1572 508 2
r 1593 443 1650 501 2
. 1673 457 1691 477 2
W 133 3310 233 3412 3
h 263 3304 317 3408 3
e 342 3308 394 3364 3
n 419 3314 480 3370 3
h 587 3315 644 3414 3
e 664 3311 718 3378 3
t 797 3315 841 3410 3
u 861 3306 908 3368 3
r 933 3316 988 3367 3
n 1009 3310 1071 3378 3
e 1081 3308 1132 3372 3
d 1140 3299 1186 3403 3
t 1265 3305 1309 3397 3
o 1325 3305 1376 3355 3
l 1472 3313 1494 3396 3
o 1520 3309 1564 3361 3
o 1587 3300 1632 3359 3
k 1654 3303 1717 3383 3
a 1773 3298 1832 3356 3
t 1848 3301 1887 3387 3
h 1962 3298 2019 3394 3
e 2036 3297 2087 3358 3
r 2099 3303 2156 3354 3
, 2188 3281 2203 3318 3
s 2259 3291 2297 3367 3
h 2319 3298 2379 3380 3
e 2385 3291 2441 3351 3
h 125 3104 182 3210 3
o 187 3099 237 3155 3
p 251 3067 308 3158 3
e 319 3103 368 3173 3
d 388 3086 432 3179 3
h 467 3086 529 3174 3
e 537 3080 592 3150 3
w 682 3090 767 3152 3
o 777 3088 828 3146 3
u 842 3068 896 3141 3
l 913 3085 932 3170 3
d 948 3079 1000 3162 3
s 1050 3082 1086 3138 3
e 1102 3083 1156 3137 3
e 1165 3078 1225 3141 3
: 1247 3086 1273 3152 3
H 150 2865 226 2990 3
e 262 2880 309 2944 3
r 332 2888 376 2942 3
h 471 2880 530 2981 3
a 544 2876 594 2930 3
i 612 2876 628 2956 3
r 646 2884 699 2929 3
: 728 2890 746 2952 3

t 805 2878 853 2964 3
h 868 2881 919 2965 3
e 926 2875 981 2939 3
u 1064 2861 1121 2930 3
t 1140 2866 1186 2960 3
t 1201 2875 1240 2949 3
e 1264 2867 1320 2926 3
r 1326 2876 1380 2918 3
d 1478 2850 1529 2954 3
a 1543 2845 1601 2898 3
r 1613 2852 1671 2899 3
k 1692 2852 1762 2933 3
o 1851 2851 1892 2904 3
f 1915 2851 1954 2934 3
s 2032 2841 2079 2911 3
t 2092 2846 2132 2928 3
a 2142 2843 2201 2899 3
r 2204 2847 2258 2892 3
l 2286 2842 2307 2932 3
e 2321 2842 2379 2907 3
s 2396 2840 2432 2896 3
s 2437 2822 2476 2893 3
n 144 2664 212 2736 3
i 227 2660 248 2735 3
g 277 2633 339 2713 3
h 347 2665 415 2748 3
t 434 2662 474 2747 3
s 591 2655 633 2718 3
p 660 2624 712 2713 3
i 743 2653 762 2725 3
l 790 2655 813 2731 3
l 820 2653 842 2744 3
i 867 2650 892 2719 3
n 925 2647 981 2695 3
g 1001 2612 1055 2694 3
a 1152 2646 1223 2704 3
c 1240 2634 1292 2690 3
r 1302 2644 1349 2692 3
o 1360 2641 1403 2687 3
s 1422 2639 1456 2695 3
s 1471 2634 1503 2690 3
t 1588 2622 1634 2719 3
h 1656 2629 1707 2715 3
e 1720 2624 1770 2684 3
b 1847 2626 1909 2713 3
l 1924 2629 1942 2714 3
u 1973 2615 2030 2675 3
e 2053 2617 2104 2678 3
s 2160 2616 2198 2680 3
i 2234 2618 2251 2693 3
l 2281 2612 2309 2705 3
k 2321 2617 2384 2697 3
U 134 2441 203 2526 3
n 200 2437 257 2492 3
d 267 2429 307 2511 3
e 309 2435 351 2484 3
r 352 2440 397 2492 3
s 415 2439 448 2498 3
h 455 2442 507 2521 3
e 520 2445 563 2492 3
e 573 2443 613 2491 3
t 630 2440 668 2528 3
. 688 2446 704 2463 3
H 153 2204 222 2324 3
e 244 2217 294 2288 3
r 323 2232 365 2283 3
f 427 2216 471 2316 3
a 490 2218 554 2279 3
c 562 2219 608 2277 3
e 617 2215 669 2279 3
: 682 2205 702 2259 3
p 765 2170 826 2270 3
a 836 2206 895 2259 3
l 905 2212 925 2294 3
e 938 2206 986 2273 3
a 1068 2200 1132 2255 3
s 1154 2195 1185 2254 3
s 1278 2180 1313 2254 3
p 1332 2160 1390 2255 3
r 1400 2199 1446 2249 3
i 1469 2200 1487 2264 3
n 1509 2199 1566 2249 3
g 1583 2159 1645 2248 3
s 1717 2192 1750 2259 3
n 1771 2185 1827 2241 3
o 1844 2186 1885 2237 3
w 1899 2188 1982 2236 3
, 2007 2164 2023 2206 3
g 2087 2152 2143 2232 3
l 2153 2178 2178 2263 3
o 2195 2171 2240 2227 3
w 2258 2172 2326 2226 3
i 2347 2178 2363 2246 3
n 2383 2172 2438 2230 3
g 2456 2140 2511 2219 3
w 165 1977 242 2042 3
i 263 1981 284 2063 3
t 297 1973 343 2059 3
h 346 1967 413 2067 3
l 503 1980 527 2075 3
i 538 1980 559 2050 3
g 586 1944 648 2035 3
h 644 1967 709 2061 3
t 715 1981 753 2069 3
s 846 1978 878 2043 3
t 894 1974 935 2075 3
o 957 1975 1000 2035 3
l 1033 1984 1051 2065 3
e 1071 1975 1134 2047 3
n 1145 1985 1203 2047 3
f 1282 1972 1336 2068 3
r 1340 1976 1390 2030 3
o 1394 1966 1443 2026 3
m 1457 1966 1533 2027 3
t 1605 1966 1657 2062 3
h 1660 1969 1712 2063 3
e 1717 1970 1773 2036 3
m 1846 1958 1951 2038 3
o 1977 1965 2029 2017 3
o 2052 1961 2107 2016 3
n 2126 1966 2196 2022 3
. 2228 1987 2243 2005 3
H 156 1749 231 1862 3
e 248 1748 302 1824 3
r 308 1754 369 1807 3
b 426 1752 491 1854 3
o 510 1750 560 1806 3
d 590 1750 644 1841 3
y 669 1723 730 1807 3
: 743 1762 765 1835 3
s 855 1752 893 1829 3
u 904 1758 955 1808 3
g 968 1727 1030 1812 3
g 1045 1722 1100 1810 3
e 1118 1758 1174 1820 3
s 1194 1754 1233 1823 3
t 1250 1760 1297 1855 3
i 1319 1766 1337 1831 3
v 1361 1756 1413 1814 3
e 1418 1758 1475 1822 3
c 1569 1744 1620 1819 3
u 1629 1745 1693 1811 3
r 1714 1754 1775 1811 3
v 1796 1752 1849 1813 3
a 1853 1739 1916 1802 3
t 1932 1746 1975 1838 3
u 1995 1743 2057 1803 3
r 2077 1752 2128 1803 3
e 2139 1745 2188 1807 3
s 2205 1741 2241 1805 3
b 146 1529 212 1643 3
e 227 1526 280 1590 3
n 285 1534 345 1593 3
e 357 1534 409 1597 3
a 419 1534 481 1590 3
t 490 1535 532 1623 3
h 545 1525 610 1627 3
t 685 1524 747 1629 3
h 736 1524 799 1626 3
e 804 1517 873 1588 3
c 961 1520 1017 1590 3
o 1021 1521 1071 1576 3
v 1087 1524 1143 1584 3
e 1148 1524 1203 1586 3
r 1209 1529 1251 1577 3
l 1264 1525 1290 1613 3
e 1310 1521 1366 1585 3
t 1385 1527 1427 1608 3
, 1477 1503 1495 1539 3
a 1560 1509 1624 1583 3
l 1634 1523 1658 1608 3
s 1670 1506 1710 1581 3
o 1731 1506 1781 1566 3
o 1853 1499 1902 1558 3
f 1925 1506 1972 1596 3
s 2051 1503 2091 1569 3
i 2112 1501 2130 1588 3
l 2156 1502 2177 1587 3
k 2188 1504 2257 1588 3
, 2280 1482 2296 1512 3
e 162 1316 227 1389 3
m 259 1314 342 1385 3
b 368 1307 423 1406 3
l 441 1314 463 1412 3
a 495 1295 562 1364 3
z 582 1297 642 1369 3
o 661 1305 716 1363 3
n 724 1310 791 1370 3
e 807 1306 872 1368 3
d 895 1297 953 1390 3
w 1052 1298 1124 1359 3
i 1152 1304 1174 1397 3
t 1193 1302 1243 1385 3
h 1254 1293 1313 1391 3
a 1400 1303 1464 1354 3
f 1527 1288 1577 1385 3
a 1594 1293 1651 1345 3
i 1665 1293 1684 1362 3
r 1705 1294 1756 1343 3
l 1770 1293 1794 1378 3
y 1817 1257 1875 1346 3
e 1950 1275 2015 1343 3
m 2029 1269 2101 1338 3
b 2120 1275 2167 1361 3
r 2184 1274 2234 1327 3
o 2247 1263 2290 1325 3
i 2304 1277 2322 1356 3
d 2345 1268 2390 1349 3
e 2397 1268 2445 1330 3
d 2452 1261 2500 1355 3
p 137 1042 194 1164 3
a 209 1085 274 1145 3
i 294 1089 322 1194 3
r 352 1090 410 1144 3
o 509 1075 560 1138 3
f 575 1070 616 1160 3
w 702 1075 784 1129 3
h 781 1075 839 1168 3
i 856 1072 877 1155 3
t 898 1074 943 1162 3
e 963 1073 1019 1142 3
c 1130 1072 1188 1145 3
r 1206 1077 1261 1131 3
a 1276 1071 1341 1129 3
n 1352 1075 1406 1133 3
e 1426 1072 1487 1133 3
s 1511 1068 1554 1139 3
, 1610 1055 1629 1099 3
t 1690 1064 1730 1160 3
h 1743 1057 1799 1158 3
e 1808 1058 1868 1123 3
i 1895 1070 1915 1141 3
r 1927 1069 1982 1120 3
t 2087 1056 2126 1158 3
h 2139 1051 2206 1150 3
r 2210 1049 2258 1103 3
o 2276 1042 2315 1099 3
a 2330 1042 2382 1098 3
t 2387 1049 2428 1139 3
s 2449 1038 2480 1112 3
c 114 867 179 959 3
r 180 871 227 925 3
i 227 871 245 952 3
m 264 865 353 927 3
s 381 861 417 932 3
o 435 867 491 921 3
n 509 863 571 922 3
w 657 860 741 921 3
i 756 857 776 943 3
t 794 850 838 936 3
h 848 846 926 945 3
m 1008 842 1110 914 3
a 1124 845 1179 906 3
t 1198 843 1233 941 3
i 1262 850 1285 928 3
n 1311 845 1376 912 3
g 1393 817 1457 903 3
f 1554 837 1606 935 3
r 1622 841 1676 898 3
e 1689 842 1748 904 3
n 1764 842 1821 903 3
z 1849 845 1907 905 3
y 1948 808 2012 899 3
, 2057 823 2075 862 3
d 2146 832 2200 924 3
a 2228 834 2286 892 3
n 2300 833 2352 891 3
c 2362 825 2412 890 3
i 2428 825 2449 905 3
n 117 647 182 718 3
g 204 614 271 712 3
a 412 644 487 712 3
n 505 649 567 704 3
d 599 631 651 729 3
d 779 630 835 736 3
u 842 625 923 690 3
e 939 627 988 691 3
l 1001 624 1024 726 3
i 1041 634 1067 709 3
n 1089 636 1151 690 3
g 1165 603 1223 689 3
i 1305 639 1326 723 3
n 1335 633 1396 696 3
m 1474 626 1566 694 3
i 1574 633 1595 713 3
d 1610 621 1660 710 3
a 1692 625 1754 685 3
i 1763 631 1784 694 3
r 1806 628 1868 679 3
, 1905 612 1922 648 3
a 2008 622 2074 684 3
g 2090 584 2152 674 3
a 2173 603 2240 666 3
i 2239 612 2264 690 3
n 2283 612 2341 670 3
s 2362 601 2395 666 3
t 2419 611 2460 698 3

```
a 138 442 209 503 3
f 274 424 327 532 3
i 344 436 366 507 3
e 378 424 439 489 3
l 455 422 476 517 3
```

```
d 491 420 535 509 3
o 641 413 693 482 3
f 712 412 750 511 3
g 824 379 890 485 3
o 906 408 954 476 3
```

```
l 986 405 1010 507 3
d 1041 402 1097 506 3
```

B. Source Codes

B..1 Tess2Speech Mobile Application

Base Path = Tess2Speech/

Path = Tess2Speech/build.gradle

```
// Top-level build file where you can add configuration options common to all sub-projects/modules.

buildscript {
    repositories {
        jcenter()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:1.5.0'
        // NOTE: Do not place your application dependencies here; they belong
        // in the individual module build.gradle files
    }
}

allprojects {
    repositories {
        jcenter()
    }
}
```

Path = Tess2Speech/settings.gradle

```
include ':tess2speech'
include ':libraries:android-pdfview-master:android-pdfview'
include ':libraries:tess-two'
include ':libraries:gesture-imageview'
include ':libraries:cropper-master:cropper'
include ':libraries:android-file-picker_activity'
```

Path = Tess2Speech/libraries

This folder contains different open-source libraries used in developing Tess2Speech:

1. android-file-picker_activity by Anders Kaloer [41]
2. android-pdfview-master by Joan Zapata [38]
3. cropper-master by ArthurHub [37]
4. epublib by Paul Siegman [40]
5. gesture-imageview by Jason Polites [36]
6. tess-two by Robert Theis [35]

Path = Tess2Speech/tess2Speech/build.gradle

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 23
    buildToolsVersion '23.0.2'
    defaultConfig {
        applicationId "anteraaron.tess2speech"
        minSdkVersion 16
        targetSdkVersion 23
        versionCode 1
        versionName "1.0"
        signingConfig signingConfigs.release
    }
    buildTypes {
        release {
            debuggable false
            minifyEnabled false
            proguardFiles
                getDefaultProguardFile('
                proguard-android.txt'), '
                proguard-rules.pro'
            signingConfig signingConfigs.
                release
        }
    }
    productFlavors {
    }
}

//repositories {
 //    mavenCentral()
//}

dependencies {
    compile fileTree(include: ['*.jar'], dir:
        'libs')
    testCompile 'junit:junit:4.12'
    compile project(':libraries:tess-two')
    compile project(':libraries:gesture-
        imageview')
    compile project(':libraries:android-
        pdfview-master:android-pdfview')
    compile project(':libraries:cropper-master
        :cropper')
    compile project(':libraries:android-file-
        picker_activity')
    compile 'com.android.support:appcompat-v7
        :23.1.1'
    compile 'com.android.support:design
        :23.1.1'
    compile 'com.android.support:support-v4
        :23.1.1'
    compile 'commons-io:commons-io:2.4'
    compile files('E:/Aaron/Dropbox/Workspace/
        Tess2Speech/libraries/epublib/epublib
        -core-latest.jar')
    compile files('E:/Aaron/Dropbox/Workspace/
        Tess2Speech/libraries/epublib/slf4j-
        android-1.6.1-RC1.jar')
}
```

Path = Tess2Speech/tess2Speech/proguard-rules.pro

```
# Add project specific ProGuard rules here.
# By default, the flags in this file are
    appended to flags specified
# in E:\Aaron\Android\sdk/tools/proguard/
    proguard-android.txt
# You can edit the include path and order by
    changing the proguardFiles
# directive in build.gradle.
#
# For more details, see
#   http://developer.android.com/guide/
    developing/tools/proguard.html

# Add any project specific keep options here:

# If your project uses WebView with JS,
    uncomment the following
# and specify the fully qualified class name
    to the JavaScript interface
# class:
#-keepclassmembers class fqcn.of.javascript.
    interface.for.webview {
#   public *;
#}
```

Path = Tess2Speech/tess2Speech/src/main/AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!--
Copyright 2016 Anter Aaron M. Custodio

 Licensed under the Apache License, Version
      2.0 (the "License");
 you may not use this file except in
      compliance with the License.
 You may obtain a copy of the License at

 http://www.apache.org/licenses/LICENSE-2.0

 Unless required by applicable law or agreed
      to in writing, software
 distributed under the License is distributed
      on an "AS IS" BASIS,
 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
      either express or implied.
 See the License for the specific language
      governing permissions and
 limitations under the License.
-->
<manifest xmlns:android="http://schemas.
    android.com/apk/res/android"
  package="anteraaron.tess2speech">

  <uses-permission android:name="android.
    permission.READ_EXTERNAL_STORAGE" />
  <uses-permission android:name="android.
    permission.WRITE_EXTERNAL_STORAGE" />
  <uses-permission android:name="android.
    permission.READ_INTERNAL_STORAGE" />
  <uses-permission android:name="android.
    permission.WRITE_INTERNAL_STORAGE" />
  <uses-permission android:name="android.
    permission.READ_USER_DICTIONARY" />
  <uses-permission android:name="android.
    permission.READ_PHONE_STATE" />
  <uses-feature
    android:name="android.hardware.camera"
    android:required="true" />

  <application
    android:allowBackup="true"
    android:icon="@mipmap/logo"
    android:label="@string/app_name"
    android:largeHeap="true"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">

    <!-- Splash Screen Activity -->
    <activity android:name="anteraaron.
     tess2speech.Activity_SplashScreen"
      android:theme="@style/AppTheme.
     NoActionBar"
      android:screenOrientation="portrait">
      <intent-filter>
        <action android:name="android.intent.
     action.MAIN" />
        <category android:name="android.intent
     .category.LAUNCHER" />
      </intent-filter>
    </activity>

    <!-- Main Activity -->
    <activity
      android:name="anteraaron.tess2speech.
     Activity_Main"
      android:label="@string/app_name"
      android:theme="@style/AppTheme.
     NoActionBar">
    </activity>

    <!-- Rotate Activity -->
    <activity android:name="anteraaron.
     tess2speech.Activity_Rotate"
      android:label="Rotate Image"
      android:screenOrientation="portrait">
    </activity>

    <!-- Preference Activity -->
    <activity android:name="anteraaron.
     tess2speech.Activity_Settings">
    </activity>

    <!-- Crop Activity -->
    <activity android:name="anteraaron.
     tess2speech.Activity_Crop"
      android:label="Crop Image">
    </activity>

    <!-- File Browser-->
    <activity android:name="kaloer.filepicker.
     FilePickerActivity"
      android:icon="@mipmap/logo"
      android:enabled="true"
      android:label="Browse File"
      android:exported="true">
      <intent-filter>
        <action android:name="android.intent.
     action.GET_CONTENT" />

        <category android:name="android.intent
     .category.DEFAULT" />
        <category android:name="android.intent
     .category.OPENABLE" />

        <data android:mimeType="*/*" />
      </intent-filter>
    </activity>

  </application>

</manifest>
```

Path = Tess2Speech/tess2Speech/src/main/java/anteraaron/tess2speech/Activity_Crop.java

```
package anteraaron.tess2speech;

import android.content.Intent;
import android.net.Uri;
import android.support.v7.app.
    AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import com.theartofdev.edmodo.cropper.
    CropImageView;
import java.io.File;

/**
 * Activity for cropping Images using Edmodo
     Cropper (see... https://github.com/
     ArthurHub/Android-Image-Cropper/wiki).
 * @author Anter Aaron M. Custodio
 * @version 2016.3.22
 */
public class Activity_Crop extends
    AppCompatActivity {

 @Override
 protected void onCreate(final Bundle
     savedInstanceState) {
  super.onCreate(savedInstanceState);
  setContentView(R.layout.activity_crop);
  //Retrieve Uri of the passed image from
     Activity_Main.
  final Uri imageUri = getIntent().
     getParcelableExtra("imageUri");
  //Create Edmodo Cropper ImageView.
  final CropImageView imageView_crop = (
     CropImageView)findViewById(R.id.
     CropImageView);
  imageView_crop.setFixedAspectRatio(false);
  imageView_crop.setGuidelines(CropImageView.
     Guidelines.ON_TOUCH);
  imageView_crop.setImageUriAsync(imageUri);
  //Create confirm crop button
  final Button button_crop_confirm = (Button)
     findViewById(R.id.button_crop_confirm);
  button_crop_confirm.setOnClickListener(new
     View.OnClickListener() {
   @Override
   public void onClick(View v) {
    //Calls an AsyncTask that saves the
     cropped image to a file
    new Task_SaveCroppedImage(Activity_Crop.
     this, new Task_SaveCroppedImage.
     AsyncResponse() {
     @Override
     public void processFinished(File file) {
      //Return file after saving the cropped
     image to get file path.
      final Uri fileUri = Uri.fromFile(file);
      //Add the cropped image file to gallery.
      Libraries.galleryAddPic(fileUri);
      //Pass the file uri back to
     Activity_Main with OK result.
      final Intent intent = new Intent();
      intent.putExtra("imageUri", fileUri);
      setResult(RESULT_OK, intent);
      //Clear imageView
      imageView_crop.clearImage();
      imageView_crop.destroyDrawingCache();
      imageView_crop.removeAllViews();
      finish();
     }
    }).execute(imageView_crop.getCroppedImage
     ());//the parameter for doInBackground in
     Task_Crop is the cropped image.
   }
  });
  //Create cancel crop button.
  final Button button_crop_cancel = (Button)
     findViewById(R.id.button_crop_cancel);
  button_crop_cancel.setOnClickListener(new
     View.OnClickListener() {
   @Override
   public void onClick(View v) {
    //Returns cancel result to Activity_Main.
    setResult(RESULT_CANCELED);
    finish();
   }
  });
 }
}
```

Path = Tess2Speech/tess2Speech/src/main/java/anteraaron/tess2speech/Activity_Main.java

```
/*
 * Copyright 2016 Anter Aaron M. Custodio
 *
 * Licensed under the Apache License, Version
     2.0 (the "License");
 * you may not use this file except in
     compliance with the License.
 * You may obtain a copy of the License at
 *
 *      http://www.apache.org/licenses/LICENSE
     -2.0
 *
 * Unless required by applicable law or agreed
     to in writing, software
 * distributed under the License is
     distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY
     KIND, either express or implied.
 * See the License for the specific language
     governing permissions and
 * limitations under the License.
 */

package anteraaron.tess2speech;

import android.app.Activity;
import android.content.Intent;
import android.graphics.Bitmap;
import android.media.AudioManager;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.SystemClock;
import android.support.v4.app.
    FragmentTransaction;
import android.support.v7.app.ActionBar;
import android.os.Bundle;
import android.support.v7.app.
    AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Toast;
import com.joanzapata.pdfview.PDFView;
import com.polites.android.GestureImageView;
import java.io.File;
import java.net.MalformedURLException;
import kaloer.filepicker.FilePickerActivity;

/**
 * Main Activity for Tess2Speech.
 * Includes displaying of Fragments, Toolbar,
     Logo, calling Intents, Intent Results
     etc.
 *
 * @author Anter Aaron M. Custodio
 * University of the Philippines Manila
 * B.S. Computer Science Student
 * @since 2016.3.22
 */
public class Activity_Main extends
```

```
    AppCompatActivity{
/**
     * An inner class that contains Constants
         for activity Results.
     */
public static final class Activity_Result {
 public static final int
    CAPTURE_IMAGE_ACTIVITY_REQUEST_CODE =
    100; //Request code for capture image
 public static final int
    BROWSE_IMAGE_ACTIVITY_REQUEST_CODE = 200;
     //Request code for browse image
 public static final int
    BROWSE_MULTIPLE_IMAGE_ACTIVITY_REQUEST_CODE
     = 201; //Request code for browse
    multiple image
 public static final int
    BROWSE_PDF_ACTIVITY_REQUEST_CODE = 202;
    //Request code for browse pdf
 public static final int
    BROWSE_PDF_CONVERT_TO_IMAGE_ACTIVITY_REQUEST_CODE
     = 203; //Request code for browse pdf
 public static final int
    BROWSE_EPUB_ACTIVITY_REQUEST_CODE = 204;
    //Request code for browse pdf
 public static final int
    ROTATE_BITMAP_ACTIVITY_REQUEST_CODE =
    300; //Request code for rotate image
 public static final int
    CROP_IMAGE_ACTIVITY_REQUEST_CODE = 400;
    //Request code for crop image
}

/**
     * An inner class that contains Constants
         for Application States.
     */
public static final class State {
 public static final int IS_CANVAS = 0; //
    Application is displaying the converted
    text.
 public static final int IS_IMAGE = 1; //
    Application is displaying the input image
 public static final int IS_PDF = 2; //
    Application is displaying PDF
 public static final int IS_EPUB = 3; //
    Application is displaying EPUB
 public static final int IS_TEXT_FROM_CANVAS
    = 4; //Application is displaying the
    converted text from canvas.
 public static final int IS_TEXT_FROM_IMAGE =
     5; //Application is displaying the
    converted text from image fragment.
 public static final int IS_TEXT_FROM_PDF =
    6; //Application is displaying the
    converted text from PDF fragment.
 public static final int IS_TEXT_FROM_EPUB =
    7; //Application is displaying the
    converted text from Epub fragment.
}

public static Button button_play, button_text
    ;

private AsyncTask<Boolean, Boolean, Boolean>
    save;
private AsyncTask<String, Void, Void> speak;

private static Uri imageUri, pdfUri, epubUri;
     //Image, PDF or, Epub uri being
    displayed
public static int state; //Determine the
    current state or Fragment displayed in
    the App

private long lastClickTime; //variable that
    saves the last clicked time.

@Override
protected void onCreate(final Bundle
    savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.activity_main);

 final Toolbar toolbar = (Toolbar)
    findViewById(R.id.toolbar);
 toolbar.setNavigationIcon(null);
 setSupportActionBar(toolbar);

 //logo in action bar
 final ActionBar ab = getSupportActionBar();
 if (ab != null) {
  ab.setLogo(R.mipmap.logo);
  ab.setDisplayUseLogoEnabled(true);
  //remove back button in toolbar
  ab.setDisplayHomeAsUpEnabled(false);
  ab.setHomeButtonEnabled(false);
 }
 //Initialize variables
 state = State.IS_CANVAS;
 this.lastClickTime = 0;

 //always change the music volume when
    pressing button
 setVolumeControlStream(AudioManager.
    STREAM_MUSIC);

 //Cast UIs
 createUI();
}

@Override
public void onStart(){
 //construct libraries
 new Libraries(this);
 super.onStart();
}

@Override
public boolean onCreateOptionsMenu(final Menu
     menu){
 // Inflate the menu; this adds items to the
    action bar if it is present.
 getMenuInflater().inflate(R.menu.menu_main,
    menu);
 final MenuItem saveText = menu.findItem(R.id
    .saveText);
 final MenuItem savePdf = menu.findItem(R.id.
    savePdf);
 final MenuItem saveAudio = menu.findItem(R.
    id.saveAudio);

 if (state == State.IS_CANVAS || state ==
    State.IS_IMAGE || state == State.IS_PDF
    || state == State.IS_EPUB) {
  //Disable save as .txt menu
  saveText.setEnabled(false);
  saveText.setTitle("Save as .txt (Convert
    text first)");
  //Disable save as .pdf menu
  savePdf.setEnabled(false);
  savePdf.setTitle("Save as .pdf (Convert
    text first)");
  //Disable save audio menu
  saveAudio.setEnabled(false);
  saveAudio.setTitle("Save Audio (Convert
    text first)");
 } else {
  //Enable save as .txt menu
  saveText.setEnabled(true);
  saveText.setTitle("Save as .txt");
  //Enable save as .pdf menu
  savePdf.setEnabled(true);
  savePdf.setTitle("Save as .pdf");
  //Enable save audio menu
  saveAudio.setEnabled(true);
  saveAudio.setTitle("Save Audio");
 }
 return true;
}

@Override
public boolean onOptionsItemSelected(final
    MenuItem item) {
 /*Handle action bar item clicks here. The
    action bar will
     automatically handle clicks on the
         Home/Up button, so long
     as you specify a parent activity in
         AndroidManifest.xml.*/

 //Stop text to speech and tesseract
 Libraries.tts.stop();
 Libraries.tesseract.clear();
 button_play.setText(R.string.button_play);

 final int id = item.getItemId();

 switch (id) {
  //Toolbar camera icon on Click
  case R.id.camera: {
   //if camera button is clicked
   final Intent_Camera camera = new
    Intent_Camera(this);
   final Intent intent = camera.takePicture()
    ;

   if (intent != null) {
    //Start intent activity for camera
    startActivityForResult(intent,
    Activity_Result.
    CAPTURE_IMAGE_ACTIVITY_REQUEST_CODE);
   } else {
    //Error notification
    Toast.makeText(this, "An error has
    occured", Toast.LENGTH_SHORT).show();
   }
```

```
 break;
}

//Toolbar browse image icon on Click
case R.id.browse: {
 //If browse image button is clicked
 final Intent intent = Intent_BrowseFiles.
  browseImage();
 if (intent != null) {
  //Start intent activity for browsing file
  startActivityForResult(Intent.
  createChooser(intent, "Complete action
  using"), Activity_Result.
  BROWSE_IMAGE_ACTIVITY_REQUEST_CODE);
 } else {
  //Error notification
  Toast.makeText(this, "An error has
  occured!", Toast.LENGTH_SHORT).show();
 }
 break;
}

//Toolbar browse pdf on Click
case R.id.browsePdf: {
 //If browse PDF button is clicked
 final Intent intent = Intent_BrowseFiles.
  browsePdf();
 if(intent != null) {
  //Start intent activity for browsing pdf
  startActivityForResult(Intent.
  createChooser(intent, "Complete action
  using"), Activity_Result.
  BROWSE_PDF_ACTIVITY_REQUEST_CODE);
 } else {
  //Error notification
  Toast.makeText(this, "An error has
  occured!", Toast.LENGTH_SHORT).show();
 }
 break;
}

//Toolbar browse epub on Click
case R.id.browseEpub: {
 //If browse epub button is clicked
 final Intent intent = Intent_BrowseFiles.
  browseEpub();
 if(intent != null) {
  //Start intent activity for browsing pdf
  startActivityForResult(Intent.
  createChooser(intent, "Complete action
  using"), Activity_Result.
  BROWSE_EPUB_ACTIVITY_REQUEST_CODE);
 } else {
  //Error notification
  Toast.makeText(this, "An error has
  occured!", Toast.LENGTH_SHORT).show();
 }
 break;
}

//Toolbar save as .txt on click
case R.id.saveText: {
 //Launch SaveText AsyncTask
 new Task_SaveText(this);
 break;
}

//Toolbar save audio on click
case R.id.saveAudio: {
 //Launch SaveAudio AsyncTask
 new Task_SaveAudio(this);
 break;
}

//Toolbar save PDF on click
case R.id.savePdf: {
 //Launch SavePdf AsyncTask
 new Task_SavePdf(this);
 //use fragment_pdf as Fragment
 state = State.IS_PDF;
 //Display PDFViewer Fragment
 displayFragment();
 break;
}

//Toolbar Convert image to searchable PDF
  on click
case R.id.imageToPdf: {
 //User has an option to choose multiple
  images and compile them to one pdf
 final Intent intent = Intent_BrowseFiles.
  browseMultipleImages();
 if(intent != null) {
  //Start intent activity for browsing file
  startActivityForResult(Intent.
  createChooser(intent, "Complete action
  using"), Activity_Result.
  BROWSE_MULTIPLE_IMAGE_ACTIVITY_REQUEST_CODE
  );
 } else {
  //Error notification
  Toast.makeText(this, "An error has
  occured!", Toast.LENGTH_SHORT).show();
 }
 break;
}

//Toolbar Convert PDF to image on click
case R.id.pdfToImage: {
 //Choose which PDF to convert
 final Intent intent = Intent_BrowseFiles.
  browsePdf();
 if(intent != null) {
  //Start intent activity for browsing pdf
  startActivityForResult(Intent.
  createChooser(intent, "Complete action
  using"), Activity_Result.
  BROWSE_PDF_CONVERT_TO_IMAGE_ACTIVITY_REQUEST_CODE
  );
 } else {
  //Error notification
  Toast.makeText(this, "An error has
  occured!", Toast.LENGTH_SHORT).show();
 }
 break;
}

//Toolbar Settings on click
case R.id.settings: {
 //Start the preference screen
 final Intent intent = new Intent(this,
  Activity_Settings.class);
 startActivityForResult(intent, 0);
 break;
}

//Toolbar Exit on Click
case R.id.exit: {
 //Exit activity
 finish();
 break;
}

//default
default: {
 break;
}

}
return super.onOptionsItemSelected(item);
}

/**
    * Create UI for the Application.
    */
public void createUI() {
 //Create imageView for PDFs.
 Libraries.pdfView = (PDFView) findViewById(R
    .id.pdfView);
 //Create imageView for InputImages.
 Libraries.imageView = (GestureImageView)
    findViewById(R.id.imageView);

 //If converted text button is pressed.
 button_text = (Button)findViewById(R.id.
    button_text);
 button_text.setOnClickListener(new View.
    OnClickListener() {
  @Override
  public void onClick(View v) {

   //double-clicking prevention, using
   threshold of 1000 ms
   if (SystemClock.elapsedRealtime() -
   lastClickTime < 1000) {
    return;
   }
   lastClickTime = SystemClock.
   elapsedRealtime();

   //Check if the user wants to retain the
   previous converted text.
   if (button_text.getText().equals(
   getResources().getString(R.string.
   button_text))) {
    Fragment_ConvertedText.checkAppend();

    switch (state) {
     //Convert image to text, but do not play
    .
     case State.IS_IMAGE: {
      new ToText(Activity_Main.this).execute(
    imageUri, false);
      break;
     }

     //In PDF Fragment
     case State.IS_PDF: {
```

```
        Libraries.pdfToText(false);
        break;
       }

       //In EPUB Fragment
       case State.IS_EPUB: {
        new Task_WebViewToText(Activity_Main.
      this).execute(false);
        break;
       }

       //In Canvas
       case State.IS_CANVAS: {
        //Save text in canvas but do not play.
        new Task_SaveCanvas(Activity_Main.this)
      .execute(false);
        break;
       }

       default:
        break;
      }

      //Set current state
      switch (state) {
       case State.IS_CANVAS:
        state = State.IS_TEXT_FROM_CANVAS;
        break;
       case State.IS_IMAGE:
        state = State.IS_TEXT_FROM_IMAGE;
        break;
       case State.IS_PDF:
        state = State.IS_TEXT_FROM_PDF;
        break;
       case State.IS_EPUB:
        state = State.IS_TEXT_FROM_EPUB;
        break;
       default:
        break;
      }
      //Display Text Fragment
      displayFragment();

      button_text.
     setCompoundDrawablesWithIntrinsicBounds
     (0, R.mipmap.ic_action_eye_closed, 0, 0);
      button_text.setText(R.string.
     button_text_back);
      Fragment_ConvertedText.editText.
     requestFocus();

     } else {
      //Set current state
      switch (state) {
       case State.IS_TEXT_FROM_CANVAS:
        state = State.IS_CANVAS;
        break;
       case State.IS_TEXT_FROM_IMAGE:
        state = State.IS_IMAGE;
        break;
       case State.IS_TEXT_FROM_PDF:
        state = State.IS_PDF;
        break;
       case State.IS_TEXT_FROM_EPUB:
        state = State.IS_EPUB;
        break;
       default:
        break;
      }
      //Display Canvas Fragment/InputImage
     Fragment.
      displayFragment();
      button_text.
     setCompoundDrawablesWithIntrinsicBounds
     (0, R.mipmap.ic_action_eye_open, 0, 0);
      button_text.setText(R.string.button_text)
     ;
      Fragment_DrawingCanvas.drawingView.
     requestFocus();
     }
     //refresh activity menu
     invalidateOptionsMenu();
    }
   });

   //If back button in pdf is pressed.
   Fragment_PdfViewer.button_pdf_back = (Button
      ) findViewById(R.id.button_pdf_back);
   Fragment_PdfViewer.button_pdf_back.
      setOnClickListener(new View.
      OnClickListener() {
    @Override
    public void onClick(View v) {

     //double-clicking prevention, using
      threshold of 1000 ms
     if (SystemClock.elapsedRealtime() -
      lastClickTime < 1000) {
      return;
     }
     lastClickTime = SystemClock.
      elapsedRealtime();

     //Cancel speaking when back button is
      pressed.
     if (save != null) {
      save.cancel(true);
     } else if (speak != null) {
      speak.cancel(true);
     }
     Libraries.tts.stop();
     Libraries.tesseract.clear();
     button_play.setText(R.string.button_play);

     //set state
     state = State.IS_CANVAS;

     Libraries.imageView.setImageDrawable(null)
      ;
     displayFragment();
     //Check if the user wants to retain the
      previous converted text.
     Fragment_ConvertedText.checkAppend();
    }
   });

   //If play button is pressed
   button_play = (Button) findViewById(R.id.
      button_play);
   button_play.setOnClickListener(new View.
      OnClickListener() {
    @Override
    public void onClick(View v) {

     //allow double click on play.
     lastClickTime = SystemClock.
      elapsedRealtime();

     switch (state) {
      //Play only if it is in DrawingCanvas
     mode.
      case State.IS_CANVAS: {
       //If user is on Canvas.
       Fragment_ConvertedText.checkAppend();
       if (button_play.getText().equals(
     getResources().getString(R.string.
     button_play))) {
        save = new Task_SaveCanvas(
     Activity_Main.this);
        save.execute(true);
       } else {
        save.cancel(true);
        Libraries.tts.stop();
        Libraries.tesseract.clear();

        //Re-enable buttons
        Activity_Main.button_text.setEnabled(
     true);
        Activity_Main.button_text.setClickable(
     true);
        button_play.
     setCompoundDrawablesWithIntrinsicBounds
     (0, R.mipmap.ic_action_playback_play, 0,
     0);
        button_play.setText(R.string.
     button_play);
       }
       break;
      }

      //If user is on display pdf fragment.
      case State.IS_PDF: {
       Fragment_ConvertedText.checkAppend();
       if (button_play.getText().equals(
     getResources().getString(R.string.
     button_play))) {
        Libraries.pdfToText(true);
       } else {
        Libraries.tts.stop();
        Libraries.tesseract.clear();

        //Re-enable buttons
        Activity_Main.button_text.setEnabled(
     true);
        Activity_Main.button_text.setClickable(
     true);
        button_play.
     setCompoundDrawablesWithIntrinsicBounds
     (0, R.mipmap.ic_action_playback_play, 0,
     0);
        button_play.setText(R.string.
     button_play);
       }
       break;
      }

      //If the user is on Convert Image.
      case State.IS_IMAGE: {
```

```
        Fragment_ConvertedText.checkAppend();
        if (button_play.getText().equals(
      getResources().getString(R.string.
      button_play))) {
          new ToText(Activity_Main.this).execute(
      imageUri, true);
        } else {
          Libraries.tts.stop();
          Libraries.tesseract.clear();

          //Re-enable buttons
          Activity_Main.button_text.setEnabled(
      true);
          Activity_Main.button_text.setClickable(
      true);
          button_play.
      setCompoundDrawablesWithIntrinsicBounds
      (0, R.mipmap.ic_action_playback_play, 0,
      0);
          button_play.setText(R.string.
      button_play);
        }
        break;
      }

      //If the user is on Epub
      case State.IS_EPUB: {
        if (button_play.getText().equals(
      getResources().getString(R.string.
      button_play))) {
          new Task_WebViewToText(Activity_Main.
      this).execute(true);
        } else {
          Libraries.tts.stop();
          Libraries.tesseract.clear();

          //Re-enable buttons
          Activity_Main.button_text.setEnabled(
      true);
          Activity_Main.button_text.setClickable(
      true);
          button_play.
      setCompoundDrawablesWithIntrinsicBounds
      (0, R.mipmap.ic_action_playback_play, 0,
      0);
          button_play.setText(R.string.
      button_play);
        }
        break;
      }

      //Play directly if it's a converted text
      fragment.
      default: {
        if (button_play.getText().equals(
      getResources().getString(R.string.
      button_play))) {
          speak = new ToSpeech();
          speak.execute(Fragment_ConvertedText.
      editText.getText().toString());
        } else {
          speak.cancel(true);
          Libraries.tts.stop();
          Libraries.tesseract.clear();

          //Re-enable buttons
          Activity_Main.button_text.setEnabled(
      true);
          Activity_Main.button_text.setClickable(
      true);
          button_play.
      setCompoundDrawablesWithIntrinsicBounds
      (0, R.mipmap.ic_action_playback_play, 0,
      0);
          button_play.setText(R.string.
      button_play);
        }
        break;
      }

    }
  }
});

//If rotate button is pressed.
Fragment_InputImage.button_rotate.
    setOnClickListener(new View.
    OnClickListener() {
  @Override
  public void onClick(View v) {

    //double-clicking prevention, using
     threshold of 1000 ms
    if (SystemClock.elapsedRealtime() -
     lastClickTime < 1000){
      return;
    }
    lastClickTime = SystemClock.
     elapsedRealtime();

    //Create intent to launch
    final Intent intent = new Intent(
     Activity_Main.this, Activity_Rotate.class
     );
    intent.putExtra("imageUri", imageUri);
    startActivityForResult(intent,
     Activity_Result.
     ROTATE_BITMAP_ACTIVITY_REQUEST_CODE);
  }
});

//If crop button in input image is pressed
Fragment_InputImage.button_crop.
    setOnClickListener(new View.
    OnClickListener() {
  @Override
  public void onClick(View v) {
    //double-clicking prevention, using
     threshold of 1000 ms
    if (SystemClock.elapsedRealtime() -
     lastClickTime < 1000){
      return;
    }
    lastClickTime = SystemClock.
     elapsedRealtime();

    //Create intent to launch
    Intent intent = new Intent(Activity_Main.
     this, Activity_Crop.class);
    intent.putExtra("imageUri", imageUri);
    startActivityForResult(intent,
     Activity_Result.
     CROP_IMAGE_ACTIVITY_REQUEST_CODE);
  }
});

//If back button in input image is pressed
Fragment_InputImage.button_back.
    setOnClickListener(new View.
    OnClickListener() {
  @Override
  public void onClick(View v) {

    //double-clicking prevention, using
     threshold of 1000 ms
    if (SystemClock.elapsedRealtime() -
     lastClickTime < 1000){
      return;
    }
    lastClickTime = SystemClock.
     elapsedRealtime();

    //Cancel speaking when back button is
     pressed
    if (save != null) {
      save.cancel(true);
    } else if (speak != null) {
      speak.cancel(true);
    }
    Libraries.tts.stop();
    Libraries.tesseract.clear();

    button_play.setText(R.string.button_play);

    //set state
    state = State.IS_CANVAS;

    Libraries.imageView.setImageDrawable(null)
     ;
    displayFragment();
    //Check if the user wants to retain the
     previous converted text
    Fragment_ConvertedText.checkAppend();
  }
});

//Hide other fragments. Only show canvas
    fragment
displayFragment();
}

@Override
protected void onSaveInstanceState(final
    Bundle savedInstanceState) {
  super.onSaveInstanceState(savedInstanceState
      );
  //Save the App's state when App state is
     changed.

  if (Directories.imageFromCameraUri != null)
     {
    //Save image taken by camera after rotation
     .
    savedInstanceState.putString("
     cameraImageUri", Directories.
     imageFromCameraUri.toString());
  }
  if (imageUri != null && state == State.
```

```
    IS_IMAGE) {
   //Save image in imageView after rotation.
   savedInstanceState.putParcelable("
     imageDisplayed", imageUri);
  } else if (pdfUri != null && state == State.
     IS_PDF) {
   //Save PDF in imageView after rotation.
   savedInstanceState.putParcelable("
     pdfDisplayed", pdfUri);
  } else if (epubUri != null && state == State
     .IS_EPUB) {
   savedInstanceState.putParcelable("
     epubDisplayed", epubUri);
   if (getSupportFragmentManager().
     findFragmentByTag("fragment_epub").
     isVisible()) {
    savedInstanceState.putString("htmlContent"
     , Fragment_EPubViewer_TOC.htmlContent);
   }
  }
  //Save current state of fragments after
     rotation.
  savedInstanceState.putInt("state", state);
  //Save texts in EditTextView after rotation.
  savedInstanceState.putString("editText",
     Fragment_ConvertedText.editText.getText()
     .toString());

}

@SuppressWarnings("ConstantConditions")
@Override
protected void onRestoreInstanceState(final
     Bundle savedInstanceState) {
 //Restore the App's state when App state is
     changed.
 super.onRestoreInstanceState(
     savedInstanceState);
 if (savedInstanceState.containsKey("state"))
     {
  //Restore state of fragments after rotation
     .
  state = savedInstanceState.getInt("state");
  displayFragment();
 }
 if (savedInstanceState.containsKey("
     cameraImageUri")) {
  //Restore image taken by camera after
     rotation
  Directories.imageFromCameraUri = Uri.parse(
     savedInstanceState.getString("
     cameraImageUri"));
 }
 if (savedInstanceState.containsKey("editText
     ")) {
  //Restore text in EditTextView after
     rotation
  Fragment_ConvertedText.editText.setText(
     savedInstanceState.getString("editText"))
     ;
 }
 if (savedInstanceState.containsKey("
     imageDisplayed") && state == State.
     IS_IMAGE) {
  //Restore image in imageView after rotation
  imageUri = savedInstanceState.getParcelable
     ("imageDisplayed");
  setImage();
 } else if (savedInstanceState.containsKey("
     imageDisplayed") && state == State.IS_PDF
     ) {
  //Restore PDF in imageView after rotation
  pdfUri = savedInstanceState.getParcelable("
     pdfDisplayed");
  Libraries.displayPdf(pdfUri);
 } else if (savedInstanceState.containsKey("
     epubDisplayed") && state == State.IS_EPUB
     ) {
  epubUri = savedInstanceState.getParcelable(
     "epubDisplayed");
  //Recreate EPUB table of contents
  final Fragment_EPubViewer_TOC fragment = (
     Fragment_EPubViewer_TOC)
     getSupportFragmentManager().
     findFragmentByTag("fragment_epub_toc");
  fragment.createAdapter(epubUri, Directories
     .EPUB_CACHE_DIR);

  //Display webView
  if (savedInstanceState.containsKey("
     htmlContent")) {
   final String htmlContent =
     savedInstanceState.getString("htmlContent
     ");

   //Convert EPUB_CACHE_DIR to URL string
   String url = null;
   try {
    url = new File(Directories.EPUB_CACHE_DIR
     ).toURI().toURL().toString() + File.
     separator + "OEBPS" + File.separator;
   } catch (MalformedURLException e) {
    Log.e("Url error", "Stack trace:", e);
   }
   Fragment_EPubViewer.webView.
     loadDataWithBaseURL(url, htmlContent, "
     text/html", "utf-8", null);

   //Hide current fragment and display
     fragment_epub
   final FragmentTransaction ft =
     getSupportFragmentManager().
     beginTransaction();
   ft.hide(getSupportFragmentManager().
     findFragmentByTag("fragment_epub_toc"));
   ft.show(getSupportFragmentManager().
     findFragmentByTag("fragment_epub"));
   ft.commit();
  }
 }
}

@Override
protected void onActivityResult(final int
     requestCode, final int resultCode, final
     Intent data) {
 super.onActivityResult(requestCode,
     resultCode, data);

 switch (requestCode) {
  //Result from Camera Activity.
  case Activity_Result.
     CAPTURE_IMAGE_ACTIVITY_REQUEST_CODE: {
   if (resultCode == RESULT_OK) {
    //Use fragment_image as Fragment.
    state = State.IS_IMAGE;
    //Display InputImage.
    displayFragment();
    //Put image in InputImage fragment.
    imageUri = Directories.imageFromCameraUri
     ;
    setImage();
    Libraries.galleryAddPic(imageUri);
    //Image captured and saved on the
     directory.
    Toast.makeText(this, "Image saved to: " +
     imageUri.getPath(), Toast.LENGTH_LONG).
     show();
   } else if (resultCode == RESULT_CANCELED)
     {
    //Image capture cancelled by the user.
    Toast.makeText(this, "Image capture
     cancelled", Toast.LENGTH_SHORT).show();
   } else {
    //An error has occured during saving.
    Toast.makeText(this, "An error has
     occured", Toast.LENGTH_LONG).show();
   }
   break;
  }

  //Result from browse image.
  case Activity_Result.
     BROWSE_IMAGE_ACTIVITY_REQUEST_CODE: {
   if (resultCode == RESULT_OK) {

    if (data.getData() == null && data.
     hasExtra(FilePickerActivity.
     EXTRA_FILE_PATH)) {
     //If built-in file browser is used (
     android-file-picker_activity).
     imageUri = Uri.fromFile(new File(data.
     getStringExtra(FilePickerActivity.
     EXTRA_FILE_PATH)));
    } else {
     //If other file browsers are used.
     imageUri = data.getData();
    }

    //Check if imageUri is not null and has
     proper image format, return error notice
     if it is null or in wrong format.
    if (imageUri != null &&
     Intent_BrowseFiles.checkImageExtension(
     this, imageUri)) {
     //Use fragment_image as Fragment.
     state = State.IS_IMAGE;
     //Display InputImage.
     displayFragment();
     //Set image to imageView
     setImage();
    } else {
     Toast.makeText(this, "Select an image
     with the appropriate format.", Toast.
     LENGTH_LONG).show();
    }

   } else if (resultCode == RESULT_CANCELED)
```

```
 {
  //Image capture cancelled by the user.
  Toast.makeText(this, "Browse Image
  cancelled", Toast.LENGTH_SHORT).show();
 } else {
  //An error has occured during saving.
  Toast.makeText(this, "An error has
  occured", Toast.LENGTH_LONG).show();
 }
 break;
}

//Result from browse PDF.
case Activity_Result.
 BROWSE_PDF_ACTIVITY_REQUEST_CODE: {
 if (resultCode == RESULT_OK) {

  if (data.getData() == null && data.
  hasExtra(FilePickerActivity.
  EXTRA_FILE_PATH)) {
   //If built-in file browser is used (
  android-file-picker_activity).
   pdfUri = Uri.fromFile(new File(data.
  getStringExtra(FilePickerActivity.
  EXTRA_FILE_PATH)));
  } else {
   //If other file browsers are used.
   pdfUri = data.getData();
  }
  //Check if pdfUri is not null and has
  proper format, return error notice if it
  is null or in wrong format.
  if (pdfUri != null && Intent_BrowseFiles.
  checkPdfExtension(this, pdfUri)) {
   //use fragment_pdf as Fragment.
   state = State.IS_PDF;
   //Display PDF in PDFViewer
   Libraries.displayPdf(pdfUri);
   //Display PDFViewer.
   displayFragment();
  } else {
   Toast.makeText(this, "Select a PDF with
  an appropriate format.", Toast.
  LENGTH_LONG).show();
  }

 } else if (resultCode == RESULT_CANCELED)
 {
  //Image capture cancelled by the user.
  Toast.makeText(this, "Browse PDF
  cancelled", Toast.LENGTH_SHORT).show();
 } else {
  //An error has occured during saving.
  Toast.makeText(this, "An error has
  occured", Toast.LENGTH_LONG).show();
 }
 break;
}

//Result from browse Epub.
case Activity_Result.
 BROWSE_EPUB_ACTIVITY_REQUEST_CODE: {
 if (resultCode == RESULT_OK) {
  Fragment_EPubViewer.clearWebView();

  if (data.getData() == null && data.
  hasExtra(FilePickerActivity.
  EXTRA_FILE_PATH)) {
   //If built-in file browser is used (
  android-file-picker_activity).
   epubUri = Uri.fromFile(new File(data.
  getStringExtra(FilePickerActivity.
  EXTRA_FILE_PATH)));
  } else {
   //If other file browsers are used.
   epubUri = data.getData();
  }

  //Check if epubUri is not null and has
  proper format, return error notice if it
  is null or in wrong format.
  if (epubUri != null && Intent_BrowseFiles
  .checkEpubExtension(this, epubUri)) {
   new Task_UnzipEpub(this, new
  Task_UnzipEpub.AsyncResponse() {
    @Override
    public void processFinished(Uri epubUri
  , String cachePath) {
     //Recreate EPUB table of contents
     final Fragment_EPubViewer_TOC fragment
  = (Fragment_EPubViewer_TOC)
  getSupportFragmentManager().
  findFragmentByTag("fragment_epub_toc");
     fragment.createAdapter(epubUri,
  cachePath);
     //use fragment_epub as Fragment.
     state = State.IS_EPUB;
     //Display EpubViewer.
     displayFragment();
    }
   }).execute(epubUri, Directories.
  EPUB_CACHE_DIR);
  } else {
   Toast.makeText(this, "Select an Epub
  with an appropriate format.", Toast.
  LENGTH_LONG).show();
  }

 } else if (resultCode == RESULT_CANCELED)
 {
  //Image capture cancelled by the user.
  Toast.makeText(this, "Browse Epub
  cancelled", Toast.LENGTH_SHORT).show();
 } else {
  //An error has occured during saving.
  Toast.makeText(this, "An error has
  occured", Toast.LENGTH_LONG).show();
 }
 break;
}

//Result from rotate bitmap.
case Activity_Result.
 ROTATE_BITMAP_ACTIVITY_REQUEST_CODE: {
 if (resultCode == RESULT_OK) {
  //If rotating and saving is successful,
  display the rotated image on image view
  imageUri = data.getParcelableExtra("
  imageUri");
  setImage();
 } else if (resultCode == RESULT_CANCELED)
 {
  //Image rotation is cancelled
  Toast.makeText(this, "Image rotation
  cancelled", Toast.LENGTH_SHORT).show();
 }
 break;
}

//Result from crop image.
case Activity_Result.
 CROP_IMAGE_ACTIVITY_REQUEST_CODE: {
 if (resultCode == RESULT_OK) {
  //If cropping and saving is successful,
  display the cropped image on image view
  imageUri = data.getParcelableExtra("
  imageUri");
  Toast.makeText(Activity_Main.this, "
  Cropped image saved to: " + imageUri.
  getPath(), Toast.LENGTH_LONG).show();
  setImage();
 } else if (resultCode == RESULT_CANCELED)
 {
  //Crop picture cancelled by user
  Toast.makeText(this, "Crop cancelled",
  Toast.LENGTH_SHORT).show();
 } else {
  //An error has occured during crop
  Toast.makeText(this, "An error has
  occured", Toast.LENGTH_LONG).show();
 }
 break;
}

//Result from Image to PDF
case Activity_Result.
 BROWSE_MULTIPLE_IMAGE_ACTIVITY_REQUEST_CODE
 : {
 if (resultCode == RESULT_OK) {

  //call save PDF class (AsyncTask auto
  execute)
  new Task_SavePdf(this, data);
  //use fragment_pdf as Fragment
  state = State.IS_PDF;

  //Display PDFViewer
  displayFragment();
 } else if (resultCode == RESULT_CANCELED)
 {
  //Image capture cancelled by the user
  Toast.makeText(this, "Browse File
  cancelled", Toast.LENGTH_SHORT).show();
 } else {
  //An error has occured during saving
  Toast.makeText(this, "An error has
  occured", Toast.LENGTH_LONG).show();
 }
 break;
}

//Result from PDF to Image
case Activity_Result.
 BROWSE_PDF_CONVERT_TO_IMAGE_ACTIVITY_REQUEST_CODE
 : {
 if (resultCode == RESULT_OK) {

  if (data.getData() == null && data.
```

```
    hasExtra(FilePickerActivity.
    EXTRA_FILE_PATH)) {
      //If built-in file browser is used (
    android-file-picker_activity).
      pdfUri = Uri.fromFile(new File(data.
    getStringExtra(FilePickerActivity.
    EXTRA_FILE_PATH)));
    } else {
      //If other file browsers are used.
      pdfUri = data.getData();
    }

    //Check if pdfUri is not null and has
    proper format, return error notice if it
    is null or in wrong format.
    if (pdfUri != null && Intent_BrowseFiles.
    checkPdfExtension(this, pdfUri)) {
      //use fragment_pdf as Fragment
      state = State.IS_PDF;
      //call PDF to Image class
      Libraries.pdfToImage(pdfUri);
      //Display PDFViewer
      displayFragment();
    } else {
      Toast.makeText(this, "Select a PDF with
    an appropriate format.", Toast.
    LENGTH_LONG).show();
    }

   } else if (resultCode == RESULT_CANCELED)
    {
    //Image capture cancelled by the user
    Toast.makeText(this, "Convert to PDF
    cancelled", Toast.LENGTH_SHORT).show();
   } else {
    //An error has occured during saving
    Toast.makeText(this, "An error has
    occured", Toast.LENGTH_LONG).show();
   }
   break;
  }

  default: {
   break;
  }
 }

 displayFragment();
}
/**
     * Set the image to be displayed in the
         imageView.
     * This is needed to be able to display
         large bitmaps without lagging.
     */
private void setImage() {

 //Save the copy of the original file to
    cache.
 new Task_FileFromUri(this, "cache." +
    Intent_BrowseFiles.getMimeType(this,
    imageUri), new Task_FileFromUri.
    AsyncResponse() {
  @Override
  public void processFinished(File file) {
   //Reduce the dimension & size of the
    bitmap copy.
    final Bitmap image = BitmapLoader.
    decodeSampledBitmap(file, Libraries.
    imageView.getWidth(), Libraries.imageView
    .getHeight());
   //Display the bitmap copy.
   Libraries.imageView.setImageBitmap(image);
   Libraries.imageView.setScaleType(ImageView
    .ScaleType.FIT_CENTER);
   //Change the displayed text on button_text
   button_text.setText(R.string.button_text);
  }
 }).execute(imageUri);
}

/**
     * A call to this function determines
         which Fragment will be displayed
         based on the App's state.
     */
private void displayFragment() {
 final FragmentTransaction ft =
    getSupportFragmentManager().
    beginTransaction();
 switch (state) {
  //Display Canvas Fragment
  case State.IS_CANVAS: {
   Libraries.pdfView.recycle();
   Libraries.imageView.setRecycle(true);
   Libraries.imageView.recycle();
   Fragment_EPubViewer.clearWebView();

   ft.hide(getSupportFragmentManager().
    findFragmentByTag("fragment_text"));
   ft.hide(getSupportFragmentManager().
    findFragmentByTag("fragment_image"));
   ft.hide(getSupportFragmentManager().
    findFragmentByTag("fragment_pdf"));
   ft.hide(getSupportFragmentManager().
    findFragmentByTag("fragment_epub_toc"));
   ft.hide(getSupportFragmentManager().
    findFragmentByTag("fragment_epub"));
   ft.show(getSupportFragmentManager().
    findFragmentByTag("fragment_canvas"));
   button_text.setText(R.string.button_text);
   break;
  }

  //Display imageviewer Fragment
  case State.IS_IMAGE: {
   Libraries.pdfView.recycle();
   Fragment_DrawingCanvas.drawingView.
    destroyDrawingCache();
   Fragment_EPubViewer.clearWebView();

   ft.hide(getSupportFragmentManager().
    findFragmentByTag("fragment_text"));
   ft.hide(getSupportFragmentManager().
    findFragmentByTag("fragment_canvas"));
   ft.hide(getSupportFragmentManager().
    findFragmentByTag("fragment_pdf"));
   ft.hide(getSupportFragmentManager().
    findFragmentByTag("fragment_epub_toc"));
   ft.hide(getSupportFragmentManager().
    findFragmentByTag("fragment_epub"));
   ft.show(getSupportFragmentManager().
    findFragmentByTag("fragment_image"));

   button_text.setText(R.string.button_text);
   break;
  }

  //Display PDFViewer Fragment
  case State.IS_PDF: {
   Libraries.imageView.setRecycle(true);
   Libraries.imageView.recycle();
   Fragment_DrawingCanvas.drawingView.
    destroyDrawingCache();
   Fragment_EPubViewer.clearWebView();

   ft.hide(getSupportFragmentManager().
    findFragmentByTag("fragment_text"));
   ft.hide(getSupportFragmentManager().
    findFragmentByTag("fragment_canvas"));
   ft.hide(getSupportFragmentManager().
    findFragmentByTag("fragment_image"));
   ft.hide(getSupportFragmentManager().
    findFragmentByTag("fragment_epub_toc"));
   ft.hide(getSupportFragmentManager().
    findFragmentByTag("fragment_epub"));
   ft.show(getSupportFragmentManager().
    findFragmentByTag("fragment_pdf"));

   button_text.setText(R.string.button_text);
   break;
  }

  //Display Epub Fragment
  case State.IS_EPUB: {
   Libraries.pdfView.recycle();
   Libraries.imageView.setRecycle(true);
   Libraries.imageView.recycle();

   ft.hide(getSupportFragmentManager().
    findFragmentByTag("fragment_text"));
   ft.hide(getSupportFragmentManager().
    findFragmentByTag("fragment_image"));
   ft.hide(getSupportFragmentManager().
    findFragmentByTag("fragment_pdf"));
   ft.hide(getSupportFragmentManager().
    findFragmentByTag("fragment_canvas"));
   ft.hide(getSupportFragmentManager().
    findFragmentByTag("fragment_epub"));
   ft.show(getSupportFragmentManager().
    findFragmentByTag("fragment_epub_toc"));

   button_text.setText(R.string.button_text);
   break;
  }

  //Display Text Fragment
  case State.IS_TEXT_FROM_CANVAS:
  case State.IS_TEXT_FROM_IMAGE:
  case State.IS_TEXT_FROM_PDF:
  case State.IS_TEXT_FROM_EPUB: {

   ft.hide(getSupportFragmentManager().
    findFragmentByTag("fragment_canvas"));
   ft.hide(getSupportFragmentManager().
    findFragmentByTag("fragment_image"));
   ft.hide(getSupportFragmentManager().
    findFragmentByTag("fragment_pdf"));
```

```
        ft.hide(getSupportFragmentManager().
         findFragmentByTag("fragment_epub_toc"));
        ft.hide(getSupportFragmentManager().
         findFragmentByTag("fragment_epub"));
        ft.show(getSupportFragmentManager().
         findFragmentByTag("fragment_text"));

        button_text.setText(R.string.
         button_text_back);
        break;
       }

       default: {
        break;
       }
      }
      //commit Fragment changes.
      ft.commit();
    }

    @Override
    protected void onPause() {
     Libraries.tts.stop();
     super.onPause();
    }

    @Override
    protected void onStop() {
     Libraries.tts.stop();
     Libraries.tts.shutdown();
     super.onStop();
    }

    @Override
    protected void onDestroy () {
     super.onDestroy();
     //Garbage Collect
     Libraries.tts.stop();
     Libraries.tts.shutdown();
     Libraries.tesseract.clear();
     Libraries.tesseract.end();
     Fragment_DrawingCanvas.drawingView.
        destroyDrawingCache();
     Fragment_DrawingCanvas.drawingView.
        invalidate();
     Libraries.imageView.setRecycle(true);
     Libraries.imageView.recycle();
     Libraries.pdfView.recycle();
     Fragment_EPubViewer.clearWebView();
     Fragment_EPubViewer.webView.destroy();
     //Delete epubCache and cacheDir
     if (isFinishing()) {
      Directories.delete(new File(Directories.
       EPUB_CACHE_DIR));
      Directories.delete(getCacheDir());
     }
    }
}
```

Path = `Tess2Speech/tess2Speech/src/main/java/anteraaron/tess2speech/Activity\_Rotate.java`

```
package anteraaron.tess2speech;

import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.Matrix;
import android.graphics.RectF;
import android.net.Uri;
import android.support.v7.app.
    AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.webkit.MimeTypeMap;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.SeekBar;
import android.widget.TextView;

import java.io.File;

/**
 * Activity for Rotating Images.
 * @author Anter Aaron M. Custodio
 * @since 2016.3.22
 */
public class Activity_Rotate extends
    AppCompatActivity{

 private static final int STARTING_ANGLE =
     180; //Starting angle is angle - 180.
     This implies that the amount of starting
     rotation is 0.
 private static final int MAX_ANGLE = 360; //
     360 - 180 = 180. This implies that the
     max rotation can only be at -180 to 180.
 private static final String LABEL = "Rotated
     angle: "; //Rotation value indicator

 private ImageView imageView_rotate;
 private TextView textView_rotate;
 private static int angle;
 private static String imagePath, imageName;

 @Override
 protected void onCreate(final Bundle
     savedInstanceState) {
  super.onCreate(savedInstanceState);
  setContentView(R.layout.activity_rotate);
  //get uri of the image displayed
  final Uri uri = getIntent().
     getParcelableExtra("imageUri");
  //get the original name of the image
  imageName = new File(uri.getPath()).getName
     ();
  //create UI for this activity

  //Set initial text on textView
  final String angleLabel = LABEL + "0";
  this.textView_rotate = (TextView)
     findViewById(R.id.textView_rotate);
  this.textView_rotate.setText(angleLabel);

  //Cast imageView
  this.imageView_rotate = (ImageView)
     findViewById(R.id.imageView_rotate);
  setImage(uri);

  //Cast seekBar
  final SeekBar seekBar = (SeekBar)
     findViewById(R.id.seekBar_rotate);
  seekBar.setMax(MAX_ANGLE);
  seekBar.setProgress(STARTING_ANGLE);
  seekBar.setOnSeekBarChangeListener(new
     SeekBar.OnSeekBarChangeListener() {
   @Override
   public void onProgressChanged(final SeekBar
      seekBar, final int progress, final
     boolean fromUser) {
    //The current angle is current seekBar
     progress - 180 (reason for -180 in the
     constant value above).
    angle = progress - 180;
    rotateImage(angle);
   }
   @Override
   public void onStartTrackingTouch(SeekBar
     seekBar) {}

   @Override
   public void onStopTrackingTouch(SeekBar
     seekBar) {}
  });

  //Create save canvas button.
  final Button button_rotate_save = (Button)
     findViewById(R.id.button_rotate_save);
  button_rotate_save.setOnClickListener(new
     View.OnClickListener() {
   @Override
   public void onClick(View v) {
    //Calls an AsyncTask that saves the
     rotated image to a file.
    new Task_SaveRotatedBitmap(Activity_Rotate
     .this, new Task_SaveRotatedBitmap.
     AsyncResponse() {
      @Override
      public void processFinished(Uri uri) {
       //Return file after saving the rotated
     image to get file path.
       final Intent intent = new Intent();
       intent.putExtra("imageUri", uri);
       setResult(RESULT_OK, intent);
       imageView_rotate.destroyDrawingCache();
```

```
            finish();
        }
      }).execute(imagePath, angle, imageName);
    }
  });

  //Create reset button.
  final Button button_rotate_reset = (Button)
     findViewById(R.id.button_rotate_reset);
  button_rotate_reset.setOnClickListener(new
     View.OnClickListener() {
    @Override
    public void onClick(View v) {
     //Resets the angle. Returns the rotation
      of the image to its original orientation.
     angle = 0;
     seekBar.setProgress(180);
     rotateImage(angle);
    }
  });

  //Create cancel button.
  final Button button_rotate_cancel = (Button)
      findViewById(R.id.button_rotate_exit);
  button_rotate_cancel.setOnClickListener(new
     View.OnClickListener() {
    @Override
    public void onClick(View v) {
     //Returns cancel result to Activity_Main.
     setResult(RESULT_CANCELED);
     finish();
    }
  });
 }

 /**
     * Method that accepts a rotation value,
          and rotate the image displayed in
          imageView_rotate and adjusts its
          size accordingly so that the whole
          image is still displayed.
     * @param angle a float value from -180 to
          180.
     * @since 2016.3.22
     */
 private void rotateImage(final float angle){

  //Pre-compute some trig functions.
  final double radians = Math.toRadians(angle)
     ;
  final double sin = Math.abs(Math.sin(radians
     ));
  final double cos = Math.abs(Math.cos(radians
     ));

  //Figure out total width and height of new
     bitmap using trig computations.
  final float newWidth = (float) (this.
     imageView_rotate.getWidth() * cos + this.
     imageView_rotate.getHeight() * sin);
  final float newHeight = (float) (this.
     imageView_rotate.getWidth() * sin + this.
     imageView_rotate.getHeight() * cos);

  //Figure out the amount of adjustment of
     imageView to fit the rotated bitmap.
  final float widthAdjustment = (newWidth -
     this.imageView_rotate.getWidth())/2;
  final float heightAdjustment = (newHeight -
     this.imageView_rotate.getHeight())/2;

  //Set the view and drawable's bound.
  this.imageView_rotate.setScaleType(ImageView
     .ScaleType.MATRIX);
  final RectF drawableRect = new RectF(0, 0,
     this.imageView_rotate.getDrawable().
     getBounds().width(), this.
     imageView_rotate.getDrawable().getBounds
     ().height());
  final RectF viewRect = new RectF(0, 0, this.
     imageView_rotate.getWidth() -
     widthAdjustment, this.imageView_rotate.
     getHeight() - heightAdjustment);

  //Use Matrix for rotation
  final Matrix matrix = new Matrix();
  matrix.setRectToRect(drawableRect, viewRect,
      Matrix.ScaleToFit.CENTER);
  matrix.postRotate(angle, (this.
     imageView_rotate.getWidth() -
     widthAdjustment) / 2, (this.
     imageView_rotate.getHeight() -
     heightAdjustment) / 2);
  matrix.postTranslate(widthAdjustment/2,
     heightAdjustment/2);
  this.imageView_rotate.setImageMatrix(matrix)
     ;

  //Refresh the image view
  this.imageView_rotate.invalidate();

  //Change the value of rotation value
     indicator.
  final String angleLabel = LABEL + angle;
  this.textView_rotate.setText(angleLabel);
 }

 /**
     * A method that accepts an uri, and
          displays it to imageView_rotate view
          .
     * @param uri uri of the image that will
          be displayed in imageView_rotate.
     * @since 2016.3.22
     */
 private void setImage(final Uri uri) {
  //Get file extension of image from uri.
  final MimeTypeMap mime = MimeTypeMap.
     getSingleton();
  final String extension = mime.
     getExtensionFromMimeType(
     getContentResolver().getType(uri));
  //Get resized file path from uri.
  new Task_FileFromUri(this, "cache." +
     extension, new Task_FileFromUri.
     AsyncResponse() {
   @Override
   public void processFinished(File file) {
    //Display the image to imageView_rotate.
    final Bitmap image = BitmapLoader.
     decodeSampledBitmap(file,
     imageView_rotate.getWidth(),
     imageView_rotate.getHeight());
    imageView_rotate.setImageBitmap(image);
    imageView_rotate.setScaleType(ImageView.
     ScaleType.FIT_CENTER);
    imagePath = file.getAbsolutePath();
   }
  }).execute(uri);
 }
}
```

Path = Tess2Speech/tess2Speech/src/main/java/anteraaron/tess2speech/Activity_Settings.java

```
package anteraaron.tess2speech;

import android.content.Intent;
import android.content.pm.PackageManager;
import android.content.pm.ResolveInfo;
import android.os.Bundle;
import android.preference.PreferenceActivity;
import android.service.textservice.
     SpellCheckerService;

/**
 * Activity for Preferences Settings.
 * @author Anter Aaron M. Custodio
 * @since 2016.3.22
 */
public class Activity_Settings extends
     PreferenceActivity {
 public static boolean spellCheckerPresent;
 @Override
 public void onCreate(final Bundle
     savedInstanceState) {
  super.onCreate(savedInstanceState);
  //check if there is a built in spell-checker
  final PackageManager pm = getPackageManager
     ();
  final Intent spell = new Intent(
     SpellCheckerService.SERVICE_INTERFACE);
  final ResolveInfo info = pm.resolveService(
     spell, 0);
  //Set spellCheckerPresent to false if
     spellChecker is not present
  spellCheckerPresent = (info != null);
  getFragmentManager().beginTransaction().
     replace(android.R.id.content, new
     Fragment_Settings()).commit();
 }
}
```

Path = Tess2Speech/tess2Speech/src/main/java/anteraaron/tess2speech/Activity_SplashScreen.java

```
package anteraaron.tess2speech;

import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.preference.PreferenceManager;
import android.speech.tts.TextToSpeech;
import android.support.v7.app.
    AppCompatActivity;
import android.view.WindowManager;
import android.widget.Toast;
import java.io.File;

/**
 * An Activity that displays the Splash screen
       and copy the required files.
 * @author Anter Aaron M. Custodio
 * @since 2016.3.22
 */
public class Activity_SplashScreen extends
    AppCompatActivity {
 private static final int
    SPLASH_DISPLAY_LENGTH = 600; //splash
    screen display duration.
 private static final int
    CHECK_TTS_REQUEST_CODE = 1;

 @Override
 protected void onCreate(final Bundle
    savedInstanceState) {
  super.onCreate(savedInstanceState);
  getWindow().setFlags(WindowManager.
    LayoutParams.FLAG_FULLSCREEN,
    WindowManager.LayoutParams.
    FLAG_FULLSCREEN);
  setContentView(R.layout.splash_screen);

  //Set default preferences
  PreferenceManager.setDefaultValues(this, R.
    xml.preferences, false);

  //If SD card is present.
  if (Directories.checkExternalStorage()) {
   //check if directories/files needed are
     present.
   if (requiredFilesPresent()) {
    //Display splash screen without copying
     required files.
    new Handler().postDelayed(new Runnable() {
     @Override
     public void run() {
      //Check if there is a TTS installed.
      final Intent intent = new Intent();
      intent.setAction(TextToSpeech.Engine.
    ACTION_CHECK_TTS_DATA);
      startActivityForResult(intent,
    CHECK_TTS_REQUEST_CODE);
     }
    }, SPLASH_DISPLAY_LENGTH);
   } else {
    //If directories do not exists, start
     copying required files and display splash
      screen.
    new Task_CopyAssets(this, new
     Task_CopyAssets.AsyncResponse() {
     @Override
     public void processFinished() {
      //Check if there is a TTS installed.
      final Intent intent = new Intent();
      intent.setAction(TextToSpeech.Engine.
    ACTION_CHECK_TTS_DATA);
      startActivityForResult(intent,
    CHECK_TTS_REQUEST_CODE);
     }
    }).execute();
   }
  } else {
   //Display alert Dialog that SD card is not
     mounted
   final AlertDialog.Builder builder = new
     AlertDialog.Builder(this);
   builder.setTitle("SD Card not mounted");
   builder.setMessage("An SD Card is needed to
     run this application. Please insert or
    mount an SD card");
   builder.setCancelable(false);
   builder.setPositiveButton("Ok", new
     DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog
    , int which) {
     dialog.cancel();
     finish();
    }
   });

   final AlertDialog dialog = builder.create()
    ;
   dialog.show();
  }
 }

 @Override
 protected void onActivityResult (int
    requestCode, int resultCode, Intent data)
     {
  if (requestCode == CHECK_TTS_REQUEST_CODE) {
   if (resultCode == TextToSpeech.Engine.
    CHECK_VOICE_DATA_PASS) {
    // All checks OK. Create an Intent that
     will start the Main Activity after
     displaying splash screen.
    final Intent intent = new Intent(
     Activity_SplashScreen.this, Activity_Main
     .class);
    startActivity(intent);
    finish();
    //SplashScreen transition animation.
    overridePendingTransition(android.R.anim.
     fade_in, android.R.anim.fade_out);
   }
  } else {
   final Intent installIntent = new Intent();
   installIntent.setAction(TextToSpeech.Engine
    .ACTION_INSTALL_TTS_DATA);
   startActivity(installIntent);
   Toast.makeText(this,"TTS installing", Toast
    .LENGTH_LONG).show();
  }
 }

 /**
     * Create required folders and checks
         whether the required asset files are
          already in the SD card.
     * @return true if the files are present.
     */
 @SuppressWarnings({"ResultOfMethodCallIgnored
    ", "ConstantConditions"})
 private boolean requiredFilesPresent(){
  //create required folders for this app

  //Music Directory
  File dir = new File(Directories.AUDIO_DIR);
  if(!dir.exists()){
   dir.mkdir();
  }
  //Document Directory
  dir = new File(Directories.PDF_DIR);
  if(!dir.exists()){
   dir.mkdir();
  }
  //Document Directory
  dir = new File(Directories.TEXT_DIR);
  if(!dir.exists()){
   dir.mkdir();
  }
  //Epub Directory
  dir = new File(Directories.EPUB_CACHE_DIR);
  if(!dir.exists()){
   dir.mkdir();
  }
  //Pictures Directory
  dir = new File(Directories.IMAGES_DIR);
  if(!dir.exists()){
   dir.mkdir();
  }
  //Tessdata folder that will contain .
    traineddatas
  dir = new File(getExternalFilesDir("tessdata
    ").getPath());
  if(!dir.exists()){
   dir.mkdir();
  }
  //Count the content of the tessdata folder
    to determine whether the required files
    are present
  final File[] contents = dir.listFiles();
  return contents.length > 0;
 }
}
```

Path = Tess2Speech/tess2Speech/src/main/java/anteraaron/tess2speech/

BitmapLoader.java

```
package anteraaron.tess2speech;

import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import java.io.File;

/**
 * A class that contains method for Resampling
      large bitmaps to avoid Out of Memory
     Errors and Accelerate image conversion.
 * @author Android Development tutorial &
     Anter Aaron M. Custodio
 * @since 2016.3.22
 */
public class BitmapLoader {

 /**
     * Method that calls different image pre-
         processing.
     */
 public static Bitmap imagePreProcess (Bitmap
     bitmap) {
  bitmap = ImagePreProcess.toGrayScale(bitmap)
     ;
  bitmap = ImagePreProcess.
     changeBitmapContrastBrightness(bitmap);

  return bitmap;
 }

 /**
     * Reduces the size of bitmap based on the
         calculated InSampleSize.
     * @param file the bitmap file to be
         resampled.
     * @param reqWidth required width of the
         bitmap.
     * @param reqHeight required height of the
         bitmap.
     * @return returns the resampled bitmap
         with reqWidth x reqHeight dimension.
     */
 public static Bitmap decodeSampledBitmap(
     final File file, final int reqWidth,
     final int reqHeight) {

  //First decode with inJustDecodeBounds=true
     to check dimensions
  final BitmapFactory.Options options = new
     BitmapFactory.Options();
  options.inJustDecodeBounds = true;
  BitmapFactory.decodeFile(file.
     getAbsolutePath(), options);
  //Calculate inSampleSize
  options.inSampleSize = calculateInSampleSize
     (options, reqWidth, reqHeight);
  //Decode bitmap with inSampleSize set
  options.inJustDecodeBounds = false;

  return BitmapFactory.decodeFile(file.
     getAbsolutePath(), options);
 }

 /**
     * Calculates the appropriate InSampleSize
         of the Bitmap.
     * @param options options containing the
         decoded bounds of the bitmap.
     * @param reqWidth required width of the
         bitmap.
     * @param reqHeight required height of the
         bitmap.
     * @return returns the appropriate sample
         size for resizing the bitmap.
     */
 private static int calculateInSampleSize(
     final BitmapFactory.Options options,
     final int reqWidth, final int reqHeight)
     {
  // Raw height and width of image
  final int height = options.outHeight;
  final int width = options.outWidth;
  int inSampleSize = 1;

  //Resize only if the bitmap is larger than
     the required dimension.
  if (height > reqHeight || width > reqWidth)
     {
   final int halfHeight = height / 2;
   final int halfWidth = width / 2;
   // Calculate the largest inSampleSize value
      that is a power of 2 and keeps both
     height and width larger than the
     requested height and width.
   while ((halfHeight / inSampleSize) >
     reqHeight && (halfWidth / inSampleSize) >
      reqWidth) {
    inSampleSize *= 2;
   }
  }
  return inSampleSize;
 }
}
```

Path = Tess2Speech/tess2Speech/src/main/java/anteraaron/tess2speech/

Directories.java

```
package anteraaron.tess2speech;

import android.net.Uri;
import android.os.Environment;

import java.io.File;

/**
 * Class that contains constant values for the
      application's Directories and File
     paths.
 * @author Anter Aaron M. Custodio
 * @since 2016.3.22
 */
public final class Directories{
 //Saved photo directory
 public static Uri imageFromCameraUri;
 //Saved audio directory
 public static final String AUDIO_DIR =
     Environment.
     getExternalStoragePublicDirectory(
     Environment.DIRECTORY_MUSIC).
     getAbsolutePath();
 //Saved pdf directory
 public static final String PDF_DIR =
     Environment.
     getExternalStoragePublicDirectory("
     Documents").getAbsolutePath();
 //Saved text directory
 public static final String TEXT_DIR =
     Environment.
     getExternalStoragePublicDirectory("
     Documents").getAbsolutePath();
 //Saved input images
 public static final String IMAGES_DIR =
     Environment.
     getExternalStoragePublicDirectory(
     Environment.DIRECTORY_PICTURES).
     getAbsolutePath() + File.separator + "
     Tesseract";
 //Saved text directory
 public static final String EPUB_CACHE_DIR =
     Environment.
     getExternalStoragePublicDirectory("
     Documents").getAbsolutePath() + File.
     separator + "epubCache";

 /**
     * Checks if an SD card is mounted.
     * @return returns true if there is an SD
```

```
        card mounted.
    */
 public static boolean checkExternalStorage(){
  return Environment.getExternalStorageState()
     .equals(Environment.MEDIA_MOUNTED);
 }

 /**
     * A recursive way for deleting a file or
         directory and its contents.
     * @param fileOrDirectory the file or
         directory to be deleted.
     */
 public static void delete(final File
     fileOrDirectory) {
  if (fileOrDirectory.isDirectory()) {
   for (final File child : fileOrDirectory.
     listFiles()) {
    delete(child);
   }
  }
  fileOrDirectory.delete();
 }
}
```

Path = Tess2Speech/tess2Speech/src/main/java/anteraaron/tess2speech/

DrawingView.java

```
package anteraaron.tess2speech;

import android.content.Context;
import android.util.AttributeSet;
import android.util.Log;
import android.view.View;
import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.Path;
import android.view.MotionEvent;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

/**
 * The Drawable Canvas View of the application
       (Using gestures).
 * @author Anter Aaron M. Custodio
 * @since 2016.3.22
 */
public class DrawingView extends View {

 private Path path; //Drawn path
 private Paint paint; //Color of the path
 private float x, y;
 private static final float TOLERANCE = 2; //
     Tolerance level to a void aliasing
 private static final float STROKE_WIDTH = 10f
     ; //Size of stroke
 private static final float ERASE_WIDTH = 25f;
      //Size of deleting stroke
 private float strokeWidth;
 private int selectedColor = Color.BLACK;
 private boolean drawPoint;

 //Hash Maps for maintaining the drawn canvas.
 private Map<Path, Integer> colorsMap;
 private Map<Path, Float> widthMap;
 private ArrayList<Path> paths;

 public DrawingView(final Context c, final
     AttributeSet attrs) {
  super(c, attrs);

  //Initialize
  this.strokeWidth = STROKE_WIDTH;
  this.drawPoint = false;

  //Initialize HashMaps
  this.colorsMap = new HashMap<>();
  this.widthMap = new HashMap<>();
  this.paths = new ArrayList<>();

  //Set new Path
  this.path = new Path();

  //Set new Paint with the desired attributes.
  this.paint = new Paint();
  this.paint.setAntiAlias(true);
  this.paint.setColor(Color.BLACK);
  this.paint.setStyle(Paint.Style.STROKE);
  this.paint.setStrokeJoin(Paint.Join.ROUND);
  this.paint.setStrokeWidth(5f);
 }

 //Override onSizeChanged
 @Override
 protected void onSizeChanged(final int width,
      final int height, final int oldWidth,
     final int oldHeight) {
  super.onSizeChanged(width, height, oldWidth,
      oldHeight);

  // draw onto the defined Bitmap
  final Bitmap bitmap = Bitmap.createBitmap(
     width, height, Bitmap.Config.ARGB_8888);
  new Canvas(bitmap);
 }

 //Override onDraw
 @Override
 protected void onDraw(Canvas canvas) {
  super.onDraw(canvas);
  //Draw the paths when onDraw is called with
     the set paint attribute
  for (final Path p : paths){
   this.paint.setColor(this.colorsMap.get(p));
   this.paint.setStrokeWidth(this.widthMap.get
     (p));
   canvas.drawPath(p, this.paint);
  }
  //Set attributes of the stroke
  this.paint.setColor(this.selectedColor);
  this.paint.setStrokeWidth(this.strokeWidth);
  canvas.drawPath(this.path, this.paint);
 }

 /**
     * When ACTION_DOWN (gesture has started).
     * @param x x coordinate of touched part
         of the canvas
     * @param y y coordinate of touched part
         of the canvas
     */
 private void startTouch(final float x, final
     float y) {
  this.drawPoint = true;
  this.path.moveTo(x, y);
  this.x = x;
  this.y = y;
 }

 /**
     * When ACTION_MOVE (change has happened
         during pressed gesture)
     * @param x x coordinate of touched part
         of the canvas
     * @param y y coordinate of touched part
         of the canvas
     */
 private void moveTouch(final float x, final
     float y) {
  this.drawPoint = false;
  final float dx = Math.abs(x - this.x);
  final float dy = Math.abs(y - this.y);

  if (dx >= TOLERANCE || dy >= TOLERANCE) {
   this.path.quadTo(this.x, this.y, (x + this.
     x) / 2, (y + this.y) / 2);
   this.x = x;
   this.y = y;
  }
 }

 /**
     * When ACTION_UP (gesture stop)
     */
 private void upTouch() {
  this.path.lineTo(this.x, this.y);
  this.paths.add(this.path);
  this.colorsMap.put(this.path, this.
     selectedColor);
  this.widthMap.put(this.path, this.
     strokeWidth);
```

```
        this.path = new Path();

        //Create a dot
        this.path.setLastPoint(this.x, this.y);
        this.x+=5;
        this.path.lineTo(this.x, this.y);
        this.paths.add(this.path);
        this.colorsMap.put(this.path, this.
            selectedColor);
        this.widthMap.put(this.path, this.
            strokeWidth);
        this.path = new Path();
        this.x-=5;

        //Create a dot
        this.path.setLastPoint(this.x, this.y);
        this.y--;
        this.path.lineTo(this.x, this.y);
        this.paths.add(this.path);
        this.colorsMap.put(this.path, this.
            selectedColor);
        this.widthMap.put(this.path, this.
            strokeWidth);
        this.path = new Path();
        this.y++;

        //Create a dot
        this.x-=5 ;
        this.path.setLastPoint(this.x, this.y);
        this.x+=5;
        this.path.lineTo(this.x, this.y);
        this.paths.add(this.path);
        this.colorsMap.put(this.path, this.
            selectedColor);
        this.widthMap.put(this.path, this.
            strokeWidth);
        this.path = new Path();

        this.path.reset();
    }

    /**
         * Method that clears all that is written
             in the canvas.
         */
    public void clearCanvas() {
        this.colorsMap.clear();
        this.paths.clear();
        this.path.reset();
        invalidate();
    }

    /**
         * Method that changes the color of the
             paths from white to black and vice-
             versa.
         */
    public void changeInk(){
        if(this.selectedColor == Color.BLACK){
            //Set path color to white and increase the
                size of stroke to erase strokes easily.
            this.selectedColor = Color.WHITE;
            this.strokeWidth = ERASE_WIDTH;
        }else{
            //Set path color to black.
            this.selectedColor = Color.BLACK;
            this.strokeWidth = STROKE_WIDTH;
        }
    }

    //override the onTouchEvent
    @Override
    public boolean onTouchEvent(final MotionEvent
         event) {
        final float x = event.getX();
        final float y = event.getY();
        //Calls corresponding methods according to
            user gestures.
        switch (event.getAction()) {
            case MotionEvent.ACTION_DOWN:
                startTouch(x, y);
                invalidate();
                break;
            case MotionEvent.ACTION_MOVE:
                moveTouch(x, y);
                invalidate();
                break;
            case MotionEvent.ACTION_UP:
                upTouch();
                invalidate();
                break;
        }
        return true;
    }
}
```

Path = Tess2Speech/tess2Speech/src/main/java/anteraaron/tess2speech/Fragment_ConvertedText.java

```
package anteraaron.tess2speech;

import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.EditText;
import android.widget.Switch;

/**
 * A Fragment for the Converted texts which is
       embedded in the Main_Activity
 * @author Anter Aaron M. Custodio
 * @since 2016.3.22
 */
public class Fragment_ConvertedText extends
    Fragment {
    public static EditText editText;
    public static Switch switch_append;

    @Override
    public View onCreateView(final LayoutInflater
         inflater, final ViewGroup container,
        final Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return createUI(inflater.inflate(R.layout.
            fragment_converted_text, container, false
            ));
    }

    /**
         * Create UI for Converted Text Fragment
         * @param view View to be inflated.
         * @return returns the inflated View.
         */
    private View createUI(final View view) {
        //Create the lined text view in which the
            converted texts will be displayed.
        editText = (EditText)view.findViewById(R.id.
            editText);
        editText.setText("");
        //Creates switch for appending text.
        switch_append = (Switch)view.findViewById(R.
            id.switch_append);

        return view;
    }

    /**
         * Checks whether append switch is on. If
             it is not on, clear the
             linedEditText view.
         */
    public static void checkAppend(){
        //Check if the user wants to retain the
            previous converted text
        if (!switch_append.isChecked() || editText.
            getText().length() == 0) {
            Fragment_ConvertedText.editText.setText("")
                ;
        } else {
            Fragment_ConvertedText.editText.append("\n"
                );
        }
    }
}
```

Path = Tess2Speech/tess2Speech/src/main/java/anteraaron/tess2speech/

Fragment_DrawingCanvas.java

```
package anteraaron.tess2speech;

import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.Toast;

/**
 * A Fragment for the Drawing Canvas which is
      embedded in the Main_Activity
 * @author Anter Aaron M. Custodio
 * @since 2016.3.22
 */
public class Fragment_DrawingCanvas extends
     Fragment {
 public static DrawingView drawingView;

 @Override
 public View onCreateView(final LayoutInflater
      inflater, final ViewGroup container,
     final Bundle savedInstanceState) {
  //Inflate the layout for this fragment
  return createUI(inflater.inflate(R.layout.
     fragment_canvas, container, false));
 }

 /**
     * Create UI for Drawing Canvas fragment.
     * @param view View to be inflated.
     * @return returns the inflated view.
     */
 private View createUI(final View view){
  //Create the Drawing View.
  drawingView = (DrawingView)view.findViewById
     (R.id.drawing);

  //Create clear canvas button.
  final Button button_clear = (Button) view.
     findViewById(R.id.button_clear);
  button_clear.setOnClickListener(new View.
     OnClickListener() {
   @Override
   public void onClick(View v) {
    drawingView.clearCanvas();
    Toast.makeText(getActivity(), "Canvas
     Cleared!", Toast.LENGTH_SHORT).show();
   }
  });

  //Create stroke button.
  final Button button_stroke = (Button)view.
     findViewById(R.id.button_stroke);
  button_stroke.setOnClickListener(new View.
     OnClickListener() {
   @Override
   public void onClick(View v) {
   if(button_stroke.getText().equals(
     getResources().getString(R.string.
     button_stroke_stroke))){
    //Set displayed button as erase
    drawingView.changeInk();
    button_stroke.
     setCompoundDrawablesWithIntrinsicBounds
     (0, R.mipmap.ic_eraser, 0, 0);
    button_stroke.setText(getResources().
     getString(R.string.button_stroke));
   }else{
    //Set displayed button as stroke
    drawingView.changeInk();
    button_stroke.
     setCompoundDrawablesWithIntrinsicBounds
     (0, R.mipmap.ic_action_brush, 0, 0);
    button_stroke.setText(getResources().
     getString(R.string.button_stroke_stroke))
     ;
   }
   }
  });

  return view;
 }
}
```

Path = Tess2Speech/tess2Speech/src/main/java/anteraaron/tess2speech/

Fragment_EPubViewer.java

```
package anteraaron.tess2speech;

import android.annotation.SuppressLint;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.support.v4.app.
     FragmentTransaction;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.Button;
import android.widget.RelativeLayout;

import java.io.File;

/**
 * A Fragment for displaying the data of the
      clicked table of contents which is
      embedded in the Main_Activity
 * @author Anter Aaron M. Custodio
 * @since 2016.3.27
 */
public class Fragment_EPubViewer extends
     Fragment {
 public static WebView webView;

 @Override
 public View onCreateView(final LayoutInflater
      inflater, final ViewGroup container,
     final Bundle savedInstanceState) {
  // Inflate the layout for this fragment
  return createUI(inflater.inflate(R.layout.
     fragment_epub, container, false));
 }

 /**
     * Create UI for Epub Fragment
     * @param view View to be inflated.
     * @return returns the inflated View.
     */
 @SuppressLint("SetJavaScriptEnabled")
 private View createUI(final View view) {
  //Create webview
  webView = new WebView(getActivity().
     getApplicationContext());
  webView.setWebViewClient(new WebViewClient()
     );
  RelativeLayout.LayoutParams lp = new
     RelativeLayout.LayoutParams(
     RelativeLayout.LayoutParams.MATCH_PARENT,
      RelativeLayout.LayoutParams.MATCH_PARENT
     );
  lp.addRule(RelativeLayout.BELOW, R.id.
     button_epub_back_to_toc);
  webView.setLayoutParams(lp);
  webView.getSettings().setJavaScriptEnabled(
     true);

  RelativeLayout relativeLayout = (
     RelativeLayout)view.findViewById(R.id.
     epub_layout);
  relativeLayout.addView(webView);


  //create Back to Table of Contents
  final Button button_epub_back_to_toc = (
     Button)view.findViewById(R.id.
     button_epub_back_to_toc);
  button_epub_back_to_toc.setOnClickListener(
     new View.OnClickListener() {
   @Override
   public void onClick(View v) {
    //Do not clear all when only going back to
     toc.
```

```
    webView.stopLoading();
    webView.clearHistory();
    webView.clearCache(true);
    webView.clearFormData();
    webView.clearSslPreferences();
    webView.loadUrl("about:blank");

    //Hide current fragment and display
     fragment_epub_toc
    final FragmentTransaction ft = getActivity
     ().getSupportFragmentManager().
     beginTransaction();
    ft.show(getActivity().
     getSupportFragmentManager().
     findFragmentByTag("fragment_epub_toc"));
    ft.hide(getActivity().
     getSupportFragmentManager().
     findFragmentByTag("fragment_epub"));
    ft.commit();
   }
  });

  //create Back to Canvas
  final Button button_epub_back_to_canvas = (
     Button)view.findViewById(R.id.
     button_epub_back_to_canvas);
  button_epub_back_to_canvas.
     setOnClickListener(new View.
     OnClickListener() {
   @Override
   public void onClick(View v) {
    //clear webview content
    clearWebView();
    Directories.delete(new File(Directories.
     EPUB_CACHE_DIR));

    //Hide current fragment and display canvas
    final FragmentTransaction ft = getActivity
     ().getSupportFragmentManager().
     beginTransaction();
    ft.show(getActivity().
     getSupportFragmentManager().
     findFragmentByTag("fragment_canvas"));
    ft.hide(getActivity().
     getSupportFragmentManager().
     findFragmentByTag("fragment_epub"));
    ft.commit();

    //Recreate EPUB table of contents
    Fragment_EPubViewer_TOC fragment = (
     Fragment_EPubViewer_TOC)getActivity().
     getSupportFragmentManager().
     findFragmentByTag("fragment_epub_toc");
    fragment.clearAdapter();

    //set state
    Activity_Main.state = Activity_Main.State.
     IS_CANVAS;
   }
  });

  return view;
 }

 public static void clearWebView() {
  //Free webview memory
  webView.stopLoading();
  webView.clearHistory();
  webView.clearCache(true);
  webView.clearFormData();
  webView.clearSslPreferences();
  webView.loadUrl("about:blank");
  //noinspection deprecation
  webView.freeMemory();
  //webView.pauseTimers();
  webView.removeAllViews();

  //delete epub cache dir.
  //Directories.delete(new File(Directories.
     EPUB_CACHE_DIR));
 }
}
```

Path = Tess2Speech/tess2Speech/src/main/java/anteraaron/tess2speech/Fragment_EPubViewer_TOC.java

```
package anteraaron.tess2speech;

import android.net.Uri;
import android.support.v4.app.
     FragmentTransaction;
import android.support.v4.app.ListFragment;
import android.content.Context;
import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.TextView;
import java.io.File;
import java.io.IOException;
import java.io.InputStream;
import java.util.ArrayList;
import java.util.List;
import nl.siegmann.epublib.domain.Book;
import nl.siegmann.epublib.domain.Resource;
import nl.siegmann.epublib.domain.TOCReference
     ;
import nl.siegmann.epublib.epub.EpubReader;

/**
 * A Fragment for displaying Epub table of
      contents which is embedded in the
      Main_Activity
 * @author Anter Aaron M. Custodio
 * @since 2016.3.27
 */
public class Fragment_EPubViewer_TOC extends
     ListFragment {

 private List<RowData> rowData;
 private String path;
 public static String htmlContent,
     adapterContent;

 /**
      * An Inner class that serves as the Epub
           element of the list. (Rows of every
           list).
      */
 private class RowData {
  private String title;
  private Resource resource;

  public RowData() {
   super();
  }

  //Get title of Epub
  public String getTitle() {
   return title;
  }

  //Get data of Epub
  public Resource getResource() {
   return resource;
  }

  //Set the title of the Epub
  public void setTitle(final String title) {
   this.title = title;
  }

  //Set the data of the Epub
  public void setResource(final Resource
     resource) {
   this.resource = resource;
  }

 }

 /**
      * A class that creates a Custom Adapter
           for displaying the list of Epubs.
      */
 private class CustomAdapter extends
     ArrayAdapter<RowData> {
  //Constructor
  public CustomAdapter(final Context context,
     final int resource, final int
     textViewResourceId, final List<RowData>
     objects) {
```

```
  super(context, resource, textViewResourceId
    , objects);
 }

 /**
        * An innerclass that serves as a
            holder for title and row data.
        */
 private class ViewHolder{
  private View row;
  private TextView titleHolder = null;
  //Constructor
  public ViewHolder(final View row) {
   super();
   this.row = row;
  }
  //Getter for title
  public TextView getTitle() {
   if(null == this.titleHolder) {
    this.titleHolder = (TextView) row.
    findViewById(R.id.row);
   }
   return titleHolder;
  }
 }

 @Override
 public View getView(final int position, View
     convertView, final ViewGroup parent) {
  //Returns row data and title when an
    adapter resource i.d is used.
  ViewHolder holder;
  TextView title;
  final RowData rowData = getItem(position);

  if(null == convertView){
   convertView = View.inflate(getActivity(),
    R.layout.view_epub_listview_row, null);
   holder = new ViewHolder(convertView);
   convertView.setTag(holder);
  }

  holder = (ViewHolder) convertView.getTag();
  title = holder.getTitle();
  title.setText(rowData.getTitle());
  return convertView;
 }

}

@Override
public void onCreate(final Bundle
    savedInstanceState) {
 super.onCreate(savedInstanceState);
}


@Override
public View onCreateView(final LayoutInflater
     inflater, final ViewGroup container,
    final Bundle savedInstanceState) {
 // Inflate the layout for this fragment
 return createUI(inflater.inflate(R.layout.
    view_epub_listview, container, false));
}

/**
     * Create UI for Epub Table of Contents
         Fragment
     * @param view View to be inflated.
     * @return returns the inflated View.
     */
private View createUI(final View view) {

 final Button button_epub_toc_back = (Button)
    view.findViewById(R.id.
    button_epub_toc_back);
 button_epub_toc_back.setOnClickListener(new
    View.OnClickListener() {
  @Override
  public void onClick(View v) {
   //Free-up webview memory
   Fragment_EPubViewer.webView.stopLoading();
   Fragment_EPubViewer.webView.clearHistory()
    ;
   Fragment_EPubViewer.webView.clearCache(
    true);
   Fragment_EPubViewer.webView.clearFormData
    ();
   Fragment_EPubViewer.webView.
    clearSslPreferences();
   Fragment_EPubViewer.webView.loadUrl("about
    :blank");
   //noinspection deprecation
   Fragment_EPubViewer.webView.freeMemory();
   //Fragment_EPubViewer.webView.pauseTimers
    ();
   Fragment_EPubViewer.webView.removeAllViews
    ();
   //delete epub cache dir.
   Directories.delete(new File(Directories.
    EPUB_CACHE_DIR));

   //Hide current fragment and display canvas
   final FragmentTransaction ft = getActivity
    ().getSupportFragmentManager().
    beginTransaction();
   ft.show(getActivity().
    getSupportFragmentManager().
    findFragmentByTag("fragment_canvas"));
   ft.hide(getActivity().
    getSupportFragmentManager().
    findFragmentByTag("fragment_epub_toc"));
   ft.commit();

   //Clear table of contents
   clearAdapter();

   //set state
   Activity_Main.state = Activity_Main.State.
    IS_CANVAS;
  }
 });
 return view;
}

@Override
public void onViewCreated (final View view,
    final Bundle savedInstanceState) {
 //When this view has focus, filter texts
    that the user types.
 getListView().setTextFilterEnabled(true);
}

/**
     * Method that creates the adapter (epub
         table of contents) of the list.
     */
public void createAdapter(final Uri epubUri,
    final String path) {
 if (epubUri != null) {
  this.path = path;
  this.rowData = new ArrayList<>();

  //Get book content.
  try {
   final InputStream epubInputStream =
    getActivity().getContentResolver().
    openInputStream(epubUri);
   final Book book = (new EpubReader()).
    readEpub(epubInputStream);
   if (epubInputStream != null) {
    epubInputStream.close();
   }
   logContentsTable(book.getTableOfContents()
    .getTocReferences(), 0);
  } catch (IOException e) {
   Log.e("epublib", e.getMessage());
  }

  //Create an adapter based on the content of
     the book.
  final CustomAdapter customAdapter = new
    CustomAdapter(getActivity(), R.layout.
    view_epub_listview_row, R.id.row, this.
    rowData);
  setListAdapter(customAdapter);
  customAdapter.notifyDataSetChanged();
 }
}

public void clearAdapter() {
 this.rowData.clear();
 setListAdapter(null);
}

/**
     *  A method that gets the chapters (data)
          based on the table of contents as
         reference.
     * @param tocReferences the table contents
         which is the reference.
     * @param depth determines the depth of
         the data-fetch (0 is the first
         grouping in table of contents, 1 is
         the subgroup of each 0 and so on).
     */
private void logContentsTable(final List<
    TOCReference> tocReferences, final int
    depth) {
 if (tocReferences == null) {
  return;
 }
 for (final TOCReference tocReference:
    tocReferences) {
  final StringBuilder tocString = new
    StringBuilder();
```

```
    //Appends tab on how deep the search is
    // i.e.  A.
    //          a.
    //            i.
    for (int i = 0; i < depth; i++) {
     tocString.append("\t");
    }

    //get the title of each chapters and data
      of each chapters
    tocString.append(tocReference.getTitle());

    final RowData row = new RowData();
    row.setTitle(tocString.toString());
    row.setResource(tocReference.getResource())
      ;
    this.rowData.add(row);
    adapterContent = row.getTitle() + "\n";

    logContentsTable(tocReference.getChildren()
      , depth + 1);
   }
  }

  @Override
  public void onListItemClick(final android.
      widget.ListView l, final View v, final
      int position, final long id) {
   super.onListItemClick(l, v, position, id);
   final RowData rowData = this.rowData.get(
      position);
   try {
    htmlContent = new String(rowData.
      getResource().getData());
    //Convert EPUB_CACHE_DIR to URL string
    String url = new File(this.path).toURI().
      toURL().toString() + File.separator + "
      OEBPS" + File.separator;
    Fragment_EPubViewer.webView.
      loadDataWithBaseURL(url, htmlContent, "
      text/html", "utf-8", null);

    //Hide current fragment and display
      fragment_epub
    final FragmentTransaction ft = getActivity
      ().getSupportFragmentManager().
      beginTransaction();
    ft.hide(getActivity().
      getSupportFragmentManager().
      findFragmentByTag("fragment_epub_toc"));
    ft.show(getActivity().
      getSupportFragmentManager().
      findFragmentByTag("fragment_epub"));
    ft.commit();

   } catch (IOException e) {
    Log.e("IO Exception", "Stack Trace:", e);
   }
  }
 }
```

Path = Tess2Speech/tess2Speech/src/main/java/anteraaron/tess2speech/Fragment_InputImage.java

```
package anteraaron.tess2speech;

import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;

/**
 * A Fragment for the Input Image which is
     embedded in the Main_Activity
 * @author Anter Aaron M. Custodio
 * @since 2016.3.22
 */
public class Fragment_InputImage extends
    Fragment {
 public static Button button_rotate,
    button_crop, button_back;
 @Override
 public View onCreateView(final LayoutInflater
     inflater, final ViewGroup container,
    final Bundle savedInstanceState) {
  // Inflate the layout for this fragment
  final View view = inflater.inflate(R.layout.
     fragment_input_image, container, false);
  return createUI(view);
 }

 /**
     * Create UI for Input Image Fragment
     * @param view View to be inflated.
     * @return returns the inflated View.
     */
 private View createUI(final View view){
  //Create rotate button
  button_rotate = (Button) view.findViewById(R
     .id.button_rotate);
  //Create crop button
  button_crop = (Button) view.findViewById(R.
     id.button_crop);
  //Create back button
  button_back = (Button) view.findViewById(R.
     id.button_back);
  return view;
 }
}
```

Path = Tess2Speech/tess2Speech/src/main/java/anteraaron/tess2speech/Fragment_PdfViewer.java

```
package anteraaron.tess2speech;

import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;

/**
 * A Fragment for the PDF viewer which is
     embedded in the Main_Activity
 * @author Anter Aaron M. Custodio
 * @since 2016.3.22
 */
public class Fragment_PdfViewer extends
    Fragment{
 public static TextView textView_pdf;
 public static Button button_pdf_back;
 @Override
 public View onCreateView(final LayoutInflater
     inflater, final ViewGroup container,
    final Bundle savedInstanceState) {
  //Inflate the layout for this fragment
  return createUI(inflater.inflate(R.layout.
     fragment_pdf_viewer, container, false));
 }

 /**
     * Create UI for PDF viewer Fragment
     * @param view View to be inflated.
     * @return returns the inflated View.
     */
 private View createUI(final View view){
  //Create textView in PDF viewer
  textView_pdf = (TextView)view.findViewById(R
     .id.textView_pdf);
  return view;
 }
}
```

Path = Tess2Speech/tess2Speech/src/main/java/anteraaron/tess2speech/Fragment_Settings.java

```
package anteraaron.tess2speech;

import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.preference.Preference;
import android.preference.PreferenceFragment;
import android.preference.PreferenceManager;
import android.util.Log;
import android.util.SparseBooleanArray;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;
import com.googlecode.tesseract.android.
    TessBaseAPI;
import java.io.File;
import java.io.FileFilter;

/**
 * A Fragment for the Preference Settings
     which is embedded in the Main_Activity
 * @author Anter Aaron M. Custodio
 * @since 2016.3.22
 */
public class Fragment_Settings extends
    PreferenceFragment {
 @Override
 public void onCreate(final Bundle
    savedInstanceState) {
  super.onCreate(savedInstanceState);
  addPreferencesFromResource(R.xml.preferences
    );
  //If there is no built in spell-checker,
    disable this preference.
  final Preference spellchecker =
    getPreferenceManager().findPreference("
    pref_key_spell_check");
  spellchecker.setEnabled(Activity_Settings.
    spellCheckerPresent);
  //On click listener of license preference.
  final Preference licenses =
    getPreferenceManager().findPreference("
    pref_key_license");
  licenses.setOnPreferenceClickListener(new
    Preference.OnPreferenceClickListener() {
   @Override
   public boolean onPreferenceClick(Preference
     preference) {
    //create alert dialog
    final AlertDialog.Builder dialogBuilder =
     new AlertDialog.Builder(getActivity());
    dialogBuilder.setTitle("Licenses");
    //inflate layout to alertDialog
    final View licensesView = View.inflate(
     getActivity(), R.layout.
     alert_dialog_licenses, null);
    dialogBuilder.setView(licensesView);
    dialogBuilder.setPositiveButton("Ok", new
     DialogInterface.OnClickListener() {
     @Override
     public void onClick(DialogInterface
     dialog, int which) {
      dialog.dismiss();
     }
    });
    //Display the created dialogBuilder
    final AlertDialog alertDialog =
     dialogBuilder.create();
    alertDialog.show();
    return false;
   }
  });
  //on click listener for tesseract language
    change preference.
  final Preference changeLanguage =
    getPreferenceManager().findPreference("
    pref_key_change_language");
  changeLanguage.setOnPreferenceClickListener(
    new Preference.OnPreferenceClickListener
    () {
   @Override
   public boolean onPreferenceClick(Preference
     preference) {
    //create alert dialog.
    final AlertDialog.Builder dialogBuilder =
     new AlertDialog.Builder(getActivity());
    dialogBuilder.setTitle("Select Trained
     Data");
    //inflate layout to alertDialog.
    final View checkedListView = View.inflate(
     getActivity(), R.layout.
     alert_dialog_change_language_listview,
     null);
    dialogBuilder.setView(checkedListView);
    //create list view
    final ListView listView = (ListView)
     checkedListView.findViewById(R.id.
     listView_change_language);
    //Get traineddata list.
    //noinspection ConstantConditions
    final File dir = new File(getActivity().
     getExternalFilesDir("tessdata").getPath()
     );
    final File[] list = dir.listFiles(new
     FileFilter() {
     @Override
     public boolean accept(File file) {
      //Only accepts .traineddata files.
      final String name = file.getName().
     toLowerCase();
      return name.endsWith(".traineddata") &&
     file.isFile();
     }
    });
    //Array of file names.
    final String[] names = new String[list.
     length];
    for (int i=0; i<names.length; i++) {
     names[i] = list[i].getName();
    }
    // set adapter for listview.
    final ArrayAdapter<String> adapter = new
     ArrayAdapter<>(getActivity(), R.layout.
     alert_dialog_change_language_listview_row
     , names);
    listView.setAdapter(adapter);
    listView.setItemsCanFocus(false);
    //Allow multiple choices.
    listView.setChoiceMode(ListView.
     CHOICE_MODE_MULTIPLE);
    //Check the currently used language.
    final String[] initLanguage = Libraries.
     tesseract.getInitLanguagesAsString().
     split("\\+");
    final int[] initLanguagePos = new int[
     initLanguage.length];

    int ctr = 0;
    //Find the position of the currently
     initialized language/s from the checkbox
     list.
    for (int i = 0; i < names.length; i++) {
     if (names[i].equals(initLanguage[ctr] + "
     .traineddata")) {
      initLanguagePos[ctr] = i;
      i = 0;
      ctr++;
      //All of initialized language/s has been
      compared
      if (initLanguage.length <= ctr) {
       break;
      }
     }
    }
    Log.e("tag", initLanguage.length + "");
    //Set all checkboxes to false
    for (int i = 0; i <names.length; i++) {
     listView.setItemChecked(i, false);
    }

    //Check the currently initialized language
     /s from the checkbox list.
    for (int initLanguagePosValue :
     initLanguagePos) {
     listView.setItemChecked(
     initLanguagePosValue, true);
    }

    //If the user pressed ok.
    dialogBuilder.setPositiveButton("Ok", new
     DialogInterface.OnClickListener() {
     @Override
     public void onClick(DialogInterface
     dialog, int which) {
      final SparseBooleanArray checkedItems =
     listView.getCheckedItemPositions();
      String language = "";
```

```
        for (int i=0; i<listView.getAdapter().
      getCount(); i++) {
          if (checkedItems.get(i)) {
            //If its only one, do not append +
      sign.
            if (language.equals("")) {
              language += listView.getAdapter().
      getItem(i);
            } else {
              language += "+" + listView.getAdapter
      ().getItem(i);
            }
          }
        }
        //Remove .traineddata extension.
        language = language.replace(".
      traineddata", "");
        //Save language to preference to save
      the value for next start-up.
        final SharedPreferences sharedPreference
       = PreferenceManager.
      getDefaultSharedPreferences(getActivity()
      );
        sharedPreference.edit().putString("
      language", language).apply();
        Libraries.tesseract = new TessBaseAPI();
        //noinspection ConstantConditions
        Libraries.tesseract.init(getActivity().
      getExternalFilesDir(null).getPath(),
      language);
        Libraries.tesseract.setVariable("
      classify_enable_learning","0");
        Libraries.tesseract.setVariable("
      classify_enable_adaptive_matcher","0");
        dialog.dismiss();
      }
    });
    //Launch the completed dialog.
    final AlertDialog alertDialog =
     dialogBuilder.create();
    alertDialog.show();
    alertDialog.setCanceledOnTouchOutside(
     false);

    listView.setOnItemClickListener(new
     AdapterView.OnItemClickListener() {
      @Override
      public void onItemClick(AdapterView<?>
     parent, View view, int position, long id)
      {
       if (listView.getCheckedItemCount() == 0)
       {
        alertDialog.getButton(AlertDialog.
      BUTTON_POSITIVE).setEnabled(false);
       } else {
        alertDialog.getButton(AlertDialog.
      BUTTON_POSITIVE).setEnabled(true);
       }
      }
     });
     return false;
    }
   });

   //On click listener for help preference.
   final Preference help = getPreferenceManager
      ().findPreference("pref_key_help");
   help.setOnPreferenceClickListener(new
      Preference.OnPreferenceClickListener() {
    @Override
    public boolean onPreferenceClick(Preference
       preference) {
     //Create alert dialog.
     final AlertDialog.Builder dialogBuilder =
     new AlertDialog.Builder(getActivity());
     dialogBuilder.setTitle("Help");
     //Inflate layout to alertDialog.
     final View licensesView = View.inflate(
      getActivity(), R.layout.alert_dialog_help
      , null);
     final TextView textView = (TextView)
      licensesView.findViewById(R.id.
      textView_help_directories_description);
     final String directories = "Images
      Directory: " + Directories.IMAGES_DIR + "
      \n\n"
       + "PDF and Text Directory: " +
      Directories.TEXT_DIR + "\n\n"
       + "Audio Directory: " + Directories.
      AUDIO_DIR;
     textView.setText(directories);

     dialogBuilder.setView(licensesView);
     dialogBuilder.setPositiveButton("Ok", new
      DialogInterface.OnClickListener() {
      @Override
      public void onClick(DialogInterface
     dialog, int which) {
       dialog.dismiss();
      }
     });
     //Display the created dialogBuilder.
     final AlertDialog alertDialog =
      dialogBuilder.create();
     alertDialog.show();
     return false;
    }
   });
  }
 }
```

Path = Tess2Speech/tess2Speech/src/main/java/anteraaron/tess2speech/

ImagePreProcess.java

```
package anteraaron.tess2speech;

import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.ColorMatrix;
import android.graphics.ColorMatrixColorFilter
    ;
import android.graphics.Paint;

/**
 * A class that contains different method for
      image pre-processing
 * @author Anter Aaron M. Custodio
 * @since 3.22.2016
 */
public class ImagePreProcess {
 /**
     * Converts a bitmap to grayscale.
     * @param orig the bitmap to be processed.
     * @return returns the grayscaled bitmap.
     */
 public static Bitmap toGrayScale(final Bitmap
      orig) {
  final int height = orig.getHeight();
  final int width = orig.getWidth();
  //Converts image to grayscale by setting
     matrix saturation to 0.
  final Bitmap processed = Bitmap.createBitmap
     (width, height, Bitmap.Config.ARGB_8888);
  final Canvas canvas = new Canvas(processed);
  final Paint paint = new Paint();
  final ColorMatrix matrix = new ColorMatrix()
     ;
  matrix.setSaturation(0);
  final ColorMatrixColorFilter filter = new
     ColorMatrixColorFilter(matrix);
  paint.setColorFilter(filter);
  canvas.drawBitmap(orig, 0, 0, paint);
  return processed;
 }

 /**
     * @param orig input bitmap
     * @return new bitmap
     */
 //contrast 0..10 1 is default
 //brightness -255..255 0 is default
 public static Bitmap
     changeBitmapContrastBrightness(final
     Bitmap orig) {
  final float contrast = 1.85f;
  final float brightness = 35;
  final ColorMatrix cm = new ColorMatrix(new
     float[]{
      contrast, 0, 0, 0, brightness,
      0, contrast, 0, 0, brightness,
      0, 0, contrast, 0, brightness,
      0, 0, 0, 1, 0
     });
```

```
  final Bitmap processed = Bitmap.createBitmap
    (orig.getWidth(), orig.getHeight(), orig.
    getConfig());

  final Canvas canvas = new Canvas(processed);

  final Paint paint = new Paint();
  paint.setColorFilter(new
     ColorMatrixColorFilter(cm));
  canvas.drawBitmap(orig, 0, 0, paint);

  return processed;
 }
}
```

Path = Tess2Speech/tess2Speech/src/main/java/anteraaron/tess2speech/Intent_BrowseFiles.java

```
package anteraaron.tess2speech;

import android.app.Activity;
import android.content.ContentResolver;
import android.content.Intent;
import android.net.Uri;
import android.os.Build;
import android.webkit.MimeTypeMap;
import java.io.File;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collections;

import kaloer.filepicker.FilePickerActivity;

/**
 * A class that contains different methods for
       browsing files of different type.
 * @author Anter Aaron M. Custodio
 * @since 2016.3.22
 */
public final class Intent_BrowseFiles {

 //List of acceptable file formats.
 private static final ArrayList<String>
     IMAGE_EXTENSIONS = new ArrayList<>(
     Collections.unmodifiableList(Arrays.
     asList(".png", ".jpg", ".jpeg", ".bmp", "
     .webp", ".gif")));
 private static final ArrayList<String>
     PDF_EXTENSION = new ArrayList<>(
     Collections.unmodifiableList(Collections.
     singletonList(".pdf")));
 private static final ArrayList<String>
     EPUB_EXTENSION = new ArrayList<>(
     Collections.unmodifiableList(Collections.
     singletonList(".epub")));

 /**
     * Launch an intent for browsing a single
         image file.
     * @return Uri of image file
     */
 public static Intent browseImage() {
  final Intent getterIntent = new Intent(
     Intent.ACTION_GET_CONTENT);
  getterIntent.addCategory(Intent.
     CATEGORY_OPENABLE);
  getterIntent.setType("image/*");

  // Only make these files visible (for built-
     in browser only android-file-
     picker_activity)
  getterIntent.putExtra(FilePickerActivity.
     EXTRA_ACCEPTED_FILE_EXTENSIONS,
     IMAGE_EXTENSIONS);

  return getterIntent;
 }

 /**
     * Launch an intent for browsing multiple
         image file
     * @return Uri of image files
     */
 public static Intent browseMultipleImages() {

  //Browse multiple image is not supported in
     API level below 18
  if (Build.VERSION.SDK_INT < Build.
     VERSION_CODES.JELLY_BEAN_MR2) {
   return browseImage();
  } else {
   //If API level > 18
   final Intent getterIntent = new Intent(
     Intent.ACTION_GET_CONTENT);
   getterIntent.addCategory(Intent.
     CATEGORY_OPENABLE);
   getterIntent.setType("image/*");

   // Only make these files visible (for built
     -in browser only android-file-
     picker_activity)\
   // Note that built in browser does not
     support multiple images
   getterIntent.putExtra(FilePickerActivity.
     EXTRA_ACCEPTED_FILE_EXTENSIONS,
     IMAGE_EXTENSIONS);

   //Allow multiple image selection.
   getterIntent.putExtra(Intent.
     EXTRA_ALLOW_MULTIPLE, true);

   return getterIntent;
  }
 }

 /**
     * Launch an intent for browsing PDF file
     * @return Uri of PDF file
     */
 public static Intent browsePdf() {
  final Intent getterIntent = new Intent(
     Intent.ACTION_GET_CONTENT);
  getterIntent.addCategory(Intent.
     CATEGORY_OPENABLE);
  getterIntent.setType("application/pdf");

  // Only make these files visible (for built-
     in browser only android-file-
     picker_activity)
  getterIntent.putExtra(FilePickerActivity.
     EXTRA_ACCEPTED_FILE_EXTENSIONS,
     PDF_EXTENSION);

  return getterIntent;
 }

 /**
     * Launch an intent for browsing Epub file
     * @return Uri of Epub file
     */
 public static Intent browseEpub() {
  final Intent getterIntent = new Intent(
     Intent.ACTION_GET_CONTENT);
  getterIntent.addCategory(Intent.
     CATEGORY_OPENABLE);
  getterIntent.setType("application/epub+zip")
     ;

  // Only make these files visible (for built-
     in browser only android-file-
     picker_activity)
  getterIntent.putExtra(FilePickerActivity.
     EXTRA_ACCEPTED_FILE_EXTENSIONS,
     EPUB_EXTENSION);

  return getterIntent;
 }

 /**
     * Checks if an image Uri is an accepted
         format.
     * @param imageUri the uri of the image.
     * @return true if the format of the image
         is acceptable.
     */
 public static boolean checkImageExtension (
     Activity activity, Uri imageUri) {
  //Get file extension from uri.
  final String extension = "." + getMimeType(
     activity, imageUri);

  return IMAGE_EXTENSIONS.contains(extension);
 }
```

```
    /**
     * Checks if a PDF Uri is an accepted
         format.
     * @param pdfUri the uri of the PDF.
     * @return true if the format of the PDF
         is acceptable.
     */
    public static boolean checkPdfExtension (
        Activity activity, Uri pdfUri) {
     //Get file extension from uri.
     final String extension = "." + getMimeType(
        activity, pdfUri);

     return PDF_EXTENSION.contains(extension);
    }

    /**
     * Checks if an Epub Uri is an accepted
         format.
     * @param epubUri the uri of the Epub.
     * @return true if the format of the Epub
         is acceptable.
     */
    public static boolean checkEpubExtension (
        Activity activity, Uri epubUri) {
     //Get file extension from uri.
     final String extension = "." + getMimeType(
        activity, epubUri);

     return EPUB_EXTENSION.contains(extension);
    }

    /**
     * Get mime type of the given uri.
     * @param uri the uri to be determined.
     * @return the mime type (extension) of
         the uri without "." i.e. (jpg, bmp,
         png).
     */
    public static String getMimeType(Activity
        activity, Uri uri) {
     String extension;

     //Check uri format to avoid null
     if (uri.getScheme().equals(ContentResolver.
        SCHEME_CONTENT)) {
      //If scheme is a content
      final MimeTypeMap mime = MimeTypeMap.
        getSingleton();
      extension = mime.getExtensionFromMimeType(
        activity.getContentResolver().getType(uri
        ));
     } else {
      //If scheme is a File
      //This will replace spaces with %20 and
        also other special characters. This will
        avoid returning null values on file name
        with spaces and special characters.
      extension = MimeTypeMap.
        getFileExtensionFromUrl(Uri.fromFile(new
        File(uri.getPath()))).toString());

     }

     return extension;
    }
   }
```

Path = Tess2Speech/tess2Speech/src/main/java/anteraaron/tess2speech/Intent_Camera.java

```
package anteraaron.tess2speech;

import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.net.Uri;
import android.provider.MediaStore;
import android.widget.Toast;

import java.io.File;
import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Locale;

/**
 * A class that launches the Camera Intent.
 * @author Anter Aaron M. Custodio
 * @since 2016.3.22
 */
public class Intent_Camera {
 private Context context;
 //Constructor
 public Intent_Camera(final Context context){
  this.context = context;
  try{
   //Create file location
   final File imageFile = createImageFile();
   Directories.imageFromCameraUri = Uri.
     fromFile(imageFile);
  }catch (IOException ex) {
   //Error in creating File
   Toast.makeText(this.context, "Error in
     creating file!", Toast.LENGTH_LONG).show
     ();
  }
 }

 /**
     * Create the camera intent to take
         picture.
     * @return returns the Camera Intent
     */
 public Intent takePicture(){
  //Check if an SD card is mounted, else close
      the app.
  if(Directories.checkExternalStorage()) {
   return dispatchTakePictureIntent();
  }else{
   //Display alert Dialog that SD card is not
     mounted
   final AlertDialog.Builder builder = new
     AlertDialog.Builder(context);
   builder.setTitle("SD Card not mounted");
   builder.setMessage("An SD Card is needed to
     run this application. Please insert or
     mount an SD card");
   builder.setCancelable(false);
   builder.setPositiveButton("Ok", new
     DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog
     , int which) {
     dialog.cancel();
    }
   });

   final AlertDialog dialog = builder.create()
     ;
   dialog.show();
   return null;
  }
 }

 /**
     * Continuation of takePicture() to create
         camera intent.
     * @return returns the Camera Intent
     */
 private Intent dispatchTakePictureIntent() {
  final Intent cameraIntent = new Intent(
     MediaStore.ACTION_IMAGE_CAPTURE);
  //Check for camera Activity to handle intent
  if(cameraIntent.resolveActivity(this.context
     .getPackageManager()) != null){
   cameraIntent.putExtra(MediaStore.
     EXTRA_OUTPUT, Directories.
     imageFromCameraUri);
   return cameraIntent;
  } else {
   return null;
  }
 }

 /**
     * Creates the file which will be the
         container of the image taken by the
         camera.
     * @return returns Image File with name as
         timestamp.
     * @throws IOException
     */
 private File createImageFile() throws
```

```
    IOException {
  //Create an image file name
  final String timeStamp = new
     SimpleDateFormat("yyyyMMdd_HHmmss",
     Locale.ENGLISH).format(new Date());
  final String imageName = "TS_C_" + timeStamp
     + "_";
  return new File(Directories.IMAGES_DIR +
     File.separator + imageName + ".jpg");
 }
}
```

Path = Tess2Speech/tess2Speech/src/main/java/anteraaron/tess2speech/

Libraries.java

```
package anteraaron.tess2speech;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.net.Uri;
import android.os.Build;
import android.preference.PreferenceManager;
import android.speech.tts.TextToSpeech;
import android.speech.tts.
    UtteranceProgressListener;
import android.view.View;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.Toast;
import com.googlecode.tesseract.android.
    TessBaseAPI;
import com.joanzapata.pdfview.PDFView;
import com.joanzapata.pdfview.listener.
    OnPageChangeListener;
import com.polites.android.GestureImageView;
import org.apache.commons.io.FileUtils;
import java.io.File;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Locale;

/**
 * Class that contains all the libraries used
     in the Tess2Speech application
 * @author Anter Aaron M. Custodio
 * @since 3.22.2016
 */
public class Libraries implements TextToSpeech
    .OnInitListener{

 public static TextToSpeech tts; //Text to
     Speech (Android Built in)
 public static TessBaseAPI tesseract;// Image
     to Text Library
 public static PDFView pdfView; // Library to
     view PDF file to image view
 public static GestureImageView imageView; //
     Library for zoomable imageView
 private static Activity activity;
 private static String path; //container of
     file paths
 private static String pdfPath; //File paths
     of the pdfs
 private static String dir; //File path for
     converted pdf to image directory
 private static boolean play;//Determines if
     PdfToText is called with the intent to
     play
 private static boolean toImage;//Determines
     if PdfToText is called with the intent to
      play
 private static ArrayList<Integer> pages;

 //Constructor
 public Libraries (Activity activity_parameter
     ) {
  activity = activity_parameter;
  //Instantiate Tesseract
  tesseract = new TessBaseAPI();
  //Initialize Tesseract's language
  final SharedPreferences sharedPreference =
     PreferenceManager.
     getDefaultSharedPreferences(activity);
  //noinspection ConstantConditions
  tesseract.init(activity.getExternalFilesDir(
     null).getPath(), sharedPreference.
     getString("language", "eng+engh"));
  tesseract.setVariable("
     classify_enable_learning", "0");
  tesseract.setVariable("
     classify_enable_adaptive_matcher", "0");
  //Initialize Text to Speech
  tts = new TextToSpeech(activity, this);

 }

 /**
     * Uses Android PDF-View Library to
         display PDF to imageView.
     * @param uri PDF to be displayed's uri.
     */
 public static void displayPdf(final Uri uri){
  //Workaround to access file via uri
  new Task_FileFromUri(activity, "cache.pdf",
     new Task_FileFromUri.AsyncResponse() {
   @Override
   public void processFinished(File file) {
    pdfPath = file.getPath();
    pdfView.fromFile(file)
      .defaultPage(1)
      .showMinimap(false)
      .onPageChange(new OnPageChangeListener()
     {
       @Override
       public void onPageChanged(int page, int
      pageCount) {
         //Display current page and page number
         String currentPage = page + " / " +
     pageCount;
         Fragment_PdfViewer.textView_pdf.
     setText(currentPage);
       }
      })
      .enableSwipe(true)
      .load();
    pdfView.setVerticalScrollBarEnabled(true);
   }
  }).execute(uri);
 }

 /**
     * Uses VuDroid Library (this is included
         in PDFView library) to convert pdf
         to image
     * @param uri PDF's uri
     */
 public static void pdfToImage (final Uri uri)
     {

  displayPdf(uri);

  toImage = true;
  final EditText editText_input = new EditText
     (activity);

  //create alert dialog
  final AlertDialog.Builder dialogBuilder =
     new AlertDialog.Builder(activity);
  dialogBuilder.setTitle("Create Folder");
  dialogBuilder.setMessage("Folder name where
     converted images will be saved: ");
  //Add edit text to alert dialog
  final LinearLayout.LayoutParams lp = new
     LinearLayout.LayoutParams(LinearLayout.
     LayoutParams.MATCH_PARENT, LinearLayout.
     LayoutParams.MATCH_PARENT);
  editText_input.setLayoutParams(lp);
  dialogBuilder.setView(editText_input);

  //Set action for alert dialog
  dialogBuilder.setPositiveButton("Ok", new
     DialogInterface.OnClickListener() {
   @Override
   public void onClick(DialogInterface dialog,
      int which) {
    createFolder(editText_input.getText().
     toString());
   }
  });
  dialogBuilder.setNegativeButton("Cancel",
```

```
    new DialogInterface.OnClickListener() {
  @Override
  public void onClick(DialogInterface dialog,
     int which) {
   dialog.dismiss();
   Toast.makeText(activity, "Convert to image
     cancelled", Toast.LENGTH_LONG).show();
  }
 });

 //Show the created dialogBuilder.
 final AlertDialog alertDialog =
    dialogBuilder.create();
 alertDialog.show();
}

/**
     * Uses VuDroid Library (this is included
         in PDFView library) to convert pdf
         to image, then uses tesseract to
         recognize texts
     * @param play_param Play the converted
         image if this parameter is set to
         true.
     */
public static void pdfToText(final boolean
    play_param){
 toImage = false;
 play = play_param;
 path = pdfPath;
 selectPageDialog();
}

/**
     * Creates a folder in which the converted
          pages of the pdf to image will be
         stored.
     * @param folderName Name of the folder
         that will be created
     */
private static void createFolder(final String
     folderName) {

 try {
  //Set file directory
  dir = Directories.IMAGES_DIR + File.
    separator + folderName;
  final File file = new File(dir);
  //If file exists, prompt overwrite dialog
    box
  if (file.exists() && file.isDirectory()) {
   new AlertDialog.Builder(activity)
     .setTitle("Folder already exist")
     .setMessage("Files inside the folder
    will be overwritten. Overwrite?")
       //Ok button, Overwrite existing file
     .setPositiveButton("Yes", new
    DialogInterface.OnClickListener() {
       public void onClick(DialogInterface
    dialog, int which) {
        path = pdfPath;
        selectPageDialog();
       }
      })
       //Cancel Button, dismiss dialog box
     .setNegativeButton("No", new
    DialogInterface.OnClickListener() {
       public void onClick(DialogInterface
    dialog, int which) {
        dialog.dismiss();
        Toast.makeText(activity, "Convert to
    image cancelled", Toast.LENGTH_LONG).show
    ();
       }
      })
     .show()
     .setCanceledOnTouchOutside(false);
  } else {
   //noinspection ResultOfMethodCallIgnored
   file.mkdirs();
   selectPageDialog();
  }
 } catch (Exception e) {
  //An error has occured while saving
  Toast.makeText(activity, "Error in
    Converting PDF to image!", Toast.
    LENGTH_SHORT).show();
 }
}

/**
     * Creates a dialog that prompts the user
         to select which pages they would
         like to process.
     */
private static void selectPageDialog() {
 //Create alert dialog.
 final AlertDialog.Builder dialogBuilder =
    new AlertDialog.Builder(activity);
 dialogBuilder.setTitle("Convert PDF to Text"
    );
 dialogBuilder.setMessage("Pages to Convert:
    ");
 final View pageSelectView = View.inflate(
    activity, R.layout.
    alert_dialog_page_select, null);
 dialogBuilder.setView(pageSelectView);
 //Create radio group.
 final RadioGroup radioGroup = (RadioGroup)
    pageSelectView.findViewById(R.id.
    radioGroup_pdf);
 radioGroup.setOnCheckedChangeListener(new
    RadioGroup.OnCheckedChangeListener() {
  @Override
  public void onCheckedChanged(RadioGroup
    group, int checkedId) {
   final EditText editText_pdf = (EditText)
    pageSelectView.findViewById(R.id.
    editText_pdf);
   if (checkedId == R.id.radioButton_pdf_all)
     {
    //All radio Button
    editText_pdf.setEnabled(false);
    editText_pdf.setHint("");
    editText_pdf.setFocusableInTouchMode(
    false);
    editText_pdf.clearFocus();
   } else if (checkedId == R.id.
    radioButton_pdf_select) {
    //Specify pages radio button.
    editText_pdf.setEnabled(true);
    editText_pdf.setHint("Pages e.g. 1,2,5,12
    ");
    editText_pdf.setFocusableInTouchMode(true
    );
    editText_pdf.requestFocus();
   }
  }
 });
 //Set action for alert dialog.
 dialogBuilder.setPositiveButton("Ok", new
    DialogInterface.OnClickListener() {
  @Override
  public void onClick(DialogInterface dialog,
     int which) {
   processInput(pageSelectView);
   pdfToImageProcess();
  }
 });
 //Set action for cancel.
 dialogBuilder.setNegativeButton("Cancel",
    new DialogInterface.OnClickListener() {
  @Override
  public void onClick(DialogInterface dialog,
     int which) {
   dialog.dismiss();
   Toast.makeText(activity, "Convert
    cancelled", Toast.LENGTH_LONG).show();
  }
 });

 //Show the created dialog builder.
 final AlertDialog alertDialog =
    dialogBuilder.create();
 alertDialog.setCanceledOnTouchOutside(false)
    ;
 alertDialog.show();
}

/**
     * Process user input on pages to be
         converted.
     * @param pageSelectView PageSelectView
         that contains the radioButtons.
     */
private static void processInput (final View
    pageSelectView) {

 if (((RadioButton)pageSelectView.
    findViewById(R.id.radioButton_pdf_all)).
    isChecked()) {
  //If all is checked, save all the pages to
    arraylist.
  pages = new ArrayList<>();
  for (int i=0; i<pdfView.getPageCount(); i
    ++) {
   pages.add(i);
  }
 } else {
  //Else, save only the specified pages to
    arraylist.

  //Split text inputted by user by comma
  final EditText editText_pdf =(EditText)
    pageSelectView.findViewById(R.id.
    editText_pdf);
  final String[] pagesString = editText_pdf.
    getText().toString().split("\\s*,\\s*");
```

```
  //Convert pages to int
  pages = new ArrayList<>();
  for (String page : pagesString) {
   if (!page.equals("") && Integer.parseInt(
    page) > 0) {
    pages.add(Integer.parseInt(page) - 1);
   }
  }
  //Remove pages which are greater than the
    max # of pages
  final int max = pdfView.getPageCount();
  for (int i=0; i<pages.size(); i++) {
   if (pages.get(i) + 1 > max) {
    pages.remove(i);
    i = 0;
   }
  }
 }

}

/**
     * Continuation of PdfToImage(). The
         process behind conversion of Pdf to
         image.
     */
private static void pdfToImageProcess(){
 if (pages.size() > 0) {
  //While there are pages, set switch append
    to true.
  Fragment_ConvertedText.switch_append.
    setChecked(true);
  //Calls AsyncTask that converts PdfToImage
  new Task_PdfToImage(activity, new
    Task_PdfToImage.AsyncResponse() {
   @Override
   public void processFinished(File file) {
    if(pages.size() == 1) {
     if (toImage) {
      //If the user selects only to image, do
     not convert to text.
      saveConvertedImage(file, pages.get(0) +
     1);
     } else {
      //Else convert it to text and append to
     previous pages.
      new ToText(activity).execute(Uri.
    fromFile(file), play);
     }
     //Display where the converted pdf to
    image is saved
     if (dir != null && !dir.isEmpty()) {
      Toast.makeText(activity, "Images saved
    to: " + dir, Toast.LENGTH_LONG).show();
     }
    } else {
     if (toImage) {
      saveConvertedImage(file, pages.get(0) +
     1);
     } else {
      new ToText(activity).execute(Uri.
    fromFile(file), false);
     }
    }
    pages.remove(0);
    pdfToImageProcess();
    Fragment_ConvertedText.switch_append.
    setChecked(false);

   }
  }).execute(path, pages.get(0));
 }
}

/**
     * Save the converted image from cache to
         the created folder.
     * @param fileSource the file path where
         it was
     */
private static void saveConvertedImage (final
     File fileSource, int page) {
 int ctr = 0;
 File fileDestination = new File(dir + File.
    separator + "Page " + page + ".png");
 //If file name exists, iterate ctr to make
    file name unique.
 while (fileDestination.exists()) {
  ctr++;
  fileDestination = new File(dir + File.
    separator + "Page " + page + "(" + ctr +
    ")" + ".png");
 }
 try {
  //Copy file from cache to save dir using
    Apache Commons io
  FileUtils.copyFile(fileSource,
    fileDestination);
  galleryAddPic(Uri.fromFile(fileDestination)
    );
 } catch (IOException e) {
  e.printStackTrace();
 }
}

/**
     * Send broadcast to the gallery to show
         the taken picture in the gallery.
     * @param uri uri of the image that will
         be added to gallery.
     */
public static void galleryAddPic(final Uri
    uri){
 if(Build.VERSION.SDK_INT >= Build.
    VERSION_CODES.KITKAT){
  //For kitkat and newer version
  final Intent intent = new Intent(Intent.
    ACTION_MEDIA_SCANNER_SCAN_FILE, uri);
  activity.sendBroadcast(intent);
 } else {
  //For Jellybean and below
  activity.sendBroadcast(new Intent(Intent.
    ACTION_MEDIA_MOUNTED, uri));
 }
}

/**
     * Initialize text to speech library.
     * @param status Status of the TTS. Euther
         Success or fail.
     */
@Override
public void onInit(final int status) {
 if (status == TextToSpeech.SUCCESS){
  tts.setOnUtteranceProgressListener(new
    UtteranceProgressListener() {
   @Override
   public void onStart(String utteranceId) {
      activity.runOnUiThread(new Runnable() {
       @Override
       public void run() {
        //set button label to stop
        Activity_Main.button_play.
    setCompoundDrawablesWithIntrinsicBounds
    (0, R.mipmap.ic_action_playback_stop, 0,
    0);
        Activity_Main.button_play.setText(R.
    string.button_stop);

        //Disable buttons to avoid button
    mashing
        Activity_Main.button_text.setEnabled(
    false);
        Activity_Main.button_text.
    setClickable(false);
       }
      });
   }

   @Override
   public void onDone(String utteranceId) {
    activity.runOnUiThread(new Runnable() {
     @Override
     public void run() {
      //Change button to play after TTS has
    stopped playing
      Activity_Main.button_play.
    setCompoundDrawablesWithIntrinsicBounds
    (0, R.mipmap.ic_action_playback_play, 0,
    0);
      Activity_Main.button_play.setText(R.
    string.button_play);

      //Re-enable buttons
      Activity_Main.button_text.setEnabled(
    true);
      Activity_Main.button_text.setClickable(
    true);
     }
    });
   }

   @Override
   public void onError(String utteranceId) {

   }
  });
  final int result = tts.setLanguage(Locale.
    US);
  //check if language is supported
  if (result == TextToSpeech.
    LANG_MISSING_DATA || result ==
    TextToSpeech.LANG_NOT_SUPPORTED){
   Toast.makeText(activity, "Language not
    supported", Toast.LENGTH_LONG).show();
  }
 }else{
  Toast.makeText(activity, "Initialization
```

```
    Failed", Toast.LENGTH_LONG).show();
   activity.finish();
   tts.shutdown();
  }
```

```
 }
}
```

Path = Tess2Speech/tess2Speech/src/main/java/anteraaron/tess2speech/Task_CopyAssets.java

```
package anteraaron.tess2speech;

import android.app.Activity;
import android.content.res.AssetManager;
import android.os.AsyncTask;
import android.util.Log;
import android.view.View;
import android.view.animation.AlphaAnimation;
import android.view.animation.Animation;
import android.widget.TextView;
import android.widget.Toast;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;

/**
 * An AsyncTask that copies the files from the
      asset folder to the SD card.
 * @author Anter Aaron M. Custodio
 * @since 2016.3.22
 */
public class Task_CopyAssets extends AsyncTask
    <Void, Void, Void> {
 private static Activity activity;
 private AsyncResponse delegate;
 private TextView textView_splash;
 //Create an interface to determine that the
    AsyncTask is finished.
 public interface AsyncResponse {
  void processFinished();
 }

 public Task_CopyAssets(final Activity
    activity, final AsyncResponse delegate){
  Task_CopyAssets.activity = activity;
  this.delegate = delegate;
  //Create the loading text in splash screen.
  this.textView_splash = (TextView) activity.
    findViewById(R.id.textView_splash);
 }

 @Override
 protected void onPreExecute(){
  //Make TextView blink while copying files.
  this.textView_splash.setVisibility(View.
    VISIBLE);
  final Animation anim = new AlphaAnimation
    (0.0f, 1.0f);
  anim.setDuration(500); //You can manage the
    blinking time with this parameter
  anim.setStartOffset(20);
  anim.setRepeatMode(Animation.REVERSE);
  anim.setRepeatCount(Animation.INFINITE);
  //Start blinking animation
  this.textView_splash.startAnimation(anim);
 }

 @Override
 protected Void doInBackground(final Void...
    params) {
  start();
  return null;
 }

 @Override
 protected void onPostExecute(final Void param
    ) {
  //End animation when copying is finished.
  this.textView_splash.clearAnimation();
  this.textView_splash.setVisibility(View.
    INVISIBLE);
  //Use interface when process is finished.
  this.delegate.processFinished();
  super.onPostExecute(param);
 }

 /**
     * Method that starts and performs the
          copying of the asset file to
          tessdata folder
     */
 private static void start() {
  final AssetManager assetManager = activity.
    getAssets();
  String[] files = null;
  //Load assets list.
  try{
   files = assetManager.list("");
  }catch (IOException e){
   Toast.makeText(activity, "Failed to load
     asset list", Toast.LENGTH_SHORT).show();
  }

  if (files != null){
   //Copy files inside the asset folder to
     tessdata folder
   for (final String filename : files) {
    InputStream in = null;
    OutputStream out = null;

    try {
     //only copy files that has .traineddata
     in file name
     if(filename.contains(".traineddata")) {
      in = assetManager.open(filename);
      final File outFile = new File(activity.
     getExternalFilesDir("tessdata"), filename
     );
      out = new FileOutputStream(outFile);
      copyFile(in, out);
     }
    } catch (IOException e) {
     Toast.makeText(activity, "Failed to copy
     asset list: " + filename, Toast.
     LENGTH_SHORT).show();
    } finally {

     if (in != null) {
      try {
       in.close();
      } catch (IOException e) {
       Log.e("IO Exception", "Stack Trace:", e
     );
      }
     }
     if (out != null) {
      try {
       out.close();
      } catch (IOException e) {
       Log.e("IO Exception", "Stack Trace:", e
     );
      }
     }

    }
   }
  }
 }

 /**
     * The process of copying a single file.
     * @param in InputStream
     * @param out OutputStream
     * @throws IOException
     */
 private static void copyFile(final
    InputStream in, final OutputStream out)
    throws IOException {
  final byte[] buffer = new byte[1024];

  int read;
  while((read = in.read(buffer)) != -1){
   out.write(buffer, 0, read);
  }
 }
}
```

Path = Tess2Speech/tess2Speech/src/main/java/anteraaron/tess2speech/Task_FileFromUri.java

```
package anteraaron.tess2speech;

import android.app.Activity;
import android.app.ProgressDialog;
import android.content.pm.ActivityInfo;
import android.net.Uri;
import android.os.AsyncTask;
import android.util.Log;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;

/**
 * A AsyncTask which is a workaround for
      getting a File from Uri.
 * Saves the file from uri to a cache and
      returns the file object via interface.
 * @author Anter Aaron M. Custodio
 * @since 2016.3.22
 */
public class Task_FileFromUri extends
     AsyncTask<Object, Void, File> {
 private Activity activity;
 private AsyncResponse delegate;
 private String cacheName;
 private ProgressDialog progressDialog;
 //Create an interface to determine that the
     AsyncTask is finished.
 public interface AsyncResponse {
  void processFinished(final File file);
 }
 //Constructor
 public Task_FileFromUri(final Activity
     activity, final String cacheName, final
     AsyncResponse delegate) {
  this.activity = activity;
  this.cacheName = cacheName;
  this.delegate = delegate;
 }
 @Override
 protected void onPreExecute(){
  activity.setRequestedOrientation(
     ActivityInfo.SCREEN_ORIENTATION_NOSENSOR)
     ;
  //Start Progress Dialog
  createDialog();
 }

 @Override
 protected File doInBackground(final Object...
      params) {
  //Creates file in cache directory in which
     the file from uri will be saved
  final File file = new File(this.activity.
     getCacheDir() + File.separator + this.
     cacheName);
  FileOutputStream out;
  InputStream in;
  try {
   out = new FileOutputStream(file, false);
   //get uri file via input stream
   in = activity.getContentResolver().
     openInputStream((Uri)params[0]);
   //copy file
   final byte[] buffer = new byte[1024];
   assert in != null;
   int length = in.read(buffer);
   while (length != -1) {
    out.write(buffer, 0, length);
    length = in.read(buffer);
   }
   out.close();
  } catch (IOException e) {
   Log.e("IO Exception", "Stack Trace:", e);
  }
  return file;
 }

 @Override
 protected void onPostExecute(final File file)
     {
  super.onPostExecute(file);
  //cancel the progress dialog after creating
     cache.
  this.delegate.processFinished(file);
  activity.setRequestedOrientation(
     ActivityInfo.
     SCREEN_ORIENTATION_UNSPECIFIED);
  this.progressDialog.cancel();
  this.progressDialog.dismiss();
 }
 /**
      * Method that creates ProgressDialog
          while there is a background task.
      */
 private void createDialog(){
  this.progressDialog = new ProgressDialog(
     this.activity);
  this.progressDialog.setCancelable(false);
  this.progressDialog.
     setCanceledOnTouchOutside(false);
  this.progressDialog.setMessage("Opening File
     ...");
  this.progressDialog.show();
 }
}
```

Path = Tess2Speech/tess2Speech/src/main/java/anteraaron/tess2speech/Task_PdfToImage.java

```
package anteraaron.tess2speech;

import android.app.Activity;
import android.app.ProgressDialog;
import android.content.pm.ActivityInfo;
import android.graphics.Bitmap;
import android.graphics.RectF;
import android.os.AsyncTask;
import android.util.Log;
import org.vudroid.pdfdroid.codec.PdfContext;
import org.vudroid.pdfdroid.codec.PdfDocument;
import org.vudroid.pdfdroid.codec.PdfPage;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;

/**
 * An AsyncTask that converts PDF to Image
      using VuDroid from android-pdfView (See
      .. https://github.com/JoanZapata/android
     -pdfview).
 * @author Anter Aaron M. Custodio
 * @since 2016.3.22
 */
public class Task_PdfToImage extends AsyncTask
     <Object, Void, Void> {
 private Activity activity;
 private File file;
 private Bitmap bitmap;
 private AsyncResponse delegate;
 private ProgressDialog progressDialog;
 //Create an interface to determine that the
     AsyncTask is finished.
 public interface AsyncResponse {
  void processFinished(final File file);
 }
 public Task_PdfToImage(final Activity
     activity, final AsyncResponse delegate) {
  this.activity = activity;
  this.delegate = delegate;
 }

 @Override
 protected void onPreExecute(){
  activity.setRequestedOrientation(
     ActivityInfo.SCREEN_ORIENTATION_NOSENSOR)
```

```
    ;
  //Start Progress Dialog
  createDialog();
 }

 @Override
 protected Void doInBackground(final Object...
      param) {
  //Initialize PdfContext
  final PdfContext pdfContext = new PdfContext
     ();
  final PdfDocument pdf = (PdfDocument)
     pdfContext.openDocument((String) param
     [0]);

  //Choose page number to convert to image.
     doInBackground accepts (filePathOfPdf,
     page#)
  final PdfPage page = (PdfPage)pdf.getPage((
     int)param[1]);
  //Render the Pdf page to bitmap
  final RectF rect = new RectF();
  rect.bottom = rect.right = (float)1.0;
  this.bitmap = page.renderBitmap(page.
     getWidth(), page.getHeight(), rect);

  //Save generated image to cache
  this.file = new File(this.activity.
     getCacheDir() + "/cache.png");

  try {
   final FileOutputStream out = new
     FileOutputStream(this.file);
   this.bitmap.compress(Bitmap.CompressFormat.
     PNG, 100, out);

   out.flush();
   out.close();
   pdf.recycle();
   pdfContext.recycle();
  } catch (IOException e) {
   Log.e("IO Exception", "Stack Trace:", e);
  }
  return null;
 }

 @Override
 protected void onPostExecute(final Void param
     ) {
  super.onPostExecute(param);
  //Recycle bitmap to free up space.
  this.bitmap.recycle();
  //return the created image File
  this.delegate.processFinished(this.file);
  activity.setRequestedOrientation(
     ActivityInfo.
     SCREEN_ORIENTATION_UNSPECIFIED);
  //cancel the progress dialog after saving
     image.
  this.progressDialog.cancel();

 }

 /**
     * Method that creates ProgressDialog
         while there is a background task.
     */
  private void createDialog(){
   this.progressDialog = new ProgressDialog(
     this.activity);
   this.progressDialog.setCancelable(false);
   this.progressDialog.
     setCanceledOnTouchOutside(false);
   this.progressDialog.setMessage("Converting
     PDF...");
   this.progressDialog.show();
  }
 }
```

Path = Tess2Speech/tess2Speech/src/main/java/anteraaron/tess2speech/Task_SaveAudio.java

```
package anteraaron.tess2speech;

import android.app.Activity;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.content.pm.ActivityInfo;
import android.os.AsyncTask;
import android.os.Build;
import android.speech.tts.TextToSpeech;
import android.speech.tts.
     UtteranceProgressListener;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.Toast;

import java.io.File;
import java.util.ArrayList;
import java.util.HashMap;

/**
 * An AsyncTask that saves Text-to-Speech
     output.
 * @author Anter Aaron M. Custodio
 * @since 2016.3.22
 */
public class Task_SaveAudio extends AsyncTask<
     Void, Void, Void> {

 private EditText editText_input;
 private ProgressDialog progressDialog;
 private String string, path;
 private Activity activity;
 private static boolean finished;
 private static int counter;
 private static ArrayList<String>
     splittedTexts;
 private static final int MAX_CHARS = 3900;

 public Task_SaveAudio(final Activity activity
     ){
  this.activity = activity;
  this.editText_input = new EditText(activity)
     ;
  this.string = Fragment_ConvertedText.
     editText.getText().toString();
  splittedTexts = null;
  counter = 1;
  //Create contents of alert dialog
  final AlertDialog.Builder dialogBuilder =
     new AlertDialog.Builder(activity);
  dialogBuilder.setTitle("Save Converted
     Speech");
  dialogBuilder.setMessage("Input file name: "
     );

  //Add edit text to alert dialog.
  final LinearLayout.LayoutParams lp = new
     LinearLayout.LayoutParams(LinearLayout.
     LayoutParams.MATCH_PARENT, LinearLayout.
     LayoutParams.MATCH_PARENT);
  this.editText_input.setLayoutParams(lp);
  dialogBuilder.setView(this.editText_input);

  //Set action for alert dialog.
  dialogBuilder.setPositiveButton("Ok", new
     DialogInterface.OnClickListener() {
   @Override
   public void onClick(DialogInterface dialog,
      int which) {
    save();
   }
  });
  //Set cancel button for alert dialog
  dialogBuilder.setNegativeButton("Cancel",
     new DialogInterface.OnClickListener() {
   @Override
   public void onClick(DialogInterface dialog,
      int which) {
    dialog.dismiss();
    Toast.makeText(activity, "Save cancelled",
      Toast.LENGTH_LONG).show();
   }
  });
  //Show the created alert Dialog
  final AlertDialog alertDialog =
     dialogBuilder.create();
  alertDialog.show();
 }

 @Override
 protected Void doInBackground(final Void...
```

```
    params) {
 //save the audio in background
 saveAudio(this.path, this.string);
 return null;
}

@Override
protected void onPostExecute(final Void param
    ) {
 super.onPostExecute(param);
 activity.setRequestedOrientation(
    ActivityInfo.
    SCREEN_ORIENTATION_UNSPECIFIED);
 //cancel the progress dialog after saving
    audio.
 this.progressDialog.cancel();
 Toast.makeText(this.activity, "File saved to
    : " + this.path, Toast.LENGTH_LONG).show
    ();
}

/**
     * Method that saves the audio output by
         TTS.
     */
private void save(){
 //Create file for saving
 final String filename = this.editText_input.
    getText() + ".wav";
 this.path = Directories.AUDIO_DIR + File.
    separator + filename;
 final File file = new File(path);
 //If file exists, create a dialog that
    prompts the user to overwrite file
 if (file.exists()) {
  new AlertDialog.Builder(this.activity)
    .setTitle("File already exist")
    .setMessage("Do you want to overwrite the
     existing file?")

    //Ok button, Overwrite existing file
    .setPositiveButton("Yes", new
    DialogInterface.OnClickListener() {
      public void onClick(DialogInterface
    dialog, int which) {
        activity.setRequestedOrientation(
    ActivityInfo.SCREEN_ORIENTATION_NOSENSOR)
    ;
        createDialog();
        execute();
      }
    })

    //Cancel Button, dismiss dialog box
    .setNegativeButton("No", new
    DialogInterface.OnClickListener() {
      public void onClick(DialogInterface
    dialog, int which) {
        dialog.dismiss();
        Toast.makeText(activity, "Save
    cancelled", Toast.LENGTH_LONG).show();
      }
    })
    .show();
 } else {
  //Do not prompt overwrite
  createDialog();
  execute();
 }
}

/**
     * Saves the audio output of TTS.
     * @param path file path in which the
         audio file will be saved.
     * @param string The text to be converted
         to audio.
     */
@SuppressWarnings("deprecation")
private void saveAudio(final String path,
    final String string){
 if (Build.VERSION.SDK_INT >= Build.
    VERSION_CODES.LOLLIPOP) {
  //for depricated (lollipop and higher)
  if(string.length() >= MAX_CHARS) {
   splittedTexts = splitText(string);
   Libraries.tts.synthesizeToFile(
    splittedTexts.get(0), null, new File(path
    .substring(0, path.lastIndexOf('.')) + "-
    part-" + counter + ".wav"), "tts");
  } else {
   Libraries.tts.synthesizeToFile(string,
    null, new File(path), "tts");
  }


 } else {
  final HashMap<String, String> hashRender =
   new HashMap<>();
  //for non depricated (lower than lollipop)
  if (string.length() >= MAX_CHARS) {
   splittedTexts = splitText(string);
   hashRender.put(TextToSpeech.Engine.
    KEY_PARAM_UTTERANCE_ID, splittedTexts.get
    (0));
   Libraries.tts.synthesizeToFile(
    splittedTexts.get(0), hashRender, path.
    substring(0, path.lastIndexOf('.')) + "-
    part-" + counter + ".wav");
  } else {
   hashRender.put(TextToSpeech.Engine.
    KEY_PARAM_UTTERANCE_ID, string);
   Libraries.tts.synthesizeToFile(string,
    hashRender, path);
  }
 }

 Libraries.tts.setOnUtteranceProgressListener
    (new UtteranceProgressListener() {
  @Override
  public void onStart(String utteranceId) {
   finished = false;
  }

  @Override
  public void onDone(String utteranceId) {
   counter++;
   if (splittedTexts != null && splittedTexts
    .size() > 1) {
    splittedTexts.remove(0);
    if (Build.VERSION.SDK_INT >= Build.
    VERSION_CODES.LOLLIPOP) {
     Libraries.tts.synthesizeToFile(
    splittedTexts.get(0), null, new File(path
    .substring(0, path.lastIndexOf('.')) + "-
    part-" + counter + ".wav"), "tts");
    } else {
     final HashMap<String, String> hashRender
     = new HashMap<>();
     hashRender.put(TextToSpeech.Engine.
    KEY_PARAM_UTTERANCE_ID, splittedTexts.get
    (0));
     Libraries.tts.synthesizeToFile(
    splittedTexts.get(0), hashRender, path.
    substring(0, path.lastIndexOf('.')) + "-
    part-" + counter + ".wav");
    }
   } else {
    finished = true;
   }
  }

  @Override
  public void onError(String utteranceId) {
   Toast.makeText(activity, "An error has
    occured in saving audio", Toast.
    LENGTH_LONG).show();
  }
 });

 //Wait for wav file to be saved
 //noinspection StatementWithEmptyBody
 while (!finished) {
  if (string.length() < MAX_CHARS){
   break;
  }
 }
}

/**
     * Method that creates ProgressDialog
         while there is a background task.
     */
private void createDialog(){
 this.progressDialog = new ProgressDialog(
    this.activity);
 this.progressDialog.setCancelable(false);
 this.progressDialog.
    setCanceledOnTouchOutside(false);
 this.progressDialog.setMessage("Saving long
    texts take longer...");
 this.progressDialog.show();
}

/**
     * Method that splits string into MAX_CHAR
         .
     * @param text the string to be splitted.
     * @return the arraylist of string that is
         splitted into MAX_CHARS.
     */
private ArrayList<String> splitText(final
    String text) {
 final int textLength = text.length();
 final ArrayList<String> splittedText = new
    ArrayList<>();

 int index = 0;
```

```
  while (index < textLength) {
   splittedText.add(text.substring(index, Math
     .min(index + MAX_CHARS, textLength)));
   index += MAX_CHARS;
  }
   return splittedText;
  }
 }
```

Path = Tess2Speech/tess2Speech/src/main/java/anteraaron/tess2speech/Task_SaveCanvas.java

```
package anteraaron.tess2speech;

import android.app.Activity;
import android.graphics.Bitmap;
import android.net.Uri;
import android.os.AsyncTask;
import android.util.Log;
import android.view.View;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;

/**
 * An AsyncTask that Saves the user drawn text
      and convert it to text.
 * @author Anter Aaron M. Custodio
 * @since 2016.3.22
 */
public class Task_SaveCanvas extends AsyncTask
     <Boolean, Boolean, Boolean>{

 private Bitmap bitmap;
 private View content;
 private Activity activity;
 private FileOutputStream out;
 private Uri uri;

 public Task_SaveCanvas(final Activity
     activity){
  this.activity = activity;
 }

 @Override
 protected void onPreExecute() {
  super.onPreExecute();
  //Access drawingView's content and ready for
      saving
  this.content = Fragment_DrawingCanvas.
     drawingView;
  this.content.setDrawingCacheEnabled(true);
  this.content.setDrawingCacheQuality(View.
     DRAWING_CACHE_QUALITY_HIGH);
  this.bitmap = content.getDrawingCache();
 }


 @Override
 protected Boolean doInBackground(final
     Boolean... params) {
  //Set path of the Drawing Canvas cache
  final File file = new File(this.activity.
     getCacheDir() + File.separator + "cache.
     png");
  this.uri = Uri.fromFile(file);

  try{
   //noinspection ResultOfMethodCallIgnored
   file.createNewFile();
   this.out = new FileOutputStream(file);

   //bitmap should not be reused
   if(!this.bitmap.isRecycled() && !
     isCancelled()) {
    this.bitmap.compress(Bitmap.CompressFormat
     .PNG, 100, out);
   }
  }catch (IOException e){
   Log.e("IO Exception", "Stack Trace:", e);
  }

  return params[0];
 }

 @Override
 protected void onPostExecute(final Boolean
     play) {
  super.onPostExecute(play);
  try {
   this.out.flush();
   this.out.close();
   if(!isCancelled()){
    //If thread is not cancelled Convert the
     canvas image to text.
    new ToText(this.activity).execute(this.uri
     , play);
   }

  } catch (IOException e) {
   Log.e("IO Exception", "Stack Trace:", e);
  }
  this.content.setDrawingCacheEnabled(false);
  this.content.destroyDrawingCache();
  this.bitmap.recycle();
 }
}
```

Path = Tess2Speech/tess2Speech/src/main/java/anteraaron/tess2speech/Task_SaveCroppedImage.java

```
package anteraaron.tess2speech;

import android.app.Activity;
import android.app.ProgressDialog;
import android.content.pm.ActivityInfo;
import android.graphics.Bitmap;
import android.os.AsyncTask;
import android.util.Log;

import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;

/**
 * An AsyncTask that saves the cropped image.
 * @author Anter Aaron M. Custodio
 * @since 2016.3.22
 */
public class Task_SaveCroppedImage extends
     AsyncTask<Bitmap, Void, File> {

 private Activity activity;
 private AsyncResponse delegate;
 private ProgressDialog progressDialog;
 //Create an interface to determine that the
     AsyncTask is finished.
 public interface AsyncResponse {
  void processFinished(final File file);
 }
 //Constructor
 public Task_SaveCroppedImage(final Activity
     activity, final AsyncResponse delegate) {
  this.activity = activity;
  this.delegate = delegate;
 }

 @Override
 protected void onPreExecute(){
  activity.setRequestedOrientation(
     ActivityInfo.SCREEN_ORIENTATION_NOSENSOR)
     ;
```

```
        //Start Progress Dialog
        createDialog();
    }

    @Override
    protected File doInBackground(final Bitmap...
        params) {
        final Bitmap image = params[0];
        final String fileName = "crop";
        //If crop[num].png exists, iterate num and
           create crop[num].png until crop[num].png
           is unique
        int num = 0;
        File file = new File(Directories.IMAGES_DIR
           + File.separator + fileName + ".png");
        while(file.exists()) {
            num++;
            file = new File(Directories.IMAGES_DIR +
              File.separator + fileName + "(" + num + "
              ).png");
        }

        try {
            //Save file
            final FileOutputStream out = new
              FileOutputStream(file);
            image.compress(Bitmap.CompressFormat.PNG,
              100, out);
            out.flush();
            out.close();
        } catch (IOException e) {
            Log.e("IO Exception", "Stack Trace:", e);
        }
        return file;
    }

    @Override
    protected void onPostExecute(final File file)
        {
        super.onPostExecute(file);
        //return the cropped image File
        this.delegate.processFinished(file);
        activity.setRequestedOrientation(
           ActivityInfo.
           SCREEN_ORIENTATION_UNSPECIFIED);
        //cancel the progress dialog after saving
           image.
        this.progressDialog.cancel();

    }
    /**
         * Method that creates ProgressDialog
             while there is a background task.
         */
    private void createDialog(){
        this.progressDialog = new ProgressDialog(
           this.activity);
        this.progressDialog.setCancelable(false);
        this.progressDialog.
           setCanceledOnTouchOutside(false);
        this.progressDialog.setMessage("Cropping...")
           ;
        this.progressDialog.show();
    }

}
```

Path = Tess2Speech/tess2Speech/src/main/java/anteraaron/tess2speech/Task_SavePdf.java

```
package anteraaron.tess2speech;

import android.app.Activity;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.ClipData;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.ActivityInfo;
import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.Color;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Build;
import android.provider.MediaStore;
import android.text.Layout;
import android.text.StaticLayout;
import android.text.TextPaint;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.Toast;
import com.googlecode.leptonica.android.Pix;
import com.googlecode.leptonica.android.
    ReadFile;
import com.googlecode.leptonica.android.
    WriteFile;
import com.googlecode.tesseract.android.
    TessPdfRenderer;
import java.io.File;
import java.io.IOException;

import kaloer.filepicker.FilePickerActivity;

/**
 * An AsyncTask that saves the text to a PDF
     or converts an Image to PDF using
     Tesseract's PDF Renderer.
 * Note that this AsyncTask is auto-execute. .
     execute() does not need to be called.
 * @author Anter Aaron M. Custodio
 * @since 3.22.2016
 */
public class Task_SavePdf extends AsyncTask<
    Void, Void, Void>{
    private EditText editText_input;
    private Activity activity;
    private String path;
    private Intent data;
    private Uri uri;
    private ProgressDialog progressDialog;
    private boolean isData;

    /**
         * Constructor if a data/image is passed.
         * @param activity the activity from which
             the image comes from.
         * @param data the image being passed.
         */
    public Task_SavePdf(final Activity activity,
        final Intent data){
        this.activity = activity;
        this.data = data;
        this.isData = true;
        initialize();
    }

    /**
         * Constructor if no image is passed. This
             assumes that the file being
             converted to PDF is from the
             editText view
         * @param activity the activity which
             calls this class
         */
    public Task_SavePdf(final Activity activity){
        this.activity = activity;
        this.isData = false;
        initialize();
    }

    /**
         * Method that initializes the UI and
             Tesseract's PDF Renderer
         */
    private void initialize(){
        this.editText_input = new EditText(activity)
           ;

        //Create alert dialog
        final AlertDialog.Builder dialogBuilder =
           new AlertDialog.Builder(activity);
        dialogBuilder.setTitle("Save as PDF");
        dialogBuilder.setMessage("Input file name: "
           );

        //Add edit text to alert dialog
        final LinearLayout.LayoutParams lp = new
           LinearLayout.LayoutParams(LinearLayout.
           LayoutParams.MATCH_PARENT, LinearLayout.
           LayoutParams.MATCH_PARENT);
        this.editText_input.setLayoutParams(lp);
        dialogBuilder.setView(this.editText_input);
```

```
 //Set action for alert dialog
 dialogBuilder.setPositiveButton("Ok", new
    DialogInterface.OnClickListener() {
  @Override
  public void onClick(DialogInterface dialog,
     int which) {
   save();
  }
 });
 dialogBuilder.setNegativeButton("Cancel",
    new DialogInterface.OnClickListener() {
  @Override
  public void onClick(DialogInterface dialog,
     int which) {
   dialog.dismiss();
   Toast.makeText(activity, "Save cancelled",
     Toast.LENGTH_LONG).show();
  }
 });
 final AlertDialog alertDialog =
    dialogBuilder.create();
 alertDialog.show();
}

@Override
protected Void doInBackground(final Void...
    params) {
 //create PDF in background.
 createPDF();
 return null;
}

@Override
protected void onPostExecute(final Void param
    ) {
 super.onPostExecute(param);
 //display pdf
 Libraries.displayPdf(this.uri);
 activity.setRequestedOrientation(
    ActivityInfo.
    SCREEN_ORIENTATION_UNSPECIFIED);
 //cancel the progress dialog after saving
    audio.
 this.progressDialog.cancel();
 Toast.makeText(this.activity, "PDF saved to:
     " + this.path + ".pdf", Toast.
    LENGTH_LONG).show();
}

/**
     * Method that prepares for saving of pdf.
     * Asks a user for file name and creates
         file for saving, then determines if
         that file already exists.
     */
private void save(){
 //Set file directory.
 final String filename = editText_input.
    getText().toString();
 this.path = Directories.PDF_DIR + File.
    separator + filename;

 //Create File for saving.
 final File file = new File(this.path + ".pdf
    ");
 this.uri = Uri.fromFile(file);

 //If file exists, prompt overwrite dialog
    box.
 if (file.exists()) {
  new AlertDialog.Builder(this.activity)
     .setTitle("File already exist")
     .setMessage("Do you want to overwrite the
      existing file?")
       //Ok button, Overwrite existing file.
     .setPositiveButton("Yes", new
    DialogInterface.OnClickListener() {
      public void onClick(DialogInterface
    dialog, int which) {
       activity.setRequestedOrientation(
    ActivityInfo.SCREEN_ORIENTATION_NOSENSOR)
    ;
       createDialog();
       execute();
      }
     })
     //Cancel Button, dismiss dialog box.
     .setNegativeButton("No", new
    DialogInterface.OnClickListener() {
      public void onClick(DialogInterface
    dialog, int which) {
       dialog.dismiss();
       Toast.makeText(activity, "Save
    cancelled", Toast.LENGTH_LONG).show();
      }
     })
     .show();
 } else {
  createDialog();
  execute();
 }
}

/**
     * Method that converts an image to PDF or
         string to PDF.
     */
private void createPDF(){
 //Create Tesseract instance and create
    writable pdf
 TessPdfRenderer pdfRenderer = new
    TessPdfRenderer(Libraries.tesseract, this
    .path);
 Libraries.tesseract.endDocument(pdfRenderer)
    ;

 //Write to the created pdf
 pdfRenderer = new TessPdfRenderer(Libraries.
    tesseract, this.path);
 Libraries.tesseract.beginDocument(
    pdfRenderer, "Title");
 if(this.isData) {
  //If image is passed and current Build
    version is Jelly Bean above
  if ((Build.VERSION.SDK_INT >= Build.
    VERSION_CODES.JELLY_BEAN_MR2) && (data.
    getData() == null) && (!this.data.
    hasExtra(FilePickerActivity.
    EXTRA_FILE_PATH))) {
   final ClipData clipdata = this.data.
    getClipData();
   for (int i = 0; i < clipdata.getItemCount
    (); i++) {
    //Check if image format is correct
    if (Intent_BrowseFiles.
    checkImageExtension(activity, clipdata.
    getItemAt(i).getUri())) {
     addPage(pdfRenderer, clipdata.getItemAt(
    i).getUri());
    } else {
     Toast.makeText(activity, "An error has
    occured. An image has an invalid format",
     Toast.LENGTH_LONG).show();
    }

   }
  } else {
   //Else pass uri directly
   Uri imageUri;

   if (this.data.hasExtra(FilePickerActivity.
    EXTRA_FILE_PATH)) {
    //Check if the user used the built in
    browser
    imageUri = Uri.fromFile(new File(this.
    data.getStringExtra(FilePickerActivity.
    EXTRA_FILE_PATH)));
   } else {
    //The user does not use the build in
    browser.
    imageUri = this.data.getData();
   }
   //Check if image format is correct
   if (Intent_BrowseFiles.checkImageExtension
    (activity, imageUri)) {
    addPage(pdfRenderer, imageUri);
   } else {
    Toast.makeText(activity, "Select an image
     with an appropriate format.", Toast.
    LENGTH_LONG).show();
   }

  }
 } else {
  //If no image is passed, pass null
  addPage(pdfRenderer, null);
 }
 //Finalize PDF
 Libraries.tesseract.endDocument(pdfRenderer)
    ;
 pdfRenderer.recycle();
}

/**
     * Continuation of the createPDF method.
         This method is the specific method
         that renders an image/text to PDF.
     * @param pdfRenderer TessPdfRenderer
         Object
     * @param uri uri of the image else null.
         If this is null, assume text will be
          converted as PDF.
     */
private void addPage(final TessPdfRenderer
    pdfRenderer, final Uri uri){

 try {
  Pix pix;
```

```
    if (this.isData) {
     //If there is an image passed, get the
      bitmap from uri. Create a pix object from
       that bitmap.
     Bitmap bitmap = MediaStore.Images.Media.
      getBitmap(this.activity.
      getContentResolver(), uri);
     pix = ReadFile.readBitmap(bitmap);
    } else {
     //Create a pix object from the string in
      EditTextView
     pix = getTextImage(Fragment_ConvertedText.
      editText.getText().toString());
    }
    //Create a temporary file which will be the
       format of the pix file
    final File image = File.createTempFile("
     page", ".png");
    WriteFile.writeImpliedFormat(pix, image);
    Libraries.tesseract.addPageToDocument(pix,
     image.getAbsolutePath(), pdfRenderer);
    pix.recycle();
   } catch (IOException e) {
    e.printStackTrace();
   }
  }

  /**
      * A method that creates an image from
          input text.
      * @param text the text to be converted to
           image.
      * @return returns the image version of
          the text.
      */
  private Pix getTextImage(final String text) {
   //Set font size
   final TextPaint textPaint = new TextPaint();
   textPaint.setAntiAlias(true);
   textPaint.setTextSize(24);

   //Draw the text to the bitmap using
      StaticLayout.
   final StaticLayout textLayout = new
      StaticLayout(text, textPaint, 1000,
      Layout.Alignment.ALIGN_NORMAL, 1.0f, 1.0f
      , false);
   final Bitmap bitmap = Bitmap.createBitmap(
      textLayout.getEllipsizedWidth(),
      textLayout.getHeight(), Bitmap.Config.
      ARGB_8888);
   final Canvas canvas = new Canvas(bitmap);
   //Make background of texts white.
   canvas.drawColor(Color.WHITE);
   canvas.save();
   canvas.translate(0,0);
   textLayout.draw(canvas);
   //canvas.restore();
   return ReadFile.readBitmap(bitmap);
  }

  /**
      * Method that creates ProgressDialog
          while there is a background task.
      */
  private void createDialog(){
   this.progressDialog = new ProgressDialog(
      this.activity);
   this.progressDialog.setCancelable(false);
   this.progressDialog.
      setCanceledOnTouchOutside(false);
   this.progressDialog.setMessage("Saving as
      PDF...");
   this.progressDialog.show();
  }
 }
```

Path = Tess2Speech/tess2Speech/src/main/java/anteraaron/tess2speech/Task_SaveRotatedBitmap.java

```
package anteraaron.tess2speech;

import android.app.Activity;
import android.app.ProgressDialog;
import android.content.pm.ActivityInfo;
import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Matrix;
import android.net.Uri;
import android.os.AsyncTask;
import android.util.Log;
import android.widget.Toast;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;

/**
 * An AsyncTask that saves the rotated bitmap.
 * @author Anter Aaron M. Custodio
 * @since 2016.3.22
 */
public class Task_SaveRotatedBitmap extends
    AsyncTask<Object, Void, Void> {
 private ProgressDialog progressDialog;
 private Activity activity;
 private Uri uri;
 private static final int MAX_WIDTH = 300; //
    Max width after rotation
 private static final int MAX_HEIGHT = 300; //
    Max height after rotation
 //private Bitmap bitmap, rotatedBitmap,
    finalBitmap;
 private AsyncResponse delegate;

 //Create an interface to determine that the
    AsyncTask is finished.
 public interface AsyncResponse {
  void processFinished(final Uri uri);
 }

 //Constructor
 public Task_SaveRotatedBitmap(final Activity
    activity, final AsyncResponse delegate){
  this.activity = activity;
  this.delegate = delegate;
 }

 @Override
 protected void onPreExecute(){
  super.onPreExecute();
  activity.setRequestedOrientation(
     ActivityInfo.SCREEN_ORIENTATION_NOSENSOR)
     ;
  //Start progress Dialog
  this.progressDialog = new ProgressDialog(
     this.activity);
  this.progressDialog.setCancelable(false);
  this.progressDialog.
     setCanceledOnTouchOutside(false);
  this.progressDialog.setMessage("Rotating
     image...");
  this.progressDialog.show();
 }

 @Override
 protected Void doInBackground(final Object...
     params) {
  FileOutputStream out = null;

  try {
   //rotate the bitmap
   final Bitmap bitmap = BitmapLoader.
     decodeSampledBitmap(new File((String)
     params[0]), MAX_WIDTH, MAX_HEIGHT);
   final Matrix matrix = new Matrix();
   matrix.postRotate((int) params[1]);
   final Bitmap rotatedBitmap = Bitmap.
     createBitmap(bitmap, 0, 0, bitmap.
     getWidth(), bitmap.getHeight(), matrix,
     true);
   bitmap.recycle();

   //make bitmap background to white
   final Bitmap finalBitmap = Bitmap.
     createBitmap(rotatedBitmap.getWidth(),
     rotatedBitmap.getHeight(), rotatedBitmap.
     getConfig());
   final Canvas canvas = new Canvas(
     finalBitmap);
   canvas.drawColor(Color.WHITE);
   canvas.drawBitmap(rotatedBitmap, 0, 0, null
     );
   rotatedBitmap.recycle();
```

```
    //Create file for saving (params[3] is the
      original file name)
    final String fileName = params[2] + ".png";
    final File file = new File(Directories.
      IMAGES_DIR + File.separator + "r" +
      params[1] + "_" + fileName);
    out = new FileOutputStream(file);

    //Save bitmap with lossless compression
    finalBitmap.compress(Bitmap.CompressFormat.
      PNG, 100, out);
    finalBitmap.recycle();
    this.uri = Uri.fromFile(file);
   } catch (IOException e) {
    Log.e("IO Exception", "Stack Trace:", e);
   } finally {
    try {
     if(out != null) {
      out.flush();
      out.close();
     }
    } catch (IOException e) {
     Log.e("IO Exception", "Stack Trace:", e);
    }
   }
   return null;
  }

  @Override
  protected void onPostExecute(final Void
      results) {
   Libraries.galleryAddPic(this.uri);
   //Return rotated bitmap.
   this.delegate.processFinished(this.uri);
   activity.setRequestedOrientation(
      ActivityInfo.
      SCREEN_ORIENTATION_UNSPECIFIED);
   //cancel the progress dialog after saving
      audio.
   this.progressDialog.cancel();
   Toast.makeText(this.activity, "Saved to: " +
       Directories.IMAGES_DIR, Toast.
      LENGTH_LONG).show();
  }
 }
```

Path = Tess2Speech/tess2Speech/src/main/java/anteraaron/tess2speech/Task_SaveText.java

```
package anteraaron.tess2speech;

import android.app.Activity;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.content.pm.ActivityInfo;
import android.os.AsyncTask;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.Toast;
import org.apache.commons.io.FileUtils;
import java.io.File;
import java.io.IOException;

/**
 * An AsyncTask that saves text in
     LinedEditText to .txt file
 * @author Anter Aaron M. Custodio
 * @since 2016.3.22
 */
public class Task_SaveText extends AsyncTask<
    Void, Void, Void>{

 private EditText editText_input;
 private ProgressDialog progressDialog;
 private String string, path;
 private File file;
 private Activity activity;

 //Constructor
 public Task_SaveText(final Activity activity)
     {
  this.activity = activity;
  //create and editText that will be embedded
     in the alertDialog.
  this.editText_input = new EditText(activity)
     ;
  //Get the input from editText. This will be
     the fileName of the .txt file
  this.string = Fragment_ConvertedText.
     editText.getText().toString();
  //create alert dialog
  final AlertDialog.Builder dialogBuilder =
     new AlertDialog.Builder(activity);
  dialogBuilder.setTitle("Save Converted Text
     to .txt");
  dialogBuilder.setMessage("Input file name: "
     );
  //Add edit text to alert dialog
  final LinearLayout.LayoutParams lp = new
     LinearLayout.LayoutParams(LinearLayout.
     LayoutParams.MATCH_PARENT, LinearLayout.
     LayoutParams.MATCH_PARENT);
  this.editText_input.setLayoutParams(lp);
  dialogBuilder.setView(this.editText_input);

  //Set action for alert dialog
  dialogBuilder.setPositiveButton("Ok", new
     DialogInterface.OnClickListener() {

   @Override
   public void onClick(DialogInterface dialog,
      int which) {
    save();
   }
  });
  //
  dialogBuilder.setNegativeButton("Cancel",
     new DialogInterface.OnClickListener() {
   @Override
   public void onClick(DialogInterface dialog,
      int which) {
    dialog.dismiss();
    Toast.makeText(activity, "Save cancelled",
      Toast.LENGTH_LONG).show();
   }
  });
  //Display the created dialogBuilder
  final AlertDialog alertDialog =
     dialogBuilder.create();
  alertDialog.show();
 }

 @Override
 protected Void doInBackground(final Void...
     params) {
  try {
   writeFile(this.file, this.string);
  } catch (IOException e) {
   e.printStackTrace();
  }
  return null;
 }

 @Override
 protected void onPostExecute(final Void param
     ) {
  super.onPostExecute(param);
  activity.setRequestedOrientation(
     ActivityInfo.
     SCREEN_ORIENTATION_UNSPECIFIED);
  //cancel the progress dialog after saving .
     txt file.
  this.progressDialog.cancel();
  Toast.makeText(activity, "File saved to: " +
      this.path, Toast.LENGTH_LONG).show();
 }

 /**
     * Method that saves the text from
         LinedEditText to .txt.
     */
 private void save(){

  try {
   //Set file directory
   final String filename = this.editText_input
     .getText() + ".txt";
   this.path = Directories.TEXT_DIR + File.
     separator + filename;
   this.file = new File(this.path);
   //If file exists, prompt overwrite dialog
     box
```

```
    if (this.file.exists()) {
     new AlertDialog.Builder(this.activity)
        .setTitle("File already exist")
        .setMessage("Do you want to overwrite
     the existing file?")
        //Ok button, Overwrite existing file
        .setPositiveButton("Yes", new
     DialogInterface.OnClickListener() {
          public void onClick(DialogInterface
     dialog, int which) {
            activity.setRequestedOrientation(
     ActivityInfo.SCREEN_ORIENTATION_NOSENSOR)
     ;
            createDialog();
            execute();
          }
        })
        //Cancel Button, dismiss dialog box
        .setNegativeButton("No", new
     DialogInterface.OnClickListener() {
          public void onClick(DialogInterface
     dialog, int which) {
            dialog.dismiss();
            Toast.makeText(activity, "Save
     cancelled", Toast.LENGTH_LONG).show();
          }
        })
        .show();
    } else {
     createDialog();
     execute();
    }
   } catch (Exception e) {
    //An error has occured while saving
    Toast.makeText(this.activity, "Error saving
         file!", Toast.LENGTH_SHORT).show();
    }
   }

   /**
       * Continuation of Task_SaveText.save().
           Write the string to a text file.
       * @param file file in which the text will
           be written.
       * @param string text that will be written
           to the file.
       * @throws IOException
       */
   private void writeFile(final File file,
       final String string) throws IOException {
    //noinspection ResultOfMethodCallIgnored
    FileUtils.writeStringToFile(file, string);
   }

   /**
       * Method that creates ProgressDialog
           while there is a background task.
       */
   private void createDialog(){
    this.progressDialog = new ProgressDialog(
       this.activity);
    this.progressDialog.setCancelable(false);
    this.progressDialog.
       setCanceledOnTouchOutside(false);
    this.progressDialog.setMessage("Saving...");
    this.progressDialog.show();
   }
  }
```

Path = Tess2Speech/tess2Speech/src/main/java/anteraaron/tess2speech/Task_UnzipEpub.java

```
package anteraaron.tess2speech;

import android.app.Activity;
import android.app.ProgressDialog;
import android.content.pm.ActivityInfo;
import android.net.Uri;
import android.os.AsyncTask;
import android.widget.Toast;

import java.io.BufferedInputStream;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.util.zip.ZipEntry;
import java.util.zip.ZipInputStream;

/**
 * An AsyncTask that extracts epub contents to
       folder EPUB_CACHE_DIR to access css
      files and images
 * @author Anter Aaron M. Custodio
 * @since 2016.3.27
 */
public class Task_UnzipEpub extends AsyncTask<
     Object,Void,Boolean> {
 private ProgressDialog progressDialog;
 private AsyncResponse delegate;
 private Activity activity;
 private String cachePath;
 private Uri epubUri;
 //Create an interface to determine that the
     AsyncTask is finished.
 public interface AsyncResponse {
  void processFinished(final Uri epubUri,
     final String path);
 }
 //Constructor
 public Task_UnzipEpub(final Activity activity
     , final AsyncResponse delegate) {
  this.activity = activity;
  this.delegate = delegate;
 }

 @Override
 protected void onPreExecute(){
  activity.setRequestedOrientation(
     ActivityInfo.SCREEN_ORIENTATION_NOSENSOR)
     ;
  //Start Progress Dialog
  createDialog();
 }

 @Override
 protected Boolean doInBackground(final Object
     ... params) {
  //Accepts the uri of epub, and the
     extraction path (cachePath).
  this.epubUri = (Uri)params[0];
  this.cachePath = (String)params[1];
  return unZip(this.epubUri, this.cachePath);
 }

 @Override
 protected void onPostExecute(final Boolean
     success) {
  super.onPostExecute(success);
  if (!success) {
   //An error has occured in extracting epub.
   Toast.makeText(this.activity, "Error
     extracting epub", Toast.LENGTH_LONG).show
     ();
  }
  //return extracted epub cache path and the
     original epub uri.
  this.delegate.processFinished(this.epubUri,
     this.cachePath);
  activity.setRequestedOrientation(
     ActivityInfo.
     SCREEN_ORIENTATION_UNSPECIFIED);
  //cancel the progress dialog after creating
     cache.
  this.progressDialog.cancel();
  this.progressDialog.dismiss();
 }

 /**
     * Method that unzips the epub to a
         selected path.
     * @param epubUri the uri of the epub to
         be extracted.
     * @param cachePath the path where the
         epub contents will be extracted.
     * @return true if extracting is
         successful.
     */
  public boolean unZip(final Uri epubUri,
     final String cachePath) {
   InputStream is;
   ZipInputStream zis;

   try {
```

```
    String filename;
    is = this.activity.getContentResolver().
     openInputStream(epubUri);
    assert is != null;
    zis = new ZipInputStream(new
     BufferedInputStream(is));
    ZipEntry ze;

    byte[] buffer = new byte[1024];

    int count;
    while ((ze = zis.getNextEntry()) != null)
    {
     filename = ze.getName();

     //Create directory if it does not exist.
     if (ze.isDirectory()) {
      final File dir = new File(cachePath,
     filename);
      //noinspection ResultOfMethodCallIgnored
      dir.mkdirs();
      continue;
     } else {
      //Create parent directory if it is a
     file.
      final File dir = new File(cachePath,
     filename);
      final String parent = dir.getParentFile
     ().getPath();
      final File parentDir = new File(parent);
      //noinspection ResultOfMethodCallIgnored
      parentDir.mkdirs();
     }

     //Copy the current file to the selected
     path
     final FileOutputStream fout = new
      FileOutputStream(cachePath + File.
      separator + filename);
      while ((count = zis.read(buffer)) != -1)
      {
      fout.write(buffer, 0, count);
      }

      fout.close();
      zis.closeEntry();
     }

     is.close();
     zis.close();
    } catch (IOException e) {
     e.printStackTrace();
     return false;
    }
    return true;
    }

 /**
     * Method that creates ProgressDialog
         while there is a background task.
     */
  private void createDialog(){
   this.progressDialog = new ProgressDialog(
      this.activity);
   this.progressDialog.setCancelable(false);
   this.progressDialog.
      setCanceledOnTouchOutside(false);
   this.progressDialog.setMessage("Preparing
      Epub...");
   this.progressDialog.show();
  }
 }
```

Path = Tess2Speech/tess2Speech/src/main/java/anteraaron/tess2speech/Task_WebViewToText.java

```
package anteraaron.tess2speech;

import android.app.Activity;
import android.app.ProgressDialog;
import android.content.pm.ActivityInfo;
import android.os.AsyncTask;
import android.text.Html;
import android.text.Spanned;
import android.util.Log;

/**
 * An AsyncTask that converts Epub displayed
      on WebView to Text.
 * @author Anter Aaron M. Custodio
 * @since 2016.3.30
 */
public class Task_WebViewToText extends
     AsyncTask<Boolean,Void,Boolean>{
 private Activity activity;
 private ProgressDialog progressDialog;
 private static Spanned convertedText;
 private Boolean finished;

 public Task_WebViewToText(final Activity
     activity) {
  this.activity = activity;
  this.finished = false;
 }

 @Override
 protected void onPreExecute(){
  super.onPreExecute();
  //Start progress Dialog
  activity.setRequestedOrientation(
     ActivityInfo.SCREEN_ORIENTATION_NOSENSOR)
     ;
  createDialog();
 }

 @Override
 protected Boolean doInBackground(final
     Boolean... params) {
  //get displayed the html formatted text
     displayed in webView.
  convertedText = Html.fromHtml(
     Fragment_EPubViewer_TOC.htmlContent);
  //only set text to textView if converted
     text fragment is visible (params[0](play)
      == false) to save processing time.
  if (!params[0]) {
   publishProgress();
   while(!finished){
    //Wait until textView finished setting
     text to display progress dialog while
     textView is setting text.
    Log.d("Wait", "Waiting for textview to set
      text.");
   }
  }
  return params[0];
 }

 @Override
 protected void onProgressUpdate(final Void...
      params){
  // Check stuff on the UI
  super.onProgressUpdate(params);
  //Change textView's value in ConvertedText
     Fragment
  Fragment_ConvertedText.editText.append(
     convertedText);
  Fragment_ConvertedText.editText.setSelection
     (Fragment_ConvertedText.editText.length()
     );
  finished = true;
 }

 @Override
 protected void onPostExecute (final Boolean
     play) {
  super.onPostExecute(play);
  activity.setRequestedOrientation(
     ActivityInfo.
     SCREEN_ORIENTATION_UNSPECIFIED);
  //End progress dialog
  this.progressDialog.cancel();

  //If play parameter is true, continue to
     ToSpeech() AsyncTask
  if(play) {
   new ToSpeech().execute(convertedText.
     toString());
  }
 }

 /**
     * Method that creates ProgressDialog
         while there is a background task.
```

```
     */
    private void createDialog(){
     this.progressDialog = new ProgressDialog(
        this.activity);
     this.progressDialog.setCancelable(false);
     this.progressDialog.
        setCanceledOnTouchOutside(false);
     this.progressDialog.setMessage("Converting
        to text...");
     this.progressDialog.show();
    }
```

Path = Tess2Speech/tess2Speech/src/main/java/anteraaron/tess2speech/ToSpeech.java

```
package anteraaron.tess2speech;

import android.annotation.TargetApi;
import android.os.AsyncTask;
import android.os.Build;
import android.speech.tts.TextToSpeech;

import java.util.ArrayList;
import java.util.HashMap;

/**
 * An AsyncTask that converts texts to speech.
 * @author Anter Aaron M. Custodio
 * @since 2016.3.22
 */
public class ToSpeech extends AsyncTask<String
    , Void, Void> {
 private static final int MAX_CHARS = 3900; //
    Max characters that tts can read.
 private static String text;

 @Override
 protected void onPreExecute() {
  super.onPreExecute();
  //set button label to stop
  Activity_Main.button_play.
     setCompoundDrawablesWithIntrinsicBounds
     (0, R.mipmap.ic_action_playback_stop, 0,
     0);
  Activity_Main.button_play.setText(R.string.
     button_stop);

  //Disable buttons to avoid button mashing
  Activity_Main.button_text.setEnabled(false);
  Activity_Main.button_text.setClickable(false
     );
 }

 @Override
 protected Void doInBackground(final String...
     params) {
  text = params[0];
  return null;
 }

 @Override
 protected void onPostExecute (final Void
     result) {
  super.onPostExecute(result);
  speak(text);
 }

 /**
     * Read text using text to speech for
         below Lollipop.
     * @param text text to be converted to
         speech.
     * @deprecated for API lower than lollipop
         . ttsGreater21 is alternative
     */
 @SuppressWarnings("deprecation")
 private void ttsUnder20(final String text) {
  final HashMap<String, String> map = new
     HashMap<>();
  map.put(TextToSpeech.Engine.
     KEY_PARAM_UTTERANCE_ID, "Message");

  //If the number of chars exceeds the tts
     limit
  if (text.length() >= MAX_CHARS) {
   final ArrayList<String> splittedText =
     splitText(text);
   //Add the splitted texts to tts queue.
   for (final String queueElement :
     splittedText) {
    Libraries.tts.speak(queueElement,
     TextToSpeech.QUEUE_ADD, map);
   }
  } else {
   //If not, play directly
   Libraries.tts.speak(text, TextToSpeech.
     QUEUE_FLUSH, map);
  }
 }
 /**
     * Read text using text to speech for
         Lollipop and above.
     * @param text text to be converted to
         speech.
     */
 @TargetApi(Build.VERSION_CODES.LOLLIPOP)
 private void ttsGreater21(final String text)
     {
  final String uId = this.hashCode() + "";

  //If the number of chars exceeds the tts
     limit
  if (text.length() >= MAX_CHARS) {
   final ArrayList<String> splittedText =
     splitText(text);
   //Add the splitted texts to tts queue.
   for (final String queueElement :
     splittedText) {
    Libraries.tts.speak(queueElement,
     TextToSpeech.QUEUE_ADD, null, uId);
   }
  } else {
   //If not, play directly.
   Libraries.tts.speak(text, TextToSpeech.
     QUEUE_FLUSH, null, uId);
  }
 }

 /**
     * Read text using text to speech.
     * @param text text to be converted to
         speech.
     */
 private void speak(final String text){
  if(Build.VERSION.SDK_INT >= Build.
     VERSION_CODES.LOLLIPOP){
   //for depricated OS (lollipop and higher)
   ttsGreater21(text);
  }else{
   //for non depricated OS (lower than
     lollipop)
   //noinspection deprecation
   ttsUnder20(text);
  }
 }

 /**
     * Method that splits string into MAX_CHAR
         .
     * @param text the string to be splitted.
     * @return the arraylist of string that is
         splitted into MAX_CHARS.
     */
 private ArrayList<String> splitText(final
     String text) {
  final int textLength = text.length();
  final ArrayList<String> splittedText = new
     ArrayList<>();

  int index = 0;
  while (index < textLength) {
   splittedText.add(text.substring(index, Math
     .min(index + MAX_CHARS, textLength)));
   index += MAX_CHARS;
  }

  return splittedText;
 }

}
```

Path = Tess2Speech/tess2Speech/src/main/java/anteraaron/tess2speech/ToText.java

```java
package anteraaron.tess2speech;

import android.app.Activity;
import android.app.ProgressDialog;
import android.content.SharedPreferences;
import android.content.pm.ActivityInfo;
import android.graphics.Bitmap;
import android.net.Uri;
import android.os.AsyncTask;
import android.preference.PreferenceManager;
import android.provider.MediaStore;
import android.util.Log;

import com.googlecode.leptonica.android.
    WriteFile;

import java.io.File;
import java.io.IOException;

/**
 * An AsyncTask that converts the image to
     text using Google's Tesseract (See...
     https://github.com/rmtheis/tess-two)
 * @author Anter Aaron M. Custodio
 * @since 2016.3.22
 */
public class ToText extends AsyncTask<Object,
    Void, Boolean>{
 //Maximum width of image after accelerating
    image process
 private static final int MAX_WIDTH = 300;
 //Maximum height of image after accelerating
    image process
 private static final int MAX_HEIGHT = 300;

 private Activity activity;
 private ProgressDialog progressDialog;
 private static String convertedText;
 //Constructor
 public ToText (final Activity activity) {
  this.activity = activity;
  convertedText = "";
 }

 @Override
 protected void onPreExecute(){
  super.onPreExecute();
  //Disable buttons to avoid button mashing
  Activity_Main.button_text.setEnabled(false);
  Activity_Main.button_text.setClickable(false
     );
  Activity_Main.button_play.setEnabled(false);
  Activity_Main.button_play.setClickable(false
     );
  activity.setRequestedOrientation(
     ActivityInfo.SCREEN_ORIENTATION_NOSENSOR)
     ;
  //Start progress Dialog
  createDialog();
 }

 @Override
 protected Boolean doInBackground(final Object
     ... params) {
  try {
   //Check if image pre process is enabled
   PreferenceManager.setDefaultValues(this.
     activity, R.xml.preferences, false);
   final SharedPreferences settings =
     PreferenceManager.
     getDefaultSharedPreferences(this.activity
     );
   //Create a bitmap of desired input image (
     original or reduced based on preference)
   Bitmap bitmap = loadBitmapFromUri((Uri)
     params[0], settings.getBoolean("
     pref_key_accelerate", true));
   //perform image pre processing
   if (settings.getBoolean("
     pref_key_image_preprocess", true)) {
    bitmap = BitmapLoader.imagePreProcess(
     bitmap);
   }
   //Check if bitmap is empty
   if(bitmap != null) {
    //Feed the image to tesseract
    Libraries.tesseract.setImage(bitmap);
    //Output
    convertedText = Libraries.tesseract.
     getUTF8Text().replaceAll("((?<!\\.)\n)",
     " ");
   }
  } catch (IOException e) {
    Log.e("IO Exception", "Stack Trace:", e);
  }

  return (boolean)params[1];
 }

 @Override
 protected void onPostExecute(final Boolean
     play) {
  super.onPostExecute(play);
  //Remove comment out to view thresholded
     image on phone
    //Libraries.imageView.setImageBitmap(
     WriteFile.writeBitmap(Libraries.tesseract
     .getThresholdedImage()));

  Libraries.tesseract.clear();
  Libraries.tesseract.setVariable("
     classify_enable_learning", "0");
  Libraries.tesseract.setVariable("
     classify_enable_adaptive_matcher", "0");

  //Change textView's value in ConvertedText
     Fragment
  Fragment_ConvertedText.editText.append(
     convertedText);
  Fragment_ConvertedText.editText.setSelection
     (Fragment_ConvertedText.editText.length()
     );

  activity.setRequestedOrientation(
     ActivityInfo.
     SCREEN_ORIENTATION_UNSPECIFIED);
  //End progress dialog
  this.progressDialog.cancel();

  //Re-enable buttons
  Activity_Main.button_text.setEnabled(true);
  Activity_Main.button_text.setClickable(true)
     ;
  Activity_Main.button_play.setEnabled(true);
  Activity_Main.button_play.setClickable(true)
     ;

  //If play parameter is true, continue to
     ToSpeech() AsyncTask
  if(play) {
   new ToSpeech().execute(
     Fragment_ConvertedText.editText.getText()
     .toString());
  }
 }

 /**
     *  Loads a bitmap from an uri.
     * @param uri the Uri of the bitmap to be
         loaded.
     * @param reduceImageSize returns a
         reduced size image if this parameter
         is set to true.
     * @throws IOException
     */
 private Bitmap loadBitmapFromUri(final Uri
     uri, final boolean reduceImageSize)
     throws IOException {
  //Accelerate image pre-process by reducing
     image size
  Bitmap bitmap;
  if (reduceImageSize) {
   //If reduceImageSize is true

   //Get file extension of the file from uri
   final String extension = Intent_BrowseFiles
     .getMimeType(activity, uri);
   //Resample bitmap to reduce size &
     dimension

   bitmap = BitmapLoader.decodeSampledBitmap(
     new File(this.activity.getCacheDir() +
     File.separator + "cache." + extension),
     MAX_WIDTH, MAX_HEIGHT);
  } else {
   //Load the original bitmap without reducing
      quality. (Consumes memory)
   bitmap = MediaStore.Images.Media.getBitmap(
     this.activity.getContentResolver(), uri);
  }

  return bitmap;
 }

 private void createDialog() {
  this.progressDialog = new ProgressDialog(
     this.activity);
  this.progressDialog.setCancelable(false);
  this.progressDialog.
     setCanceledOnTouchOutside(false);
  this.progressDialog.setMessage("Converting
     to text...");
  this.progressDialog.show();
 }
}
```

Path = Tess2Speech/tess2Speech/src/main/res/layout/activity_crop.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.
    android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/
    tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:paddingBottom="@dimen/
    activity_vertical_margin"
  android:paddingLeft="@dimen/
    activity_horizontal_margin"
  android:paddingRight="@dimen/
    activity_horizontal_margin"
  android:paddingTop="@dimen/
    activity_vertical_margin"
  tools:context="anteraaron.tess2speech.
    Activity_Crop"
  android:backgroundTintMode="multiply"
  android:background="@color/colorBackground">


  <com.theartofdev.edmodo.cropper.
    CropImageView
    xmlns:custom="http://schemas.android.com/
    apk/res-auto"
    android:id="@+id/CropImageView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    custom:cropScaleType="fitCenter"
    android:layout_alignParentRight="true"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true"
    android:layout_above="@+id/
    button_crop_cancel"
    android:background="#292929" />

  <Button
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/button_crop_confirm"
    android:drawableBottom="@mipmap/
    ic_action_tick"
    android:background="?android:attr/
    selectableItemBackground"
    android:layout_alignParentBottom="true"
    android:layout_alignParentLeft="true"
    android:layout_marginLeft="56dp"
    android:layout_below="@+id/CropImageView"
    />

  <Button
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/button_crop_cancel"
    android:drawableLeft="@mipmap/
    ic_action_cancel"
    android:background="?android:attr/
    selectableItemBackground"
    android:layout_marginRight="60dp"
    android:layout_marginEnd="60dp"
    android:layout_alignParentBottom="true"
    android:layout_alignEnd="@+id/
    CropImageView" />
</RelativeLayout>
```

Path = Tess2Speech/tess2Speech/src/main/res/layout/activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.
    CoordinatorLayout xmlns:android="http://
    schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/
    res-auto"
  xmlns:tools="http://schemas.android.com/
    tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:fitsSystemWindows="true"
  tools:context=".Activity_Main">

  <android.support.design.widget.AppBarLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:theme="@style/AppTheme.
    AppBarOverlay"
    android:paddingLeft="-10dp">

    <android.support.v7.widget.Toolbar
      android:id="@+id/toolbar"
      android:layout_width="match_parent"
      android:layout_height="?attr/
    actionBarSize"
      android:background="@color/colorPrimary"
      app:popupTheme="@style/AppTheme.
    PopupOverlay" />

  </android.support.design.widget.AppBarLayout
    >

  <include layout="@layout/content_main" />

</android.support.design.widget.
    CoordinatorLayout>
```

Path = Tess2Speech/tess2Speech/src/main/res/layout/activity_rotate.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.
    android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/
    tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:paddingBottom="@dimen/
    activity_vertical_margin"
  android:paddingLeft="@dimen/
    activity_horizontal_margin"
  android:paddingRight="@dimen/
    activity_horizontal_margin"
  android:paddingTop="@dimen/
    activity_vertical_margin"
  tools:context=".Activity_Rotate"
  android:background="@color/colorBackground">

  <ImageView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/imageView_rotate"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:adjustViewBounds="true"
    android:layout_above="@+id/textView_rotate
    "
    android:background="#292929"
    android:contentDescription="@string/
    imageView_rotate_content_desc" />


  <SeekBar
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/seekBar_rotate"
    android:indeterminate="false"
    android:layout_above="@+id/
    button_rotate_save" />

  <TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
```

```
    android:textAppearance="?android:attr/
     textAppearanceSmall"
    android:id="@+id/textView_rotate"
    android:layout_above="@+id/seekBar_rotate"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />

  <Button
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/button_rotate_reset"
    android:drawableLeft="@mipmap/
     ic_action_undo"
    android:background="?android:attr/
     selectableItemBackground"
    android:layout_alignParentBottom="true"
    android:layout_toLeftOf="@+id/
     button_rotate_exit"
    android:layout_toStartOf="@+id/
     button_rotate_exit" />


  <Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/button_rotate_save"
    android:drawableBottom="@mipmap/
     ic_action_tick"
    android:background="?android:attr/
     selectableItemBackground"
    android:layout_alignParentBottom="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="false"
    android:layout_alignTop="@+id/
     button_rotate_reset" />

  <Button
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/button_rotate_exit"
    android:drawableLeft="@mipmap/
     ic_action_cancel"
    android:background="?android:attr/
     selectableItemBackground"
    android:layout_alignParentBottom="true"
    android:layout_alignRight="@+id/
     seekBar_rotate"
    android:layout_alignEnd="@+id/
     seekBar_rotate"
    android:layout_alignTop="@+id/
     button_rotate_reset" />

</RelativeLayout>
```

Path = Tess2Speech/tess2Speech/src/main/res/layout/alert_dialog_change_language_listview.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.
     android.com/apk/res/android"
   android:orientation="vertical"
     android:layout_width="match_parent"
   android:layout_height="match_parent">

  <ListView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/listView_change_language"
    android:choiceMode="multipleChoice" />
</LinearLayout>
```

Path = Tess2Speech/tess2Speech/src/main/res/layout/alert_dialog_change_language_listview_row.xml

```
<CheckedTextView xmlns:android="http://schemas
     .android.com/apk/res/android"
   android:id="@+id/checkedTextView"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:gravity="center"
   android:drawableLeft="?android:attr/
     listChoiceIndicatorMultiple"
   android:checked="false" />
```

Path = Tess2Speech/tess2Speech/src/main/res/layout/alert_dialog_help.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.
     android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:fillViewport="false">

  <LinearLayout xmlns:android="http://schemas.
     android.com/apk/res/android"
    android:orientation="vertical"
     android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:paddingStart="15dp"
    android:paddingEnd="15dp">

    <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:textAppearance="?android:attr/
     textAppearanceLarge"
      android:text="@string/
     textView_help_title"
      android:textStyle="bold"
      android:id="@+id/textView_help_title"
      android:textColor="#000000"
      android:layout_marginBottom="15dp" />

    <!--Tips-->
    <TextView
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:textAppearance="?android:attr/
     textAppearanceLarge"
      android:gravity="center_vertical"
      android:text="@string/textView_help_tips
     "
      android:textStyle="bold"
      android:id="@+id/textView_help_tips"
      android:textColor="#000000"
      android:background="@color/colorAccent"
      android:textSize="20sp" />

    <TextView
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:text="@string/tips_description"
```

```
    android:id="@+id/textView_help_tips_desc
"
    android:textColor="#000000"
    android:autoLink="none"
    android:linksClickable="true"
    android:layout_marginBottom="15dp" />

<!--Camera-->
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/
textAppearanceLarge"
    android:gravity="center_vertical"
    android:text="@string/
textView_help_camera"
    android:drawableLeft="@mipmap/
ic_action_camera"
    android:textStyle="bold"
    android:id="@+id/textView_help_camera"
    android:textColor="#000000"
    android:background="@color/colorAccent"
    android:textSize="20sp" />

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/camera_description
"
    android:id="@+id/
textView_help_camera_desc"
    android:textColor="#000000"
    android:autoLink="all"
    android:linksClickable="true"
    android:layout_marginBottom="15dp" />

<!--Browse Image-->
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/
textAppearanceLarge"
    android:text="@string/
textView_help_browse_image"
    android:drawableLeft="@mipmap/
ic_action_picture"
    android:gravity="center_vertical"
    android:textStyle="bold"
    android:id="@+id/
textView_help_browse_image"
    android:textColor="#000000"
    android:background="@color/colorAccent"
    android:textSize="20sp" />

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/
browse_image_description"
    android:id="@+id/
textView_help_browse_image_description"
    android:textColor="#000000"
    android:autoLink="all"
    android:linksClickable="true"
    android:layout_marginBottom="15dp" />

<!--Play-->
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/
textAppearanceLarge"
    android:text="@string/textView_help_play
"
    android:drawableLeft="@mipmap/
ic_action_playback_play"
    android:gravity="center_vertical"
    android:textStyle="bold"
    android:id="@+id/textView_help_play"
    android:textColor="#000000"
    android:background="@color/colorAccent"
    android:textSize="20sp" />

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/play_description"
    android:id="@+id/
textView_help_play_description"
    android:textColor="#000000"
    android:autoLink="all"
    android:linksClickable="true"
    android:layout_marginBottom="15dp" />

<!--Play-->
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/
textAppearanceLarge"
    android:text="@string/textView_help_stop
"
    android:drawableLeft="@mipmap/
ic_action_playback_stop"
    android:gravity="center_vertical"
    android:textStyle="bold"
    android:id="@+id/textView_help_stop"
    android:textColor="#000000"
    android:background="@color/colorAccent"
    android:textSize="20sp" />

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/stop_description"
    android:id="@+id/
textView_help_stop_description"
    android:textColor="#000000"
    android:autoLink="all"
    android:linksClickable="true"
    android:layout_marginBottom="15dp" />

<!--Clear-->
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/
textAppearanceLarge"
    android:text="@string/
textView_help_clear"
    android:drawableLeft="@mipmap/
ic_action_present"
    android:gravity="center_vertical"
    android:textStyle="bold"
    android:id="@+id/textView_help_clear"
    android:textColor="#000000"
    android:background="@color/colorAccent"
    android:textSize="20sp" />

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/clear_description"
    android:id="@+id/
textView_help_clear_description"
    android:textColor="#000000"
    android:autoLink="all"
    android:linksClickable="true"
    android:layout_marginBottom="15dp" />

<!--Erase-->
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/
textAppearanceLarge"
    android:text="@string/
textView_help_erase"
    android:drawableLeft="@mipmap/ic_eraser"
    android:gravity="center_vertical"
    android:textStyle="bold"
    android:id="@+id/textView_help_erase"
    android:textColor="#000000"
    android:background="@color/colorAccent"
    android:textSize="20sp" />

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/erase_description"
    android:id="@+id/
textView_help_erase_description"
    android:textColor="#000000"
    android:autoLink="all"
    android:linksClickable="true"
    android:layout_marginBottom="15dp" />

<!--Stroke-->
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/
textAppearanceLarge"
    android:text="@string/
textView_help_stroke"
    android:drawableLeft="@mipmap/
ic_action_brush"
    android:gravity="center_vertical"
    android:textStyle="bold"
    android:id="@+id/textView_help_stroke"
    android:textColor="#000000"
    android:background="@color/colorAccent"
    android:textSize="20sp" />

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/stroke_description
```

```
"
    android:id="@+id/
textView_help_stroke_description"
    android:textColor="#000000"
    android:autoLink="all"
    android:linksClickable="true"
    android:layout_marginBottom="15dp" />

<!--Convert to Text-->
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/
textAppearanceLarge"
    android:text="@string/
textView_help_convert_text"
    android:drawableLeft="@mipmap/
ic_action_eye_open"
    android:gravity="center_vertical"
    android:textStyle="bold"
    android:id="@+id/
textView_help_convert_text"
    android:textColor="#000000"
    android:background="@color/colorAccent"
    android:textSize="20sp" />

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/
convert_text_description"
    android:id="@+id/
textView_help_convert_text_description"
    android:textColor="#000000"
    android:autoLink="all"
    android:linksClickable="true"
    android:layout_marginBottom="15dp" />

<!--Back to Input-->
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/
textAppearanceLarge"
    android:text="@string/textView_help_back
"
    android:drawableLeft="@mipmap/
ic_action_eye_closed"
    android:gravity="center_vertical"
    android:textStyle="bold"
    android:id="@+id/textView_help_back"
    android:textColor="#000000"
    android:background="@color/colorAccent"
    android:textSize="20sp" />

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/back_description"
    android:id="@+id/
textView_help_back_description"
    android:textColor="#000000"
    android:autoLink="all"
    android:linksClickable="true"
    android:layout_marginBottom="15dp" />

<!--Rotate-->
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/
textAppearanceLarge"
    android:text="@string/
textView_help_rotate"
    android:drawableLeft="@mipmap/
ic_action_turn_left"
    android:gravity="center_vertical"
    android:textStyle="bold"
    android:id="@+id/textView_help_rotate"
    android:textColor="#000000"
    android:background="@color/colorAccent"
    android:textSize="20sp" />

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/rotate_description
"
    android:id="@+id/
textView_help_rotate_description"
    android:textColor="#000000"
    android:autoLink="all"
    android:linksClickable="true"
    android:layout_marginBottom="15dp" />

<!--Crop-->
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/
textAppearanceLarge"
    android:text="@string/textView_help_crop
"
    android:drawableLeft="@mipmap/
ic_action_crop"
    android:gravity="center_vertical"
    android:textStyle="bold"
    android:id="@+id/textView_help_crop"
    android:textColor="#000000"
    android:background="@color/colorAccent"
    android:textSize="20sp" />

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/crop_description"
    android:id="@+id/
textView_help_crop_description"
    android:textColor="#000000"
    android:autoLink="all"
    android:linksClickable="true"
    android:layout_marginBottom="15dp" />

<!--Append text-->
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/
textAppearanceLarge"
    android:text="@string/
textView_help_append"
    android:gravity="center_vertical"
    android:textStyle="bold"
    android:id="@+id/textView_help_append"
    android:textColor="#000000"
    android:background="@color/colorAccent"
    android:textSize="20sp" />

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/append_description
"
    android:id="@+id/
textView_help_append_description"
    android:textColor="#000000"
    android:autoLink="all"
    android:linksClickable="true"
    android:layout_marginBottom="15dp" />

<!--Browse PDF-->
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/
textAppearanceLarge"
    android:text="@string/
textView_help_browse_pdf"
    android:gravity="center_vertical"
    android:textStyle="bold"
    android:id="@+id/
textView_help_browse_pdf"
    android:textColor="#000000"
    android:background="@color/colorAccent"
    android:textSize="20sp" />

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/
browse_pdf_description"
    android:id="@+id/
textView_help_browse_pdf_description"
    android:textColor="#000000"
    android:autoLink="all"
    android:linksClickable="true"
    android:layout_marginBottom="15dp" />

<!--Browse Epub-->
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/
textAppearanceLarge"
    android:text="@string/
textView_help_browse_epub"
    android:gravity="center_vertical"
    android:textStyle="bold"
    android:id="@+id/
textView_help_browse_epub"
    android:textColor="#000000"
    android:background="@color/colorAccent"
    android:textSize="20sp" />

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/
```

```
        browse_epub_description"
          android:id="@+id/
        textView_help_browse_epub_description"
          android:textColor="#000000"
          android:autoLink="all"
          android:linksClickable="true"
          android:layout_marginBottom="15dp" />

        <!--Table of contents-->
        <TextView
          android:layout_width="match_parent"
          android:layout_height="wrap_content"
          android:textAppearance="?android:attr/
        textAppearanceLarge"
          android:text="@string/textView_help_toc"
          android:drawableLeft="@mipmap/
        ic_action_document"
          android:gravity="center_vertical"
          android:textStyle="bold"
          android:id="@+id/textView_help_toc"
          android:textColor="#000000"
          android:background="@color/colorAccent"
          android:textSize="20sp" />

        <TextView
          android:layout_width="match_parent"
          android:layout_height="wrap_content"
          android:text="@string/toc_description"
          android:id="@+id/
        textView_help_toc_description"
          android:textColor="#000000"
          android:autoLink="all"
          android:linksClickable="true"
          android:layout_marginBottom="15dp" />

        <!--Save as-->
        <TextView
          android:layout_width="match_parent"
          android:layout_height="wrap_content"
          android:textAppearance="?android:attr/
        textAppearanceLarge"
          android:text="@string/textView_help_save
        "
          android:gravity="center_vertical"
          android:textStyle="bold"
          android:id="@+id/textView_help_save"
          android:textColor="#000000"
          android:background="@color/colorAccent"
          android:textSize="20sp" />

        <TextView
          android:layout_width="match_parent"
          android:layout_height="wrap_content"
          android:text="@string/save_description"
          android:id="@+id/
        textView_help_save_description"
          android:textColor="#000000"
          android:autoLink="all"
          android:linksClickable="true"
          android:layout_marginBottom="15dp" />

        <!--Convert image to searchable PDF-->
        <TextView
          android:layout_width="match_parent"
          android:layout_height="wrap_content"
          android:textAppearance="?android:attr/
        textAppearanceLarge"
          android:text="@string/
        textView_help_toPdf"
          android:gravity="center_vertical"
          android:textStyle="bold"
          android:id="@+id/textView_help_toPdf"
          android:textColor="#000000"
          android:background="@color/colorAccent"
          android:textSize="20sp" />

        <TextView
          android:layout_width="match_parent"
          android:layout_height="wrap_content"
          android:text="@string/toPdf_description"
          android:id="@+id/
        textView_help_toPdf_description"
          android:textColor="#000000"
          android:autoLink="all"
          android:linksClickable="true"
          android:layout_marginBottom="15dp" />

        <!--Convert PDF to Image-->
        <TextView
          android:layout_width="match_parent"
          android:layout_height="wrap_content"
          android:textAppearance="?android:attr/
        textAppearanceLarge"
          android:text="@string/
        textView_help_toImage"
          android:gravity="center_vertical"
          android:textStyle="bold"
          android:id="@+id/textView_help_toImage"
          android:textColor="#000000"
          android:background="@color/colorAccent"
          android:textSize="20sp" />

        <TextView
          android:layout_width="match_parent"
          android:layout_height="wrap_content"
          android:text="@string/
        toImage_description"
          android:id="@+id/
        textView_help_toImage_description"
          android:textColor="#000000"
          android:autoLink="all"
          android:linksClickable="true"
          android:layout_marginBottom="15dp" />

        <!--File directories-->
        <TextView
          android:layout_width="match_parent"
          android:layout_height="wrap_content"
          android:textAppearance="?android:attr/
        textAppearanceLarge"
          android:text="@string/
        textView_help_directories"
          android:gravity="center_vertical"
          android:textStyle="bold"
          android:id="@+id/
        textView_help_directories"
          android:textColor="#000000"
          android:background="@color/colorAccent"
          android:textSize="20sp" />

        <TextView
          android:layout_width="match_parent"
          android:layout_height="wrap_content"
          android:id="@+id/
        textView_help_directories_description"
          android:textColor="#000000"
          android:autoLink="all"
          android:linksClickable="true"
          android:layout_marginBottom="15dp" />

      </LinearLayout>
    </ScrollView>
```

Path = Tess2Speech/tess2Speech/src/main/res/layout/alert_dialog_page_select.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.
    android.com/apk/res/android"
  android:orientation="vertical"
    android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:weightSum="1">

  <RadioGroup
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/radioGroup_pdf">

    <RadioButton
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="All"
      android:id="@+id/radioButton_pdf_all"
      android:checked="true" />

    <RadioButton
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Select Pages:"
      android:id="@+id/radioButton_pdf_select"
      android:checked="false" />
  </RadioGroup>

  <EditText
    android:layout_width="205dp"
    android:layout_height="wrap_content"
    android:inputType="number"
    android:ems="10"
    android:id="@+id/editText_pdf"
    android:digits="0123456789,"
    android:contextClickable="false"
    android:paddingStart="10dp"
    android:paddingEnd="10dp"
    android:phoneNumber="false"
    android:enabled="false"
    android:focusableInTouchMode="false" />

</LinearLayout>
```

Path = Tess2Speech/tess2Speech/src/main/res/layout/content_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.
    android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/
    res-auto"
  xmlns:tools="http://schemas.android.com/
    tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:id="@+id/layout"
  app:layout_behavior="@string/
    appbar_scrolling_view_behavior"
  tools:context=".Activity_Main"
  tools:showIn="@layout/activity_main"
  android:background="@color/colorBackground">

  <Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_play"
    android:id="@+id/button_play"
    android:drawableTop="@mipmap/
     ic_action_playback_play"
    android:background="?android:attr/
     selectableItemBackground"
    android:layout_alignParentBottom="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_marginLeft="5dp"
    android:textSize="11sp" />

  <Button
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_text"
    android:id="@+id/button_text"
    android:drawableTop="@mipmap/
     ic_action_eye_open"
    android:background="?android:attr/
     selectableItemBackground"
    android:layout_alignParentRight="true"
    android:layout_alignParentStart="false"
    android:layout_alignParentEnd="true"
    android:layout_alignBottom="@+id/
     button_play"
    android:layout_marginRight="15dp"
    android:textSize="11sp" />

  <fragment
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:name="anteraaron.tess2speech.
     Fragment_DrawingCanvas"
    android:tag="fragment_canvas"
    android:id="@+id/fragment"
    android:layout_alignParentTop="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_above="@+id/button_text" />
  <fragment
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:name="anteraaron.tess2speech.
     Fragment_ConvertedText"
    android:tag="fragment_text"
    android:id="@+id/fragment2"
    android:layout_alignParentTop="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_above="@+id/button_text" />
  <fragment
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:name="anteraaron.tess2speech.
     Fragment_InputImage"
    android:tag="fragment_image"
    android:id="@+id/fragment3"
    android:layout_alignParentTop="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_above="@+id/button_text" />
  <fragment
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:name="anteraaron.tess2speech.
     Fragment_PdfViewer"
    android:tag="fragment_pdf"
    android:id="@+id/fragment4"
    android:layout_alignParentTop="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_above="@+id/button_text" />
  <fragment
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:name="anteraaron.tess2speech.
     Fragment_EPubViewer_TOC"
    android:tag="fragment_epub_toc"
    android:id="@+id/fragment5"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true" />
  <fragment
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:name="anteraaron.tess2speech.
     Fragment_EPubViewer"
    android:tag="fragment_epub"
    android:id="@+id/fragment6"
    android:layout_alignParentTop="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_above="@+id/button_text" />

</RelativeLayout>
```

Path = Tess2Speech/tess2Speech/src/main/res/layout/fragment_canvas.xml

```
<RelativeLayout xmlns:android="http://schemas.
    android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/
    tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  tools:context=".Fragment_DrawingCanvas"
  android:background="@color/colorBackground">

  <anteraaron.tess2speech.DrawingView
    android:id="@+id/drawing"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFFFFFFF"
    android:layout_gravity="left|top"
    android:layout_below="@+id/button_stroke"
    />

  <Button
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_stroke"
    android:id="@+id/button_stroke"
    android:layout_gravity="center_horizontal|
    top"
    android:drawableTop="@mipmap/ic_eraser"
    android:background="?android:attr/
    selectableItemBackground"
    android:textSize="11sp"
    android:layout_alignTop="@+id/button_clear
    "
    android:layout_toRightOf="@+id/
    button_clear"
    android:layout_toEndOf="@+id/button_clear"
    android:layout_marginLeft="15dp" />

  <Button
```

```
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_clear"
    android:id="@+id/button_clear"
    android:layout_gravity="center_horizontal|
    top"
    android:drawableTop="@mipmap/
    ic_action_present"
    android:background="?android:attr/
    selectableItemBackground"
    android:textSize="11sp"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_marginLeft="10dp" />

</RelativeLayout>
```

Path = Tess2Speech/tess2Speech/src/main/res/layout/fragment_converted_text.xml

```
<RelativeLayout xmlns:android="http://schemas.
    android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/
    tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  tools:context=".Fragment_ConvertedText"
  android:background="#ffffff">

  <!-- TODO: Update blank fragment layout -->

  <Switch
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/switch_append"
    android:layout_gravity="left|bottom"
    android:checked="false"
    android:id="@+id/switch_append"
    android:textSize="21sp"
    android:layout_alignParentBottom="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true" />

  <EditText
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:inputType="textMultiLine"
    android:id="@+id/editText"
    android:gravity="top"
    android:layout_gravity="center_horizontal|
    top"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_alignParentTop="true"
    android:layout_above="@+id/switch_append"
    android:layout_alignParentEnd="true"
    android:padding="15dp"
    android:hint="@string/placeholder" />

</RelativeLayout>
```

Path = Tess2Speech/tess2Speech/src/main/res/layout/fragment_epub.xml

```
<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.
    android.com/apk/res/android"
  android:orientation="vertical"
    android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:weightSum="1"
  android:id="@+id/epub_layout">


  <Button
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_epub_back"
    android:id="@+id/button_epub_back_to_toc"
    android:drawableTop="@mipmap/
    ic_action_document"
    android:background="?android:attr/
    selectableItemBackground"
    android:layout_gravity="left|top"
    android:textSize="11sp" />

  <Button
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_back"
    android:id="@+id/
    button_epub_back_to_canvas"
    android:drawableTop="@mipmap/
    ic_action_present"
    android:background="?android:attr/
    selectableItemBackground"
    android:textSize="11sp"
    android:layout_above="@+id/webView_epub"
    android:layout_toRightOf="@+id/
    button_epub_back_to_toc"
    android:layout_toEndOf="@+id/
    button_epub_back_to_toc"
    android:layout_marginLeft="10dp" />


  <WebView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/webView_epub"
    android:layout_below="@+id/
    button_epub_back_to_toc" />
</RelativeLayout>
```

Path = Tess2Speech/tess2Speech/src/main/res/layout/fragment_input_image.xml

```
<RelativeLayout xmlns:android="http://schemas.
    android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/
    tools"
  xmlns:gesture-image="http://schemas.polites.
    com/android"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  tools:context=".Fragment_InputImage"
  android:background="@color/colorBackground">

  <!-- TODO: Update blank fragment layout -->

  <com.polites.android.GestureImageView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/imageView"
    android:layout_gravity="left|top"
    gesture-image:min-scale="0.1"
    gesture-image:max-scale="10.0"
    gesture-image:strict="false"
    android:layout_below="@+id/button_back" />

  <Button
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_back"
```

```
    android:id="@+id/button_back"
    android:drawableTop="@mipmap/
     ic_action_present"
    android:background="?android:attr/
     selectableItemBackground"
    android:layout_gravity="left|top"
    android:layout_marginLeft="10dp"
    android:textSize="11sp" />

  <Button
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_rotate"
    android:id="@+id/button_rotate"
    android:drawableTop="@mipmap/
     ic_action_turn_left"
    android:background="?android:attr/
     selectableItemBackground"
    android:layout_gravity="right|top"
    android:layout_alignParentTop="true"
    android:layout_toLeftOf="@+id/button_crop"
    android:layout_toStartOf="@+id/button_crop
     "
    android:textSize="11sp" />

  <Button
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_crop"
    android:id="@+id/button_crop"
    android:drawableTop="@mipmap/
     ic_action_crop"
    android:background="?android:attr/
     selectableItemBackground"
    android:layout_gravity="center_horizontal|
     top"
    android:layout_alignParentTop="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"
    android:layout_marginLeft="15dp"
    android:layout_marginRight="10dp"
    android:textSize="11sp" />
</RelativeLayout>
```

Path = Tess2Speech/tess2Speech/src/main/res/layout/fragment_pdf_viewer.xml

```
<FrameLayout xmlns:android="http://schemas.
    android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/
    tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  tools:context="anteraaron.tess2speech.
    Fragment_PdfViewer">

  <com.joanzapata.pdfview.PDFView
    android:id="@+id/pdfView"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" />

  <TextView
    android:layout_width="82dp"
    android:layout_height="wrap_content"
    android:id="@+id/textView_pdf"
    android:layout_gravity="right|bottom"
    android:background="#0a0a0a"
    android:alpha="0.7"
    android:textColor="#ffffff"
    android:textAlignment="center"
    android:textSize="16sp" />

  <Button
    style="?android:attr/buttonStyleSmall"
    android:layout_width="79dp"
    android:layout_height="wrap_content"
    android:text="@string/button_pdf_back"
    android:id="@+id/button_pdf_back"
    android:drawableTop="@mipmap/
     ic_action_present"
    android:background="?android:attr/
     selectableItemBackground"
    android:layout_gravity="left|top"
    android:textSize="11sp" />

</FrameLayout>
```

Path = Tess2Speech/tess2Speech/src/main/res/layout/splash_screen.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.
    android.com/apk/res/android"
  android:orientation="vertical"
    android:layout_width="match_parent"
  android:layout_height="match_parent">

  <ImageView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/imageView_splash"
    android:src="@drawable/splash"
    android:contentDescription="@string/
     imageView_splash_desc"
    android:scaleType="fitXY"
    android:visibility="visible" />

  <TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/textView_splash"
    android:id="@+id/textView_splash"
    android:layout_gravity="center_horizontal|
     bottom"
    android:textColor="#ffffff"
    android:layout_margin="10dp"
    android:visibility="invisible" />

</FrameLayout>
```

Path = Tess2Speech/tess2Speech/src/main/res/layout/view_epub_listview.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.
    android.com/apk/res/android"
  android:orientation="vertical"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:paddingLeft="8dp"
  android:paddingRight="8dp"
  android:background="@color/colorBackground">

  <Button
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_back"
    android:id="@+id/button_epub_toc_back"
    android:drawableTop="@mipmap/
     ic_action_present"
    android:background="?android:attr/
     selectableItemBackground"
    android:layout_gravity="left|top"
    android:textSize="11sp" />

  <ListView android:id="@id/android:list"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:drawSelectorOnTop="false"/>

  <TextView android:id="@id/android:empty"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
</LinearLayout>
```

Path = Tess2Speech/tess2Speech/src/main/res/layout/view_epub_listvie_row.xml

```
<TextView xmlns:android="http://schemas.
    android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:text="New Text"
  android:id="@+id/row"
  android:layout_gravity="center_horizontal"
  android:textColor="#000000"
  android:textSize="20sp"
  android:textIsSelectable="false"
  android:padding="15dp"
  android:background="#ffffff" />
```

Path = Tess2Speech/tess2Speech/src/main/res/layout-land/activity_crop.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.
    android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/
    tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:paddingBottom="@dimen/
    activity_vertical_margin"
  android:paddingLeft="@dimen/
    activity_horizontal_margin"
  android:paddingRight="@dimen/
    activity_horizontal_margin"
  android:paddingTop="@dimen/
    activity_vertical_margin"
  tools:context="anteraaron.tess2speech.
    Activity_Crop"
  android:backgroundTintMode="multiply"
  android:background="@color/colorBackground">


  <com.theartofdev.edmodo.cropper.
    CropImageView
    xmlns:custom="http://schemas.android.com/
    apk/res-auto"
    android:id="@+id/CropImageView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    custom:cropScaleType="fitCenter"
    android:layout_alignParentRight="true"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true"
    android:layout_above="@+id/
    button_crop_cancel"
    android:background="#292929" />

  <Button
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/button_crop_confirm"
    android:drawableBottom="@mipmap/
    ic_action_tick"
    android:background="?android:attr/
    selectableItemBackground"
    android:layout_alignParentBottom="true"
    android:layout_alignParentLeft="true"
    android:layout_marginLeft="56dp"
    android:layout_below="@+id/CropImageView"
    />

  <Button
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/button_crop_cancel"
    android:drawableLeft="@mipmap/
    ic_action_cancel"
    android:background="?android:attr/
    selectableItemBackground"
    android:layout_marginRight="60dp"
    android:layout_marginEnd="60dp"
    android:layout_alignParentBottom="true"
    android:layout_alignEnd="@+id/
    CropImageView" />
</RelativeLayout>
```

Path = Tess2Speech/tess2Speech/src/main/res/layout-land/activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.
    CoordinatorLayout xmlns:android="http://
    schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/
    res-auto"
  xmlns:tools="http://schemas.android.com/
    tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:fitsSystemWindows="true"
  tools:context=".Activity_Main">

  <android.support.design.widget.AppBarLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:theme="@style/AppTheme.
    AppBarOverlay"
    android:paddingLeft="-10dp">

    <android.support.v7.widget.Toolbar
      android:id="@+id/toolbar"
      android:layout_width="match_parent"
      android:layout_height="?attr/
    actionBarSize"
      android:background="@color/colorPrimary"
      app:popupTheme="@style/AppTheme.
    PopupOverlay" />

  </android.support.design.widget.AppBarLayout
    >

  <include layout="@layout/content_main" />

</android.support.design.widget.
    CoordinatorLayout>
```

Path = Tess2Speech/tess2Speech/src/main/res/layout-land/content_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.
    android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/
    res-auto"
  xmlns:tools="http://schemas.android.com/
    tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:id="@+id/layout"
  app:layout_behavior="@string/
    appbar_scrolling_view_behavior"
```

```
    tools:context=".Activity_Main"
    tools:showIn="@layout/activity_main"
    android:background="@color/colorBackground">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_play"
        android:id="@+id/button_play"
        android:drawableTop="@mipmap/
         ic_action_playback_play"
        android:background="?android:attr/
         selectableItemBackground"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_marginLeft="5dp"
        android:textSize="11sp" />

    <Button
        style="?android:attr/buttonStyleSmall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_text"
        android:id="@+id/button_text"
        android:drawableTop="@mipmap/
         ic_action_eye_open"
        android:background="?android:attr/
         selectableItemBackground"
        android:layout_alignParentRight="true"
        android:layout_alignParentStart="false"
        android:layout_alignParentEnd="true"
        android:layout_alignBottom="@+id/
         button_play"
        android:layout_marginRight="15dp"
        android:textSize="11sp" />

    <fragment
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:name="anteraaron.tess2speech.
         Fragment_DrawingCanvas"
        android:tag="fragment_canvas"
        android:id="@+id/fragment"
        android:layout_alignParentTop="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_above="@+id/button_text" />
    <fragment
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:name="anteraaron.tess2speech.
         Fragment_ConvertedText"
        android:tag="fragment_text"
        android:id="@+id/fragment2"
        android:layout_alignParentTop="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_above="@+id/button_text" />
    <fragment
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:name="anteraaron.tess2speech.
         Fragment_InputImage"
        android:tag="fragment_image"
        android:id="@+id/fragment3"
        android:layout_alignParentTop="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_above="@+id/button_text" />
    <fragment
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:name="anteraaron.tess2speech.
         Fragment_PdfViewer"
        android:tag="fragment_pdf"
        android:id="@+id/fragment4"
        android:layout_alignParentTop="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_above="@+id/button_text" />
    <fragment
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:name="anteraaron.tess2speech.
         Fragment_EPubViewer_TOC"
        android:tag="fragment_epub_toc"
        android:id="@+id/fragment5"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true" />
    <fragment
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:name="anteraaron.tess2speech.
         Fragment_EPubViewer"
        android:tag="fragment_epub"
        android:id="@+id/fragment6"
        android:layout_alignParentTop="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_above="@+id/button_text" />


</RelativeLayout>
```

Path = Tess2Speech/tess2Speech/src/main/res/layout-land/fragment_canvas.xml

```
<RelativeLayout xmlns:android="http://schemas.
     android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/
     tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Fragment_DrawingCanvas"
    android:background="@color/colorBackground">

    <anteraaron.tess2speech.DrawingView
        android:id="@+id/drawing"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="#FFFFFFFF"
        android:layout_gravity="left|top"
        android:layout_below="@+id/button_stroke"
        />

    <Button
        style="?android:attr/buttonStyleSmall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_stroke"
        android:id="@+id/button_stroke"
        android:layout_gravity="center_horizontal|
         top"
        android:drawableTop="@mipmap/ic_eraser"
        android:background="?android:attr/
         selectableItemBackground"
        android:textSize="11sp"
        android:layout_alignTop="@+id/button_clear
         "
        android:layout_toRightOf="@+id/
         button_clear"
        android:layout_toEndOf="@+id/button_clear"
        android:layout_marginLeft="15dp" />

    <Button
        style="?android:attr/buttonStyleSmall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_clear"
        android:id="@+id/button_clear"
        android:layout_gravity="center_horizontal|
         top"
        android:drawableTop="@mipmap/
         ic_action_present"
        android:background="?android:attr/
         selectableItemBackground"
        android:textSize="11sp"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_marginLeft="10dp" />

</RelativeLayout>
```

Path = Tess2Speech/tess2Speech/src/main/res/layout-land/fragment_converted_text.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Fragment_ConvertedText"
    android:background="#ffffff">

    <!-- TODO: Update blank fragment layout -->

    <Switch
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/switch_append"
        android:layout_gravity="left|bottom"
        android:checked="false"
        android:id="@+id/switch_append"
        android:textSize="21sp"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true" />

    <EditText
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:inputType="textMultiLine"
        android:id="@+id/editText"
        android:gravity="top"
        android:layout_gravity="center_horizontal|top"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_above="@+id/switch_append"
        android:layout_alignParentEnd="true"
        android:padding="15dp"
        android:hint="@string/placeholder" />

</RelativeLayout>
```

Path = Tess2Speech/tess2Speech/src/main/res/layout-land/fragment_input_image.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:gesture-image="http://schemas.polites.com/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Fragment_InputImage"
    android:background="@color/colorBackground">

    <!-- TODO: Update blank fragment layout -->

    <com.polites.android.GestureImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/imageView"
        android:layout_gravity="left|top"
        gesture-image:min-scale="0.1"
        gesture-image:max-scale="10.0"
        gesture-image:strict="false"
        android:layout_below="@+id/button_back" />

    <Button
        style="?android:attr/buttonStyleSmall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_back"
        android:id="@+id/button_back"
        android:drawableTop="@mipmap/ic_action_present"
        android:background="?android:attr/selectableItemBackground"
        android:layout_gravity="left|top"
        android:layout_marginLeft="10dp"
        android:textSize="11sp" />

    <Button
        style="?android:attr/buttonStyleSmall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_rotate"
        android:id="@+id/button_rotate"
        android:drawableTop="@mipmap/ic_action_turn_left"
        android:background="?android:attr/selectableItemBackground"
        android:layout_gravity="right|top"
        android:layout_alignParentTop="true"
        android:layout_toLeftOf="@+id/button_crop"
        android:layout_toStartOf="@+id/button_crop"
        android:textSize="11sp" />

    <Button
        style="?android:attr/buttonStyleSmall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_crop"
        android:id="@+id/button_crop"
        android:drawableTop="@mipmap/ic_action_crop"
        android:background="?android:attr/selectableItemBackground"
        android:layout_gravity="center_horizontal|top"
        android:layout_alignParentTop="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:layout_marginLeft="15dp"
        android:layout_marginRight="10dp"
        android:textSize="11sp" />
</RelativeLayout>
```

Path = Tess2Speech/tess2Speech/src/main/res/layout-land/fragment_pdf_viewer.xml

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="anteraaron.tess2speech.Fragment_PdfViewer">

    <com.joanzapata.pdfview.PDFView
        android:id="@+id/pdfView"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent" />

    <TextView
        android:layout_width="82dp"
        android:layout_height="wrap_content"
        android:id="@+id/textView_pdf"
        android:layout_gravity="right|bottom"
        android:background="#0a0a0a"
        android:alpha="0.7"
        android:textColor="#ffffff"
        android:textAlignment="center"
        android:textSize="16sp" />

    <Button
```

```
    style="?android:attr/buttonStyleSmall"
    android:layout_width="79dp"
    android:layout_height="wrap_content"
    android:text="@string/button_pdf_back"
    android:id="@+id/button_pdf_back"
    android:drawableTop="@mipmap/
    ic_action_present"
    android:background="?android:attr/
    selectableItemBackground"
    android:layout_gravity="left|top"
    android:textSize="11sp" />

</FrameLayout>
```

Path = Tess2Speech/tess2Speech/src/main/res/layout-land/view_epub_listview.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.
    android.com/apk/res/android"
  android:orientation="vertical"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:paddingLeft="8dp"
  android:paddingRight="8dp"
  android:background="@color/colorBackground">

  <Button
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_back"
    android:id="@+id/button_epub_toc_back"
    android:drawableTop="@mipmap/
    ic_action_present"
    android:background="?android:attr/
    selectableItemBackground"
    android:layout_gravity="left|top"
    android:textSize="11sp" />

  <ListView android:id="@id/android:list"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:drawSelectorOnTop="false" />

  <TextView android:id="@id/android:empty"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
</LinearLayout>
```

Path = Tess2Speech/tess2Speech/src/main/res/menu/menu_main.xml

```
<menu xmlns:android="http://schemas.android.
    com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/
    res-auto"
  xmlns:tools="http://schemas.android.com/
    tools"
  tools:context=".Activity_Main">

  <item
    android:id="@+id/camera"
    android:orderInCategory="100"
    android:title="@string/camera"
    android:icon="@mipmap/ic_action_camera"
    app:showAsAction="always" />
  <item
    android:id="@+id/browse"
    android:orderInCategory="100"
    android:title="@string/browse"
    android:icon="@mipmap/ic_action_picture"
    app:showAsAction="ifRoom" />
  <item
    android:id="@+id/browsePdf"
    android:orderInCategory="100"
    android:title="@string/browsePdf"
    app:showAsAction="never" />
  <item
    android:id="@+id/browseEpub"
    android:orderInCategory="100"
    android:title="@string/browseEpub"
    app:showAsAction="never" />
  <item
    android:id="@+id/saveText"
    android:orderInCategory="100"
    android:title="@string/save_text"
    app:showAsAction="never" />
  <item
    android:id="@+id/savePdf"
    android:orderInCategory="100"
    android:title="@string/save_pdf"
    app:showAsAction="never" />
  <item
    android:id="@+id/saveAudio"
    android:orderInCategory="100"
    android:title="@string/save_audio"
    app:showAsAction="never" />
  <item
    android:id="@+id/imageToPdf"
    android:orderInCategory="100"
    android:title="@string/image_to_pdf"
    app:showAsAction="never" />
  <item
    android:id="@+id/pdfToImage"
    android:orderInCategory="100"
    android:title="@string/pdf_to_image"
    app:showAsAction="never" />
  <item
    android:id="@+id/settings"
    android:orderInCategory="100"
    android:title="@string/settings"
    app:showAsAction="never" />
  <item
    android:id="@+id/exit"
    android:orderInCategory="100"
    android:title="@string/exit"
    app:showAsAction="never" />

</menu>
```

Path = Tess2Speech/tess2Speech/src/main/res/values/colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <color name="colorPrimary">#7B1113</color>
  <color name="colorPrimaryDark">#660101</
    color>
  <color name="colorAccent">#117b79</color>
  <color name="colorBackground"> #e6e0e0</
    color>
</resources>
```

Path = Tess2Speech/tess2Speech/src/main/res/values/dimens.xml

```
<resources>
  <!-- Default screen margins, per the Android
      Design guidelines. -->
  <dimen name="activity_horizontal_margin">16
     dp</dimen>
  <dimen name="activity_vertical_margin">16dp<
     /dimen>
  <dimen name="fab_margin">16dp</dimen>
</resources>
```

Path = Tess2Speech/tess2Speech/src/main/res/values/strings.xml

```
<resources>

  <!--App Name-->
  <string name="app_name">Tess2Speech</string>

  <!--Action Bar Menu -->
  <string name="camera">Use Camera for input</
     string>
  <string name="browse">Browse input image <
     /string>
  <string name="browsePdf">Browse input PDF
     </string>
  <string name="browseEpub">Browse input
     Epub </string>
  <string name="save_text">Save as .txt</
     string>
  <string name="save_pdf">Save as .pdf</string
     >
  <string name="save_audio">Save Audio</string
     >
  <string name="image_to_pdf">Convert image to
      searchable PDF</string>
  <string name="pdf_to_image">Convert Pdf to
     Image</string>
  <string name="settings">Settings</string>
  <string name="exit">Exit</string>

  <!--App UI strings-->
  <!--content_main.xml-->
  <string name="button_play">Play</string>
  <string name="button_stop">Stop</string>
  <string name="button_text">Convert Text\
     u0020</string>
  <string name="button_text_back">Back to
     Input</string>
  <!--activity_rotate.xml-->
  <string name="imageView_rotate_content_desc"
     >Input image</string>
  <!--fragment_canvas.xml-->
  <string name="button_stroke">Erase</string>
  <string name="button_stroke_stroke">Stroke</
     string>
  <string name="button_clear">Clear</string>
  <!--fragment_converted_text.xml-->
  <string name="switch_append">Append Text:</
     string>
  <string name="placeholder">Input text here</
     string>
  <!--fragment_input_image.xml etc.-->
  <string name="button_back">Canvas</string>
  <string name="button_rotate">Rotate</string>
  <string name="button_crop">Crop</string>
  <!--fragment_pdf_viewer.xml-->
  <string name="button_pdf_back">Canvas</
     string>
  <!--splash_screen.xml-->
  <string name="imageView_splash_desc">Splash<
     /string>
  <string name="textView_splash">Copying files
      please wait     </string>
  <!--fragment_epub.xml-->
  <string name="button_epub_back">Table of
     Contents</string>



  <!--Preferences Menu-->
  <string name="title_image_preprocess">Image
     Pre-processing</string>
  <string name="summary_image_preprocess"><![
     CDATA[Perform Grayscale and fixes poor
     illuminated images before converting to
     text. May improve accuracy but slows
     conversion time.]]></string>
  <string name="title_accelerate">Re-scale
     Image</string>
  <string name="summary_accelerate">Speeds up
     conversion speed by reducing image size.
     Turn this on if the image has large texts
     . Turn this off for images with small
     texts such as documents etc.</string>
  <string name="title_change_language">
     Tesseract Trained Data</string>
  <string name="summary_change_language">
     Change Tesseract\'s trained data. You can
      combine two or more trained data. Make
     sure to also change Text-to-Speech
     language and Spell-Checker if other
     languages other than English is selected
     .</string>
  <string name="title_intent_tts">TTS
     Settings</string>
  <string name="pref_settings_category">
     Settings</string>
  <string name="summary_intent_tts">Text-to-
     Speech Settings</string>
  <string name="title_intent_spell_check">
     Spell checker</string>
  <string name="summary_intent_spell_check
     "><![CDATA[Launch Language & Input
     settings. Turning Spell checker on
     can help you recognize questionable
     words and suggest correct words.]]></
     string>
  <string name="pref_about_category">About</
     string>
  <string name="title_help">Help</string>
  <string name="title_license">License</
     string>
  <string name="title_version">Version</
     string>
  <string name="summary_version">v1.00(Beta)
     </string>

  <!--Licenses-->
  <string name="tesstwo_license">
     by Robert Theis \n<a href="https://
        github.com/rmtheis/tess-two">
        https://github.com/rmtheis/tess-
        two</a>\n\nCopyright 2011 Robert
        Theis\nLicensed under the Apache
        License, Version 2.0
  </string>
  <string name="gestureimageview_license">
     by jasonpolites \n<a href="https://
        github.com/jasonpolites/gesture-
        imageview">https://github.com/
        jasonpolites/gesture-imageview</a
        >\n\nLicensed under the Apache
        License, Version 2.0
  </string>
  <string name="androidpdfview_license">
     by Joan Zapata \n<a href="https://
        github.com/JoanZapata/android-
        pdfview">https://github.com/
        JoanZapata/android-pdfview</a>\n\
        nCopyright 2013-2015 Joan Zapata\
        nLicensed under GNU General
        Public License v3 (compatible
        with Apache License, Version 2.0)
     \nLicense available at: <a href="http:
        //www.gnu.org/licenses">http://
        www.gnu.org/licenses</a>
  </string>
  <string name="cropper_license">
     by ArthurHub forked from <a href="
        https://github.com/edmodo/cropper
        ">https://github.com/edmodo/
        cropper</a> \n\n<a href="https://
        github.com/ArthurHub/Android-
        Image-Cropper">https://github.com
        /ArthurHub/Android-Image-Cropper
        </a>\n\nCopyright 2013 Edmodo,
        Inc.\nLicensed under the Apache
        License, Version 2.0
  </string>
  <string name="epub_license">
     by Paul Siegman from <a href="https://
        github.com/psiegman/epublib">
        https://github.com/psiegman/
        epublib</a>
  </string>
  <string name="sl4j_license">
     Copyright     2004-2013 QOS.ch \n
     All rights reserved.
     \n\nPermission is hereby granted, free
          of charge, to any person
        obtaining
```

```
        a copy of this software and
            associated documentation files
            (the
        "Software"), to deal in the Software
            without restriction, including
        without limitation the rights to use
            , copy, modify, merge, publish,
        distribute, sublicense, and/or sell
            copies of the Software, and to
        permit persons to whom the Software
            is furnished to do so, subject to
        the following conditions:

        \n\nThe above copyright notice
            and this permission notice
            shall be
        included in all copies or substantial
            portions of the Software.

        \n\nTHE SOFTWARE IS PROVIDED "AS
            IS", WITHOUT WARRANTY OF ANY
            KIND,
        EXPRESS OR IMPLIED, INCLUDING BUT
            NOT LIMITED TO THE WARRANTIES
            OF
        MERCHANTABILITY, FITNESS FOR
            A PARTICULAR PURPOSE
            AND
        NONINFRINGEMENT. IN NO EVENT SHALL
            THE AUTHORS OR COPYRIGHT HOLDERS
            BE
        LIABLE FOR ANY CLAIM, DAMAGES OR
            OTHER LIABILITY, WHETHER IN AN
            ACTION
        OF CONTRACT, TORT OR OTHERWISE,
            ARISING FROM, OUT OF OR IN
            CONNECTION
        WITH THE SOFTWARE OR THE USE OR OTHER
            DEALINGS IN THE SOFTWARE. <a
            href="http://www.slf4j.org/
            license.html">http://www.slf4j.
            org/license.html</a>
    </string>
    <string name="file_picker_license">
        by Anders Kalr \n<a href="https://
            github.com/AndersKaloer/Android-
            File-Picker-Activity">https://
            github.com/AndersKaloer/Android-
            File-Picker-Activity</a>\n\
            nCopyright 2011 Anders Kalr\
            nLicensed under the Apache
            License, Version 2.0
    </string>
    <string name="icons_license">
        Androidicons is a product by Opoloo.
            You can follow their Twitter feed
            at @opoloo, their G+ site at
            google.com/+Opoloo, or contact
            them via mail at info@opoloo.com
            concerning any questions or
            suggestions. -Guenther Beyer\n<a
            href="http://www.androidicons.com
            /">http://www.androidicons.com/</
            a>
        \n\nLicensed under Creative Commons
            Attribution-ShareAlike 4.0
            International Public License <a
            href="http://creativecommons.org/
            licenses/by-sa/4.0/">http://
            creativecommons.org/licenses/by-
            sa/4.0/</a>.
    </string>
    <string name="tess2Speech_license">
        by Anter Aaron Custodio \n\nCopyright
            2016 Anter Aaron Custodio\
            nLicensed under the Apache
            License, Version 2.0 (the "
            License");
        you may not use this file except in
            compliance with the License. You
            may obtain a copy of the License
            at
        \n<a href="http://www.apache.org/
            licenses/LICENSE-2.0">http://www.
            apache.org/licenses/LICENSE-2.0</
            a>
        \nUnless required by applicable law or
            agreed to in writing, software
        distributed under the License is
            distributed on an "AS IS" BASIS,
        WITHOUT WARRANTIES OR CONDITIONS OF
            ANY KIND, either express or
            implied.
        See the License for the specific
            language governing permissions
            and
        limitations under the License.
    </string>

    <!--Help-->
    <string name="textView_help_title">User
        Manual</string>
    <string name="textView_help_tips">Tips</
        string>
    <string name="tips_description">1. If the
        image dimension is large and only
        contains large texts, it is better to
        turn on \"Re-scale Image\" to avoid
        reading garbage texts.
        \n\n2. If the image contains small
            texts (such as PDFs, docs, etc.),
            turn off \"Re-scale image\" to
            increase accuracy.
        \n\n3. If the image contains non-black
            characters and non-white
            background, it is better to turn
            on \"Image Pre-processing\".
        \n\n4. In settings > Tesseract
            Language:
        \n\t eng.traineddata - check this for
            printed texts.
        \n\t engh.traineddata - check this for
            handwritten texts.
        \n*You can check both to read both
            handwritten and printed texts,
            but this may affect each others
            accuracy.
        \n\n5. If you want to add another .
            traineddata, Copy the desired .
            traineddata file to Storage/
            Android/data/anteraaron.
            tess2speech/files/tessdata
    </string>
    <string name="textView_help_camera">Camera
        </string>
    <string name="camera_description">Use the
        phone\'s camera to get an input image
        which will be converted to text.</
        string>
    <string name="textView_help_browse_image">
        Browse Image</string>
    <string name="browse_image_description">Get
        an input image from the phone\'s storage
        which will be converted to text.</string>
    <string name="textView_help_play">Play</
        string>
    <string name="play_description">Converts
        the image displayed/written words
        from canvas to text and then converts
        it to speech.</string>
    <string name="textView_help_stop">Stop</
        string>
    <string name="stop_description">Stops the
        Text-to-Speech playback.</string>
    <string name="textView_help_clear">Clear</
        string>
    <string name="clear_description">Deletes
        all writings in the writing canvas.</
        string>
    <string name="textView_help_erase">Erase</
        string>
    <string name="erase_description">Erases a
        stroke in the writing canvas instead
        of writing.</string>
    <string name="textView_help_stroke">Stroke
        </string>
    <string name="stroke_description">Writes a
        stroke to the writing canvas.</
        string>
    <string name="textView_help_convert_text">
        Convert Text</string>
    <string name="convert_text_description">
        Converts the image displayed/written
        words from canvas to text and display
        the editable text.</string>
    <string name="textView_help_back">Back to
        Input</string>
    <string name="back_description">Go back to
        the input image/writing canvas.</
        string>
    <string name="textView_help_rotate">Rotate
        </string>
    <string name="rotate_description">Rotates
        the image displayed.</string>
    <string name="textView_help_crop">Crop</
        string>
    <string name="crop_description">Crops the
        image displayed.</string>
    <string name="textView_help_append">Append
        Text</string>
    <string name="append_description">Does not
        clear the previously converted text.
        The next converted text will be
        appended to the previously converted
        text. (Does not work with Epubs
        because Epub texts are too long).</
        string>
```

```
    <string name="textView_help_browse_pdf">
        Browse input PDF</string>
    <string name="browse_pdf_description">Gets
        a PDF from the phone\'s storage as
        input. Note that converting a PDF to
        text or speech prompts you to select
        which pages to convert.</string>
  <string name="textView_help_browse_epub">
     Browse input Epub</string>
  <string name="browse_epub_description">Gets
     a Epub from the phone\'s storage as input
     . </string>
    <string name="textView_help_toc">Table of
        Contents</string>
    <string name="toc_description">Go back to
        the overview of the book\'s chapters.
        </string>
  <string name="textView_help_save">Save as .
     txt/.pdf/audio</string>
  <string name="save_description">Saves the
     converted text as a text file/PDF file.
     Save Audio saves the converted text\'s
     speech equivalent as a WAV file.</string>
    <string name="textView_help_toPdf">Convert
        image/s to searchable PDF.</string>
    <string name="toPdf_description">Converts
        an image to PDF which has searchable
        and selectable texts. The font size
        and style in the PDF will be the same
        as the recognized font in the image.
        (Note that multiple image select is
        only available for API 18 (JELLY BEAN
        MR2) and up. Multiple image select
        is also not available for this app\'s
        built in file browser.)</string>
  <string name="textView_help_toImage">Convert
     PDF to image</string>
  <string name="toImage_description">Converts
     the selected pages of the PDF to images
     and save it to a specific folder.</string
     >
  <string name="textView_help_directories">
     File Directories</string>
</resources>
```

Path = Tess2Speech/tess2Speech/src/main/res/values/styles.xml

```
<resources>

  <!-- Base application theme. -->
  <style name="AppTheme" parent="Theme.
     AppCompat.Light.DarkActionBar">
    <!-- Customize your theme here. -->
    <item name="colorPrimary">@color/
     colorPrimary</item>
    <item name="colorPrimaryDark">@color/
     colorPrimaryDark</item>
    <item name="colorAccent">@color/
     colorAccent</item>
    <item name="windowActionModeOverlay">true<
     /item>
  </style>

  <style name="AppTheme.NoActionBar">
    <item name="windowActionBar">false</item>
    <item name="windowNoTitle">true</item>
  </style>

  <style name="AppTheme.AppBarOverlay" parent=
     "ThemeOverlay.AppCompat.Dark.ActionBar" /
     >

  <style name="AppTheme.PopupOverlay" parent="
     ThemeOverlay.AppCompat.Light" />

</resources>
```

Path = Tess2Speech/tess2Speech/src/main/res/values-v21/styles.xml

```
<resources>

  <style name="AppTheme.NoActionBar">
    <item name="windowActionBar">false</item>
    <item name="windowNoTitle">true</item>
    <item name="
     android:windowDrawsSystemBarBackgrounds">
     true</item>
    <item name="android:statusBarColor">
     @android:color/transparent</item>
  </style>
</resources>
```

Path = Tess2Speech/tess2Speech/src/main/res/values-w820dp/dimens.xml

```
<resources>
  <!-- Example customization of dimensions
     originally defined in res/values/dimens.
     xml
     (such as screen margins) for screens with
     more than 820dp of available width. This
     would include 7" and 10" devices in
     landscape (~960dp and ~1280dp
     respectively). -->
  <dimen name="activity_horizontal_margin">64
     dp</dimen>
</resources>
```

Path = Tess2Speech/tess2Speech/src/main/res/xml/preferences.xml

```
<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen xmlns:android="http://
     schemas.android.com/apk/res/android">
  <PreferenceCategory android:title="@string/
     pref_settings_category">
    <CheckBoxPreference
      android:key="pref_key_image_preprocess"
      android:title="@string/
     title_image_preprocess"
      android:summary="@string/
     summary_image_preprocess"
      android:defaultValue="true">
    </CheckBoxPreference>
    <CheckBoxPreference
      android:key="pref_key_accelerate"
      android:title="@string/title_accelerate"
      android:summary="@string/
     summary_accelerate"
```

```
    android:defaultValue="false">
</CheckBoxPreference>
<PreferenceScreen
    android:key="pref_key_change_language"
    android:title="@string/
 title_change_language"
    android:summary="@string/
 summary_change_language">
</PreferenceScreen>
<PreferenceScreen
    android:title="@string/title_intent_tts"
    android:summary="@string/
 summary_intent_tts">
   <intent android:action="com.android.
 settings.TTS_SETTINGS" />
</PreferenceScreen>
<PreferenceScreen
    android:key="pref_key_spell_check"
    android:title="@string/
 title_intent_spell_check"
    android:summary="@string/
 summary_intent_spell_check"
    android:enabled="true">

   <intent android:action="android.intent.
 action.MAIN"
       android:targetPackage="com.android.
        settings"
          android:targetClass="com.android.
        settings.LanguageSettings" />
    </PreferenceScreen>
  </PreferenceCategory>

  <PreferenceCategory android:title="@string/
      pref_about_category">
    <PreferenceScreen
      android:key="pref_key_help"
      android:title="@string/title_help"
      android:summary="">
    </PreferenceScreen>
    <PreferenceScreen
      android:key="pref_key_license"
      android:title="@string/title_license"
      android:summary="">
    </PreferenceScreen>
    <PreferenceScreen
      android:title="@string/title_version"
      android:summary="@string/summary_version
     ">
    </PreferenceScreen>
  </PreferenceCategory>

</PreferenceScreen>
```

B..2 Tess2Speech Trainer

Tessdata folder, edited jTessBoxEditor folder, training tools folder should be all in the same directory.

Base Path = Tess2SpeechTrainer/

Path = Tess2SpeechTrainer/src/anteraaron/tess2speech/Driver.java

```
package anteraaron.tess2speech;

import javax.swing.SwingUtilities;
import javax.swing.
    UnsupportedLookAndFeelException;
/**
 * Driver for the whole Tess2Speech Trainer
 * @author Anter Aaron M. Custodio
 * @since 2016/5/2
 */
public class Driver {

  /**
   * Main method
   * @param args String arguments for main
   */
  public static void main(String[] args) {
    //Launch the GUI frame
    SwingUtilities.invokeLater(new Runnable()
        {
    @Override
    public void run() {
        try {
          new Frame();
        } catch (ClassNotFoundException e) {
          // TODO Auto-generated catch block
          e.printStackTrace();
        } catch (InstantiationException e) {
          // TODO Auto-generated catch block
          e.printStackTrace();
        } catch (IllegalAccessException e) {
          // TODO Auto-generated catch block
          e.printStackTrace();
        } catch (
            UnsupportedLookAndFeelException e
            ) {
          // TODO Auto-generated catch block
          e.printStackTrace();
        } //Instantiate the frame
      }
    });
    }

}
```

Path = Tess2SpeechTrainer/src/anteraaron/tess2speech/Frame.java

```
package anteraaron.tess2speech;

import java.awt.Dimension;

import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.UIManager;
import javax.swing.
    UnsupportedLookAndFeelException;

/**
 * The Frame of the Tess2Speech GUI
 * @author Anter Aaron M. Custodio
 * @since 2016/5/2
 */
public class Frame {
  final private JFrame frame;
  /**
   * Constructor
   * @throws UnsupportedLookAndFeelException
   * @throws IllegalAccessException
   * @throws InstantiationException
   * @throws ClassNotFoundException
   */
  public Frame() throws ClassNotFoundException
      , InstantiationException ,
      IllegalAccessException ,
      UnsupportedLookAndFeelException {
    UIManager.setLookAndFeel(
        "com.sun.java.swing.plaf.windows.
          WindowsLookAndFeel");
    this.frame = new JFrame("Tess2Speech
        Trainer");
    this.frame.setDefaultCloseOperation(JFrame
        .EXIT_ON_CLOSE);
    this.frame.setSize(new Dimension(400,440))
        ;
    this.frame.setResizable(false);
    this.frame.setLocationRelativeTo(null);
    this.frame.setVisible(true);
    this.frame.setIconImage(new ImageIcon(
        getClass().getResource("/images/logo.
        png")).getImage());

    this.frame.setContentPane(new MainPanel())
        ;
  }
}
```

Path = Tess2SpeechTrainer/src/anteraaron/tess2speech/Functions.java

```
package anteraaron.tess2speech;

import java.awt.HeadlessException;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;
import java.io.FilenameFilter;
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.StandardCopyOption;
import java.util.ArrayList;
import java.util.Arrays;
import javax.swing.JOptionPane;

/**
 * A Class that contains all of the
     functionalities of Tess2Speech trainer
 * @author Anter Aaron M. Custodio
 * @since 2016/5/3
 */
public class Functions {

  private static final String OUTPUT_DIR = "
```

```
    output";
private static final String TESSDATA_DIR = "
    tessdata";
private static final String
    TRAINING_TOOLS_DIR = "training_tools";
private static final String TESSERACT_CMD =
    "training_tools" + File.separator + "
    tesseract --tessdata-dir " + ".\\ ";
private static final String
    UNICHARSET_EXTRACTOR_CMD = "
    training_tools" + File.separator + "
    unicharset_extractor ";
private static final String MFTRAINING_CMD =
     "training_tools" + File.separator + "
    mftraining ";
private static final String
    SET_UNICHARSET_PROPERTIES_CMD = "
    training_tools" + File.separator + "
    set_unicharset_properties ";
private static final String CNTRAINING_CMD =
     "training_tools" + File.separator + "
    cntraining ";
private static final String
    WORDLIST2DAWG_CMD = "training_tools" +
    File.separator + "wordlist2dawg ";
private static final String
    COMBINE_TESSDATA_CMD = "training_tools"
     + File.separator + "combine_tessdata "
    ;
private static final String
    JTESSBOXEDITOR_DIR = "jTessBoxEditor";
/**
 * Start Tesseract Training
 */
public static void startTraining() {
  //Remove empty file paths
  final String tesseractLanguage = MainPanel
      .language_combobox.getSelectedItem().
      toString();
  final String languageName = MainPanel.
      font_textfield.getText();
  final String[] paths = MainPanel.
      tiff_textfield.getText().split(";");
  final ArrayList<String> list = new
      ArrayList<String>(Arrays.asList(paths
      ));
  list.removeAll(Arrays.asList("", null));


  //Create language dir
  final File directory = new File(OUTPUT_DIR
       + File.separator + languageName);

  if (directory.exists()) {
    final int choice = JOptionPane.
        showOptionDialog(null, "The
        Language '" + languageName + "'
        already exists. Previous training
        images will be used again. Continue
         Anyway?", "Warning", JOptionPane.
        OK_CANCEL_OPTION, JOptionPane.
        WARNING_MESSAGE, null, null, null);
    if (choice == JOptionPane.CANCEL_OPTION)
        {
      return;
    }
  } else {
    directory.mkdirs();
  }

  //Rename and Copy files to output
      directory
  String[] temp;
  String fileName;
  File sourceFile;
  File destinationFile;

  for(final String sourcePath : list){
    temp = new File(sourcePath).getName().
        split("\\.");

    //If format is lang.font.exp0.tif
    if (temp.length > 2) {
      fileName = languageName + "." + temp
          [1] + ".exp0.tif";
    } else {
      //Else if format is just image.tif
      fileName =  languageName + "." + temp
          [0] + ".exp0.tif";
    }

    //Prepare source file and destination
        file for copying
    sourceFile = new File(sourcePath);
    destinationFile = new File(OUTPUT_DIR +
        File.separator + languageName +
        File.separator + fileName);

    try {
      //Check if image file is already
          existing in folder, prompt user
          to whether to overwrite
      if (destinationFile.exists()) {
        final int choice = JOptionPane.
            showOptionDialog(null, "The
            image '" + fileName + "'
            already exists. Overwrite?", "
            Warning", JOptionPane.
            OK_CANCEL_OPTION, JOptionPane.
            WARNING_MESSAGE, null, null,
            null);
        if (choice == JOptionPane.OK_OPTION)
            {
          Files.copy(sourceFile.toPath(),
              destinationFile.toPath(),
              StandardCopyOption.
              REPLACE_EXISTING);
        }
      } else {
        Files.copy(sourceFile.toPath(),
            destinationFile.toPath(),
            StandardCopyOption.
            REPLACE_EXISTING);
      }

    } catch (IOException e) {
      JOptionPane.showMessageDialog(null, "
          Source File '" + sourcePath + "'
          not Found!", "Failure",
          JOptionPane.ERROR_MESSAGE);
      e.printStackTrace();
    }
  }

  //Get list of file without extension
  final File[] files = new File(OUTPUT_DIR +
      File.separator + languageName).
      listFiles( new FilenameFilter() {

    @Override
    public boolean accept(File dir, String
        name) {
      return name.toLowerCase().endsWith(".
          tif");
    }
  });

  String[] fileNames = new String[files.
      length];

  //format output\lang\font
  for(int i = 0; i < files.length; i++) {
    fileNames[i] = OUTPUT_DIR + File.
        separator + languageName + File.
        separator + files[i].getName().
        substring(0, files[i].getName().
        lastIndexOf('.'));
  }


  //Start creating command for cmd
  String command = "cmd /c start /wait cmd.
      exe /c\"";
  for(int i = 0; i < fileNames.length; i++)
      {
    if (!new File(fileNames[i] + ".box").
        exists() || MainPanel.checkBox.
        isSelected()) {
      command += TESSERACT_CMD + "-l " +
          tesseractLanguage + " -psm 6 " +
          fileNames[i] + ".tif " +
          fileNames[i] + " batch.nochop
          makebox && ";
    }
  }
  command += "java -jar " +
      JTESSBOXEDITOR_DIR + File.separator +
      "jTessBoxEditor.jar";
  command += " " + "\"" + new File(
      OUTPUT_DIR + File.separator +
      languageName) + "\"";
  command += " && " + OUTPUT_DIR + File.
      separator + languageName + File.
      separator + "train.bat";



  command += "\"";

  //Run command for generating box file
  try {
    createBatchFile(fileNames, languageName,
        tesseractLanguage);
    Runtime.getRuntime().exec(command).
        waitFor();
    final int choice = JOptionPane.
        showConfirmDialog(null,
        languageName + ".traineddata was
```

```
            created successfully! File saved at
             Tess2Speech Trainer/tessdata. Open
             tessdata folder?", "Successful",
            JOptionPane.YES_NO_OPTION);

        if(choice == JOptionPane.YES_OPTION) {
          Runtime.getRuntime().exec("explorer.
              exe tessdata");
        }

      } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
      } catch (HeadlessException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
      } catch (InterruptedException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
      }

    }

    /**
     * A method that creates a batch file for
         Automatic Tesseract Training
     * @param fileNames fileNames of the
         training data
     * @param languageName specified name of
         output trained data language
     * @param tesseractLanguage specified
         Tesseract language for recognizing box
          files
     * @throws IOException Error
     */
    private static void createBatchFile(String[]
        fileNames, String languageName, String
        tesseractLanguage) throws IOException{
      //Create command for batch file

      //Set directory to language being created
      String languagePath =  OUTPUT_DIR + File.
          separator + languageName + File.
          separator;

      //Set to this File's current directory and
           up by two folders
      String batchCommand = "cd /d %~dp0\ncd
          ../../\n";

      //Generate .tr file
      for (int i = 0; i < fileNames.length; i++)
          {
        //batchCommand += TESSERACT_CMD + "-l "
            + tesseractLanguage + " -psm 6 ";
        batchCommand += TESSERACT_CMD + "-psm 6
            ";
        batchCommand += fileNames[i] + ".tif ";
        batchCommand += fileNames[i] + " ";
        //batchCommand += "batch.nochop box.
            train && ";
        batchCommand += "box.train && ";
      }

      //Unicharset extractor'
      batchCommand += UNICHARSET_EXTRACTOR_CMD;
      for (int i = 0; i < fileNames.length; i++)
          {
        batchCommand += fileNames[i] + ".box ";
      }
      batchCommand += "&& move /y unicharset " +
          languagePath;

      //font propertes file
      final File fontPropertiesFile = new File(
          languagePath + "font_properties.sh");
      BufferedWriter writer = new BufferedWriter
          (new FileWriter(fontPropertiesFile));
      String fontPropertiesContent = "";
      String fontName = "";
      for (int i = 0; i < fileNames.length; i++)
          {
        fontName = fileNames[i].substring(
            fileNames[i].lastIndexOf("\\") + 1,
             fileNames[i].length() - 1);
        fontName = fontName.split("\\.")[1];
        fontPropertiesContent += fontName + " 0
            0 0 0 0\n";
      }
      writer.write(fontPropertiesContent);
      writer.flush();
      writer.close();

      //mftraining
      batchCommand += " && " + MFTRAINING_CMD +
          "-F " + languagePath + "
          font_properties.sh -U " +
          languagePath +
          "unicharset -O "+ languagePath +
          languageName + ".unicharsettemp "
          ;
      for (int i = 0; i < fileNames.length; i++)
          {
        batchCommand += fileNames[i] + ".tr ";
      }

      //set_unicharset_properties
      batchCommand += "&& " +
          SET_UNICHARSET_PROPERTIES_CMD + "-U "
          + languagePath + languageName +
          ".unicharsettemp -O " + languagePath +
           languageName + ".unicharset -
          script_dir=" + TRAINING_TOOLS_DIR
          + " && ";

      //cntraining
      batchCommand += CNTRAINING_CMD;
      for (int i = 0; i < fileNames.length; i++)
          {
        batchCommand += fileNames[i] + ".tr ";
      }

      //Optional files

      //config file
      batchCommand += "&& xcopy /y " +
          TRAINING_TOOLS_DIR + File.separator +
          "lang.config " + languagePath;
      batchCommand += " && del " + languagePath
          + languageName + ".config 2>NUL";
      batchCommand += " && ren " + languagePath
          + "lang.config " + languageName + ".
          config";

      //unicharambigs file
      batchCommand += " && xcopy /y " +
          TRAINING_TOOLS_DIR + File.separator +
          "lang.unicharambigs " + languagePath
          ;
      batchCommand += " && del " + languagePath
          + languageName + ".unicharambigs 2>
          NUL";
      batchCommand += " && ren " + languagePath
          + "lang.unicharambigs " +
          languageName + ".unicharambigs";

      //freq-dawg
      batchCommand += " && " + WORDLIST2DAWG_CMD
          + TRAINING_TOOLS_DIR + File.
          separator + "lang.freqwords.txt " +
          languagePath + languageName + ".freq-
          dawg " + languagePath +
          languageName + ".unicharset";

      //wird-dawg
      batchCommand += " && " + WORDLIST2DAWG_CMD
          + TRAINING_TOOLS_DIR + File.
          separator + "lang.wordlist " +
          languagePath + languageName + ".word-
          dawg " + languagePath +
          languageName + ".unicharset";

      //bigrams
      batchCommand += " && " + WORDLIST2DAWG_CMD
          + TRAINING_TOOLS_DIR + File.
          separator + "lang.bigrams " +
          languagePath + languageName + ".bigram
          -dawg " + languagePath +
          languageName + ".unicharset";

      //punc-dawg
      batchCommand += " && " + WORDLIST2DAWG_CMD
          + TRAINING_TOOLS_DIR + File.
          separator + "lang.punc " +
          languagePath + languageName + ".punc-
          dawg " + languagePath +
          languageName + ".unicharset";

      //number-dawg
      batchCommand += " && " + WORDLIST2DAWG_CMD
          + TRAINING_TOOLS_DIR + File.
          separator + "lang.numbers " +
          languagePath + languageName + ".number
          -dawg " + languagePath +
          languageName + ".unicharset";

      //Rename and move required files
      batchCommand += " && ren normproto " +
          languageName + ".normproto";
      batchCommand += " && move /y " +
          languageName + ".normproto " +
          languagePath;

      batchCommand += " && ren inttemp " +
          languageName + ".inttemp";
      batchCommand += " && move /y " +
          languageName + ".inttemp " +
          languagePath;
```

```
        batchCommand += " && ren pffmtable " +
            languageName + ".pffmtable";
        batchCommand += " && move /y " +
            languageName + ".pffmtable " +
            languagePath;

        batchCommand += " && ren shapetable " +
            languageName + ".shapetable";
        batchCommand += " && move /y " +
            languageName + ".shapetable " +
            languagePath;

        batchCommand += " && " +
            COMBINE_TESSDATA_CMD + languagePath +
             languageName;
        batchCommand += " && move /y " +
            languagePath + languageName + ".
            traineddata " + TESSDATA_DIR + File.
            separator;


        final File batchFile = new File(
            languagePath + "train.bat");

        writer = new BufferedWriter(new FileWriter
            (batchFile));
        writer.write(batchCommand);
        writer.flush();
        writer.close();
    }
}
```

Path = Tess2SpeechTrainer/src/anteraaron/tess2speech/MainPanel.java

```
package anteraaron.tess2speech;

import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Image;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import javax.swing.BorderFactory;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JCheckBox;
import javax.swing.JComboBox;
import javax.swing.JFileChooser;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JSeparator;
import javax.swing.JTextField;
import javax.swing.SwingConstants;
import javax.swing.event.PopupMenuEvent;
import javax.swing.event.PopupMenuListener;
import javax.swing.filechooser.
    FileNameExtensionFilter;

/**
 * The Panel where the contents of the Frame
     is placed.
 * @author Anter Aaron M. Custodio
 * @since 2016/5/2
 */
public class MainPanel extends JPanel
    implements ActionListener{

  private static final long serialVersionUID =
      1L;
  private final JButton browse_button,
      start_button;
  private final JLabel tiff_label,
      settings_label, font_label,
      language_label;
  private final JSeparator separator;
  private final JFileChooser fileChooser;
  private final ImageIcon imageIcon;
  private final Image image;
  static JTextField tiff_textfield,
      font_textfield;
  static JComboBox<String> language_combobox;
  static JCheckBox checkBox;

  /**
   * Constructor
   */
  public MainPanel() {
    //Set background color
    super();

    //Absolute Positioning
    this.setLayout(null);

    //Input image label
    this.tiff_label = new JLabel("Input Image/
        s:");
    this.tiff_label.setBounds(10, 80, 130, 30)
        ;
    this.add(this.tiff_label);

    //Browsed Tiff File path container
    tiff_textfield = new JTextField();
    MainPanel.tiff_textfield.setBounds(60,
        115, 280, 30);
    MainPanel.tiff_textfield.setBorder(
        BorderFactory.createCompoundBorder(
        MainPanel.tiff_textfield.getBorder(),
         BorderFactory.createEmptyBorder(5,
        5, 5, 5)));
    this.add(MainPanel.tiff_textfield);

    //Browse Button
    this.browse_button = new JButton("Browse
        TIFF/s");
    this.browse_button.setToolTipText("Browse
        TIFF files. File paths of multiple
        TIFF images are separated by semi-
        colon");
    this.browse_button.setBounds(135, 160,
        120, 30);
    this.browse_button.setFocusable(false);
    this.add(this.browse_button);
    this.browse_button.addActionListener(this)
        ;

    //Line separator
    this.separator = new JSeparator(
        SwingConstants.HORIZONTAL);
    this.separator.setBounds(60, 205, 320, 5);
    this.add(this.separator);

    //Training settings label
    this.settings_label = new JLabel("Settings
        ");
    this.settings_label.setBounds(10, 190, 50,
         30);
    this.add(this.settings_label);

    //Font name label
    this.font_label = new JLabel("Trained Data
         Name: ");
    this.font_label.setBounds(30, 225, 120,
        30);
    this.add(this.font_label);

    //Name of font text field
    MainPanel.font_textfield = new JTextField
        ();
    MainPanel.font_textfield.setBounds(170,
        225, 130, 30);
    MainPanel.font_textfield.setBorder(
        BorderFactory.createCompoundBorder(
        MainPanel.font_textfield.getBorder(),
         BorderFactory.createEmptyBorder(5,
        5, 5, 5)));
    this.add(MainPanel.font_textfield);

    //Language label
    this.language_label = new JLabel("Box
        Maker:");
    this.language_label.setBounds(30, 275,
        280, 30);
    this.add(this.language_label);

    //Combo box for language lists
    MainPanel.language_combobox = new
        JComboBox<String>();
    addLanguages();
    MainPanel.language_combobox.setBounds(170,
         275, 100, 30);
    MainPanel.language_combobox.
        addPopupMenuListener( new
        PopupMenuListener() {

      @Override
      public void popupMenuWillBecomeVisible(
          PopupMenuEvent e) {
```

```
        //Refresh languages if there are new
            added
        addLanguages();
      }

      @Override
      public void popupMenuWillBecomeInvisible
          (PopupMenuEvent e) {
      }

      @Override
      public void popupMenuCanceled(
          PopupMenuEvent e) {
      }
    });
    this.add(MainPanel.language_combobox);

    //Language label
    checkBox = new JCheckBox("Overwrite
        previous Box Files");
    checkBox.setBounds(30, 325, 280, 30);
    checkBox.setFocusable(false);
    this.add(checkBox);

    //Start training button
    this.start_button = new JButton("Start
        Training");
    this.start_button.setToolTipText("Execute
        Tesseract Training");
    this.start_button.setBounds(135, 365, 120,
         30);
    this.start_button.setFocusable(false);
    this.add(this.start_button);
    this.start_button.addActionListener(this);

    //Create file Chooser
    this.fileChooser = new JFileChooser();
    this.fileChooser.setCurrentDirectory(new
        File("./"));

    //Add filters to file chooser
    this.fileChooser.addChoosableFileFilter(
        new FileNameExtensionFilter("Tag
        Image File Format (*.tiff, *.tif)", "
        tif", "tiff"));
    this.fileChooser.
        setAcceptAllFileFilterUsed(false);
    this.fileChooser.setMultiSelectionEnabled(
        true);

    this.imageIcon = new ImageIcon(getClass().
        getResource("/images/banner.png"));
    this.image = imageIcon.getImage();

  }

  /**
   * Method that populates the Combo box with
       .traineddata languages
   */
  static void addLanguages() {
    MainPanel.language_combobox.removeAllItems
        ();
    final File folder = new File("./tessdata")
        ;
    final File[] files = folder.listFiles();

    for(final File file : files){
      final int index = file.getName().
          lastIndexOf('.');
      if(index > 0) {
        final String extension = file.getName
            ().substring(index + 1);
        //Add all files with .traineddata
            extension to ComboBox
        if (extension.toLowerCase().equals("
            traineddata")) {
          MainPanel.language_combobox.addItem(
              file.getName().substring(0,
              index));
        }
      }
    }
  }

  /**
   * Draw the objects to the screen
   */
  @Override
  public void paintComponent(Graphics g){
    super.paintComponent(g);
    Graphics2D g2d = (Graphics2D) g;

    g2d.drawImage(this.image, 0, 0, 400, 85,
        null);
  }


  @Override
  public void actionPerformed(ActionEvent e) {
    //If browse button is clicked.
    if(e.getSource() == this.browse_button){
      final int returnVal = this.fileChooser.
          showOpenDialog(MainPanel.this);

      if (returnVal == JFileChooser.
          APPROVE_OPTION) {
        //Display browsed file paths
        String tiff_paths = "";
        for(final File file : this.fileChooser
            .getSelectedFiles()){
          tiff_paths += file.getAbsolutePath()
              + ";";
        }
        tiff_textfield.setText(tiff_paths);
      }

    }

    //If start training is clicked
    else if (e.getSource() == this.
        start_button){
      if (tiff_textfield.getText().replaceAll(
          ";", "").isEmpty() ||
          font_textfield.getText().isEmpty())
          {
        JOptionPane.showMessageDialog(null, "
            Empty Field/s!", "Failure",
            JOptionPane.ERROR_MESSAGE);
      } else {
        Functions.startTraining();
      }
    }

  }


}
```

These are the edited files in jTessBoxEditor by VietOCR [32]:

Path = `Tess2SpeechTrainer/src/jTessBoxEditor/src/net/sourceforge/tessboxeditor/Gui.java`

```
/**
 * Edited by Anter Aaron M. Custodio
 * Removed Toolbars, Training, Tiff Generator
     tabs.
 * Changed Open button to Next/Finish button
 * @since 2016/5/3
 *
 *
 * Copyright @ 2011 Quan Nguyen
 *
 * Licensed under the Apache License, Version
     2.0 (the "License"); you may not
 * use this file except in compliance with the
     License. You may obtain a copy of
 * the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed
     to in writing, software
 * distributed under the License is
     distributed on an "AS IS" BASIS, WITHOUT
```

```
 * WARRANTIES OR CONDITIONS OF ANY KIND,
      either express or implied. See the
 * License for the specific language governing
      permissions and limitations under
 * the License.
 */
package net.sourceforge.tessboxeditor;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Component;
import java.awt.Cursor;
import java.awt.DefaultKeyboardFocusManager;
import java.awt.Font;
import java.awt.Frame;
import java.awt.GraphicsEnvironment;
import java.awt.HeadlessException;
import java.awt.Image;
import java.awt.KeyEventDispatcher;
import java.awt.KeyboardFocusManager;
import java.awt.Point;
import java.awt.Rectangle;
import java.awt.dnd.DropTarget;
import java.awt.event.*;
import java.awt.image.BufferedImage;
import java.io.*;
import java.nio.channels.*;
import java.text.*;
import java.util.*;
import java.util.logging.Level;
import java.util.logging.Logger;
import java.util.prefs.Preferences;
import javax.swing.*;
import javax.swing.JSpinner.DefaultEditor;
import javax.swing.event.*;
import javax.swing.filechooser.FileFilter;
import javax.swing.table.*;
import net.sourceforge.tessboxeditor.
    components.*;
import net.sourceforge.tessboxeditor.datamodel
    .*;
import net.sourceforge.vietocr.util.Utils;
import net.sourceforge.tess4j.util.
    ImageIOHelper;
import net.sourceforge.vietpad.components.*;
import net.sourceforge.vietpad.utilities.
    LimitedLengthDocument;
import net.sourceforge.vietpad.utilities.
    TextUtilities;

public class Gui extends javax.swing.JFrame {

  private static final long serialVersionUID =
      1L;
  public static final String APP_NAME = "
      jTessBoxEditor";
 public static final String TO_BE_IMPLEMENTED
     = "To be implemented in subclass";
 final String[] headers = {"Char", "X", "Y", "
     Width", "Height", "Ok"}; //Edited by
     Aaron
 static final boolean MAC_OS_X = System.
     getProperty("os.name").startsWith("Mac");
 static final boolean WINDOWS = System.
     getProperty("os.name").toLowerCase().
     startsWith("windows");
 static final String EOL = System.getProperty(
     "line.separator");
 static final String UTF8 = "UTF-8";
 protected ResourceBundle bundle;
 static final Preferences prefs = Preferences.
     userRoot().node("/net/sourceforge/
     tessboxeditor");
 private final Rectangle screen =
     GraphicsEnvironment.
     getLocalGraphicsEnvironment().
     getMaximumWindowBounds();
 private int filterIndex;
 private FileFilter[] fileFilters;
 private File boxFile;
 private String currentDirectory,
     outputDirectory;
 private boolean boxChanged = false; //Edited
     by Aaron
 protected boolean tableSelectAction;
 private List<TessBoxCollection> boxPages;
 protected TessBoxCollection boxes; // boxes
     of current page
 protected short imageIndex;
 private List<BufferedImage> imageList;
 protected final File baseDir;
 DefaultTableModel tableModel;
 private boolean isTess2_0Format;
 protected RowHeaderList rowHeader;
 protected Font font;

 protected static int iconMargin = 3;
 protected static boolean invertControls =
     false;
 protected static int scaleFactor = 4;
 protected static int iconPosX = 0;
 protected static int iconPosY = 0;
 protected static int iconWidth = 0;
 protected static int iconHeight = 0;
 protected static int imageWidth = 0;
 protected static int imageHeight = 0;
 protected static int movementMultiplier = 1;

 private final static Logger logger = Logger.
     getLogger(Gui.class.getName());
 /*
     * Added by Anter Aaron M. Custodio
     * Saves arguments provided by the user
     */
 public static File[] inputTiffs;
 public static int currentIndex = 0;
 public static boolean finished = false;

 /**
     * Creates new form JTessBoxEditor.
     */
 public Gui() {

  this.baseDir = Utils.getBaseDir(Gui.this);
  try {
   UIManager.setLookAndFeel(prefs.get("
     lookAndFeel", UIManager.
     getSystemLookAndFeelClassName()));
  } catch (Exception e) {
   // keep default LAF
   logger.log(Level.WARNING, e.getMessage(), e
     );
  }
  bundle = ResourceBundle.getBundle("net.
     sourceforge.tessboxeditor.Gui"); //
     NOI18N
  initComponents();

  if (MAC_OS_X) {
   new MacOSXApplication(Gui.this);

   // remove Exit menuitem
   this.jMenuFile.remove(this.jSeparatorExit);
   this.jMenuFile.remove(this.jMenuItemExit);

   // remove About menuitem
   this.jMenuHelp.remove(this.jSeparatorAbout)
     ;
   this.jMenuHelp.remove(this.jMenuItemAbout);
  }

  boxPages = new ArrayList<TessBoxCollection
     >();

  // DnD support
  new DropTarget(this.jSplitPaneEditor, new
     FileDropTargetListener(Gui.this, this.
     jSplitPaneEditor));

  this.addWindowListener(
    new WindowAdapter() {

     @Override
     public void windowClosing(WindowEvent e)
     {
      quit();
     }

     @Override
     public void windowOpened(WindowEvent e) {
      updateSave(false);
      setExtendedState(prefs.getInt("
     windowState", Frame.NORMAL));
      populateMRUList();
     }
    });

  setSize(
    snap(prefs.getInt("frameWidth", 500), 300,
      screen.width),
    snap(prefs.getInt("frameHeight", 360),
     150, screen.height));
  setLocation(
    snap(
      prefs.getInt("frameX", (screen.width -
     getWidth()) / 2),
      screen.x, screen.x + screen.width -
     getWidth()),
    snap(
      prefs.getInt("frameY", screen.y + (
     screen.height - getHeight()) / 3),
      screen.y, screen.y + screen.height -
     getHeight()));

  KeyEventDispatcher dispatcher = new
     KeyEventDispatcher() {

   @Override
```

```
    public boolean dispatchKeyEvent(KeyEvent e)
        {
      if (e.getID() == KeyEvent.KEY_PRESSED) {
       if (e.getKeyCode() == KeyEvent.VK_F3) {
        jButtonFind.doClick();
       }
      }
      return false;
     }
    };
    DefaultKeyboardFocusManager.
       getCurrentKeyboardFocusManager().
       addKeyEventDispatcher(dispatcher);

    /*
            * Edited by Anter Aaron Custodio.
            * Change Texts of Button
            * Waits for UI to complete and then
                automatically open passed file
            */
     Thread thread = new Thread(new Runnable() {

        @Override
        public void run() {
          while(!finished) {}
          if (inputTiffs.length > 0) {
            openFile(inputTiffs[currentIndex]);
            currentIndex++;
          }

          if (currentIndex == inputTiffs.length)
            {
            //Train if last index is reached
            jButtonOpen.setText("Finish");
          } else {
            jButtonOpen.setText("Next");
          }

        }
      });
     thread.start();
    }

    private int snap(final int ideal, final int
       min, final int max) {
     final int TOLERANCE = 0;
     return ideal < min + TOLERANCE ? min : (
       ideal > max - TOLERANCE ? max : ideal);
    }

    /**
         * Populates MRU List.
         */
    protected void populateMRUList() {
     JOptionPane.showMessageDialog(this,
       TO_BE_IMPLEMENTED);
    }

    /**
         * This method is called from within the
              constructor to initialize the form.
         * WARNING: Do NOT modify this code. The
              content of this method is always
         * regenerated by the Form Editor.
         */
    @SuppressWarnings({"unchecked", "rawtypes"
        })
    // <editor-fold defaultstate="collapsed" desc
       ="Generated Code">//GEN-BEGIN:
       initComponents
    private void initComponents() {

     jFileChooserInputImage = new javax.swing.
       JFileChooser();
     jDialogValidationResult = new javax.swing.
       JDialog();
     jScrollPane2 = new javax.swing.JScrollPane()
       ;
     jTextAreaValidationResult = new javax.swing.
       JTextArea();
     jPanelCommand = new javax.swing.JPanel();
     jButtonCloseDialog = new javax.swing.JButton
       ();
     jTabbedPaneMain = new javax.swing.
       JTabbedPane();
     jPanelTrainer = new javax.swing.JPanel();
     jToolBarTrainer = new javax.swing.JToolBar()
       ;
     jPanelMain = new javax.swing.JPanel();
     jLabel4 = new javax.swing.JLabel();
     jTextFieldTessDir = new javax.swing.
       JTextField();
     jButtonBrowseTess = new javax.swing.JButton
       ();
     jLabel3 = new javax.swing.JLabel();
     jTextFieldDataDir = new javax.swing.
       JTextField();
     jButtonBrowseData = new javax.swing.JButton
       ();
     jLabel1 = new javax.swing.JLabel();
     jTextFieldLang = new javax.swing.JTextField
       ();
     jLabel2 = new javax.swing.JLabel();
     jTextFieldBootstrapLang = new javax.swing.
       JTextField();
     jCheckBoxRTL = new javax.swing.JCheckBox();
     jComboBoxOps = new javax.swing.JComboBox();
     jButtonTrain = new javax.swing.JButton();
     jButtonCancel = new javax.swing.JButton();
     jButtonCancel.setEnabled(false);
     jButtonValidate = new javax.swing.JButton();
     jButtonSaveLog = new javax.swing.JButton();
     jButtonClearLog = new javax.swing.JButton();
     jScrollPane1 = new javax.swing.JScrollPane()
       ;
     jTextAreaOutput = new javax.swing.JTextArea
       ();
     jPanelStatus1 = new javax.swing.JPanel();
     filler2 = new javax.swing.Box.Filler(new
       java.awt.Dimension(0, 17), new java.awt.
       Dimension(0, 17), new java.awt.Dimension
       (32767, 17));
     jProgressBar1 = new javax.swing.JProgressBar
       ();
     jProgressBar1.setVisible(false);
     jLabelTime = new javax.swing.JLabel();
     jPanelEditor = new javax.swing.JPanel();
     jToolBarEditor = new javax.swing.JToolBar();
     jPanel4 = new javax.swing.JPanel();
     jButtonOpen = new javax.swing.JButton();
     jButtonSave = new javax.swing.JButton();
     jButtonReload = new javax.swing.JButton();
     jButtonMerge = new javax.swing.JButton();
     jButtonSplit = new javax.swing.JButton();
     jButtonInsert = new javax.swing.JButton();
     jButtonDelete = new javax.swing.JButton();
     jPanelSpinner = new javax.swing.JPanel();
     jLabelCharacter = new javax.swing.JLabel();
     jTextFieldCharacter = new javax.swing.
       JTextField();
     jTextFieldCharacter.setDocument(new
       LimitedLengthDocument(12));
     jButtonConvert = new javax.swing.JButton();
     jLabelX = new javax.swing.JLabel();
     jSpinnerX = new javax.swing.JSpinner();
     jLabelY = new javax.swing.JLabel();
     jSpinnerY = new javax.swing.JSpinner();
     jLabelW = new javax.swing.JLabel();
     jSpinnerW = new javax.swing.JSpinner();
     jLabelH = new javax.swing.JLabel();
     jSpinnerH = new javax.swing.JSpinner();
     jSplitPaneEditor = new javax.swing.
       JSplitPane();
     jTabbedPaneBoxData = new javax.swing.
       JTabbedPane();
     jPanelCoord = new javax.swing.JPanel();
     jScrollPaneCoord = new javax.swing.
       JScrollPane();
     jTable = new javax.swing.JTable() {
         private static final long
             serialVersionUID = 1L;

         public Component prepareRenderer(
             TableCellRenderer renderer, int row
             , int column) {
       Component c = super.prepareRenderer(
        renderer, row, column);
       if (font != null && column == 0) {
        c.setFont(font);
       }

         /*
                 *Added by Anter Aaron Custodio
                     for changing color of
                    selected rows
                 */
       if (!isRowSelected(row) && jTable.getModel
        ().getValueAt(row, 5) != null && (boolean
        )jTable.getModel().getValueAt(row, 5) ==
        true) {
         c.setBackground(Color.GREEN);
       } else if (isRowSelected(row)){
         c.setBackground(Color.BLUE);
       } else {
         c.setBackground(Color.WHITE);
       }

       return c;
      }
     };
     jPanelFind = new javax.swing.JPanel();
     jTextFieldFind = new javax.swing.JTextField
       ();
     jButtonFind = new javax.swing.JButton();
     jScrollPaneBoxData = new javax.swing.
       JScrollPane();
     jTextAreaBoxData = new javax.swing.JTextArea
       ();
```

```
jPanelBoxView = new javax.swing.JPanel();
jPanelNorthContainer = new javax.swing.
    JPanel();
jPanelChar = new javax.swing.JPanel();
jLabelCodepoint = new javax.swing.JLabel();
jLabelCodepoint.setFont(jLabelCodepoint.
    getFont().deriveFont(14.0f));
jTextFieldChar = new javax.swing.JTextField
    ();
jTextFieldChar.setFont(jTextFieldChar.
    getFont().deriveFont(14.0f));
jTextFieldCodepointValue = new javax.swing.
    JTextField();
jTextFieldCodepointValue.setFont(
    jTextFieldCodepointValue.getFont().
    deriveFont(14.0f));
jPanelControls = new javax.swing.JPanel();
jLabelSpinnerMargin = new javax.swing.JLabel
    ();
jSpinnerMargin = new javax.swing.JSpinner();
jLabelSpinnerScale = new javax.swing.JLabel
    ();
jSpinnerScale = new javax.swing.JSpinner();
jLabelSubimage = new SubImageView();
jPanelButtons = new javax.swing.JPanel();
jButtonPrev = new javax.swing.JButton();
jButtonNext = new javax.swing.JButton();
jScrollPaneImage = new javax.swing.
    JScrollPane();
jScrollPaneImage.getVerticalScrollBar().
    setUnitIncrement(20);
jScrollPaneImage.getHorizontalScrollBar().
    setUnitIncrement(20);
jLabelImage = new JImageLabel();
jPanelStatus = new javax.swing.JPanel();
jLabelStatus = new javax.swing.JLabel();
jLabelPageNbr = new javax.swing.JLabel();
jButtonPrevPage = new javax.swing.JButton();
jButtonNextPage = new javax.swing.JButton();
jPanelTIFFBox = new javax.swing.JPanel();
jToolBarGenerator = new javax.swing.JToolBar
    ();
jPanel3 = new javax.swing.JPanel();
jButtonInput = new javax.swing.JButton();
jLabelOutput = new javax.swing.JLabel();
jTextFieldOuputDir = new javax.swing.
    JTextField();
jButtonBrowseOutputDir = new javax.swing.
    JButton();
jTextFieldPrefix = new javax.swing.
    JTextField();
jTextFieldPrefix.setText(prefs.get("
    trainLanguage", "eng"));
jTextFieldFileName = new javax.swing.
    JTextField();
jButtonFont = new javax.swing.JButton();
jCheckBoxAntiAliasing = new javax.swing.
    JCheckBox();
jLabelNoise = new javax.swing.JLabel();
jSpinnerNoise = new javax.swing.JSpinner();
jLabelTracking = new javax.swing.JLabel();
jSpinnerTracking = new javax.swing.JSpinner
    ();
jLabelW1 = new javax.swing.JLabel();
jSpinnerW1 = new javax.swing.JSpinner();
jLabelH1 = new javax.swing.JLabel();
jSpinnerH1 = new javax.swing.JSpinner();
jButtonGenerate = new javax.swing.JButton();
jButtonClear = new javax.swing.JButton();
jScrollPaneText = new javax.swing.
    JScrollPane();
jTextAreaInput = new javax.swing.JTextArea()
    ;
jMenuBar = new javax.swing.JMenuBar();
jMenuFile = new javax.swing.JMenu();
jMenuItemOpen = new javax.swing.JMenuItem();
jMenuItemSave = new javax.swing.JMenuItem();
jMenuItemSaveAs = new javax.swing.JMenuItem
    ();
jSeparatorRecentFiles = new javax.swing.
    JPopupMenu.Separator();
jMenuRecentFiles = new javax.swing.JMenu();
jSeparatorExit = new javax.swing.JPopupMenu.
    Separator();
jMenuItemExit = new javax.swing.JMenuItem();
jMenuEdit = new javax.swing.JMenu();
this.jMenuEdit.setVisible(false);
jMenuItemMerge = new javax.swing.JMenuItem()
    ;
jMenuItemSplit = new javax.swing.JMenuItem()
    ;
jMenuItemInsert = new javax.swing.JMenuItem
    ();
jMenuItemDelete = new javax.swing.JMenuItem
    ();
jMenuSettings = new javax.swing.JMenu();
jMenuItemFont = new javax.swing.JMenuItem();
jSeparatorLAF = new javax.swing.JPopupMenu.
    Separator();
jMenuLookAndFeel = new javax.swing.JMenu();
jMenuTools = new javax.swing.JMenu();
jMenuItemMergeTiff = new javax.swing.
    JMenuItem();
jMenuItemSplitTiff = new javax.swing.
    JMenuItem();
jMenuHelp = new javax.swing.JMenu();
jMenuItemHelp = new javax.swing.JMenuItem();
jSeparatorAbout = new javax.swing.JPopupMenu
    .Separator();
jMenuItemAbout = new javax.swing.JMenuItem()
    ;

java.util.ResourceBundle bundle = java.util.
    ResourceBundle.getBundle("net/sourceforge
    /tessboxeditor/Gui"); // NOI18N
jFileChooserInputImage.setDialogTitle(bundle
    .getString("jButtonOpen.ToolTipText"));
    // NOI18N

currentDirectory = prefs.get("
    currentDirectory", null);
outputDirectory = prefs.get("outputDirectory
    ", null);
jFileChooserInputImage.setCurrentDirectory(
    currentDirectory == null ? null : new
    File(currentDirectory));
filterIndex = prefs.getInt("filterIndex", 0)
    ;
FileFilter allImageFilter = new SimpleFilter
    ("bmp;jpg;jpeg;png;tif;tiff", bundle.
    getString("All_Image_Files"));
FileFilter pngFilter = new SimpleFilter("png
    ", "PNG");
FileFilter tiffFilter = new SimpleFilter("
    tif;tiff", "TIFF");
//FileFilter textFilter = new SimpleFilter("
    box;txt", "Box Files");

jFileChooserInputImage.
    setAcceptAllFileFilterUsed(false);
jFileChooserInputImage.
    addChoosableFileFilter(allImageFilter);
jFileChooserInputImage.
    addChoosableFileFilter(pngFilter);
jFileChooserInputImage.
    addChoosableFileFilter(tiffFilter);
//jFileChooser.addChoosableFileFilter(
    textFilter);
fileFilters = jFileChooserInputImage.
    getChoosableFileFilters();
if (filterIndex < fileFilters.length) {
 jFileChooserInputImage.setFileFilter(
    fileFilters[filterIndex]);
}

jDialogValidationResult.setTitle("Validation
     Result");
jDialogValidationResult.setMinimumSize(new
    java.awt.Dimension(600, 450));

jTextAreaValidationResult.setEditable(false)
    ;
jTextAreaValidationResult.setColumns(20);
jTextAreaValidationResult.setRows(5);
jTextAreaValidationResult.setMargin(new java
    .awt.Insets(5, 5, 2, 2));
jScrollPane2.setViewportView(
    jTextAreaValidationResult);

jDialogValidationResult.getContentPane().add
    (jScrollPane2, java.awt.BorderLayout.
    CENTER);

jPanelCommand.setBorder(javax.swing.
    BorderFactory.createEmptyBorder(5, 1, 5,
    1));

jButtonCloseDialog.setText("Close");
jButtonCloseDialog.setToolTipText("Close
    Dialog");
jButtonCloseDialog.addActionListener(new
    java.awt.event.ActionListener() {
  public void actionPerformed(java.awt.event.
    ActionEvent evt) {
   jButtonCloseDialogActionPerformed(evt);
  }
});
jPanelCommand.add(jButtonCloseDialog);

jDialogValidationResult.getContentPane().add
    (jPanelCommand, java.awt.BorderLayout.
    SOUTH);

jDialogValidationResult.
    setLocationRelativeTo(this);

setDefaultCloseOperation(javax.swing.
    WindowConstants.EXIT_ON_CLOSE);
```

```
setTitle("jTessBoxEditor");

jTabbedPaneMain.setBorder(javax.swing.
    BorderFactory.createEmptyBorder(-2, 0, 0,
     0));
jTabbedPaneMain.addChangeListener(new javax.
    swing.event.ChangeListener() {
  public void stateChanged(javax.swing.event.
    ChangeEvent evt) {
   jTabbedPaneMainStateChanged(evt);
  }
});

jPanelTrainer.setLayout(new java.awt.
    BorderLayout());

jToolBarTrainer.setRollover(true);

jPanelMain.setLayout(new java.awt.FlowLayout
    (java.awt.FlowLayout.LEFT));

jLabel4.setText("Tesseract Executables");
jPanelMain.add(jLabel4);

jTextFieldTessDir.setToolTipText("Location
    of Tesseract Executables");
jTextFieldTessDir.setEnabled(false);
jTextFieldTessDir.setPreferredSize(new java.
    awt.Dimension(180, 24));
jPanelMain.add(jTextFieldTessDir);

jButtonBrowseTess.setText("...");
jButtonBrowseTess.setToolTipText("Browse");
jButtonBrowseTess.setMaximumSize(new java.
    awt.Dimension(30, 23));
jButtonBrowseTess.setMinimumSize(new java.
    awt.Dimension(30, 23));
jButtonBrowseTess.setPreferredSize(new java.
    awt.Dimension(24, 23));
jButtonBrowseTess.addActionListener(new java
    .awt.event.ActionListener() {
  public void actionPerformed(java.awt.event.
    ActionEvent evt) {
   jButtonBrowseTessActionPerformed(evt);
  }
});
jPanelMain.add(jButtonBrowseTess);

jLabel3.setText("Training Data");
jPanelMain.add(jLabel3);

jTextFieldDataDir.setToolTipText("Location
    of Source Training Data");
jTextFieldDataDir.setEnabled(false);
jTextFieldDataDir.setPreferredSize(new java.
    awt.Dimension(180, 24));
jPanelMain.add(jTextFieldDataDir);

jButtonBrowseData.setText("...");
jButtonBrowseData.setToolTipText("Browse");
jButtonBrowseData.setMaximumSize(new java.
    awt.Dimension(30, 23));
jButtonBrowseData.setMinimumSize(new java.
    awt.Dimension(30, 23));
jButtonBrowseData.setPreferredSize(new java.
    awt.Dimension(24, 23));
jButtonBrowseData.addActionListener(new java
    .awt.event.ActionListener() {
  public void actionPerformed(java.awt.event.
    ActionEvent evt) {
   jButtonBrowseDataActionPerformed(evt);
  }
});
jPanelMain.add(jButtonBrowseData);

jLabel1.setText("Language");
jPanelMain.add(jLabel1);

jTextFieldLang.setMinimumSize(new java.awt.
    Dimension(34, 19));
jTextFieldLang.setPreferredSize(new java.awt
    .Dimension(30, 24));
jPanelMain.add(jTextFieldLang);

jLabel2.setText("Bootstrap Language");
jPanelMain.add(jLabel2);

jTextFieldBootstrapLang.setMinimumSize(new
    java.awt.Dimension(34, 19));
jTextFieldBootstrapLang.setPreferredSize(new
     java.awt.Dimension(30, 24));
jPanelMain.add(jTextFieldBootstrapLang);

jCheckBoxRTL.setText("RTL");
jCheckBoxRTL.setToolTipText("Right-To-Left
    Text Direction");
jPanelMain.add(jCheckBoxRTL);

jComboBoxOps.setModel(new
    DefaultComboBoxModel(TrainingMode.values
    ()));
jComboBoxOps.setToolTipText("Training Mode")
    ;
jPanelMain.add(jComboBoxOps);

jButtonTrain.setText("Run");
jButtonTrain.setToolTipText("Start Training"
    );
jButtonTrain.addActionListener(new java.awt.
    event.ActionListener() {
  public void actionPerformed(java.awt.event.
    ActionEvent evt) {
   jButtonTrainActionPerformed(evt);
  }
});
jPanelMain.add(jButtonTrain);

jButtonCancel.setText("Cancel");
jButtonCancel.setToolTipText("Cancel
    Training");
jButtonCancel.addActionListener(new java.awt
    .event.ActionListener() {
  public void actionPerformed(java.awt.event.
    ActionEvent evt) {
   jButtonCancelActionPerformed(evt);
  }
});
jPanelMain.add(jButtonCancel);

jButtonValidate.setText("Validate");
jButtonValidate.setToolTipText("Validate
    Generated Traineddata");
jButtonValidate.addActionListener(new java.
    awt.event.ActionListener() {
  public void actionPerformed(java.awt.event.
    ActionEvent evt) {
   jButtonValidateActionPerformed(evt);
  }
});
jPanelMain.add(jButtonValidate);

jButtonSaveLog.setText("Save");
jButtonSaveLog.setToolTipText("Save Log");
jButtonSaveLog.addActionListener(new java.
    awt.event.ActionListener() {
  public void actionPerformed(java.awt.event.
    ActionEvent evt) {
   jButtonSaveLogActionPerformed(evt);
  }
});
jPanelMain.add(jButtonSaveLog);

jButtonClearLog.setText("Clear");
jButtonClearLog.setToolTipText("Clear
    Textaera");
jButtonClearLog.addActionListener(new java.
    awt.event.ActionListener() {
  public void actionPerformed(java.awt.event.
    ActionEvent evt) {
   jButtonClearLogActionPerformed(evt);
  }
});
jPanelMain.add(jButtonClearLog);

jToolBarTrainer.add(jPanelMain);

jPanelTrainer.add(jToolBarTrainer, java.awt.
    BorderLayout.PAGE_START);

jTextAreaOutput.setEditable(false);
jTextAreaOutput.setColumns(20);
jTextAreaOutput.setRows(5);
jTextAreaOutput.setMargin(new java.awt.
    Insets(5, 5, 2, 2));
jScrollPane1.setViewportView(jTextAreaOutput
    );

jPanelTrainer.add(jScrollPane1, java.awt.
    BorderLayout.CENTER);

jPanelStatus1.setLayout(new java.awt.
    FlowLayout(java.awt.FlowLayout.LEFT));
jPanelStatus1.add(filler2);

jProgressBar1.setStringPainted(true);
jPanelStatus1.add(jProgressBar1);
jPanelStatus1.add(jLabelTime);

jPanelTrainer.add(jPanelStatus1, java.awt.
    BorderLayout.SOUTH);

//jTabbedPaneMain.addTab("Trainer",
    jPanelTrainer);

jPanelEditor.setLayout(new java.awt.
    BorderLayout());

jToolBarEditor.setRollover(true);
```

```
jPanel4.setLayout(new java.awt.FlowLayout(
    java.awt.FlowLayout.LEFT, 1, 5));

jButtonOpen.setText(bundle.getString("
    jButtonOpen.Text")); // NOI18N
jButtonOpen.setToolTipText(bundle.getString(
    "jButtonOpen.ToolTipText")); // NOI18N
jButtonOpen.setFocusable(false);
jButtonOpen.setHorizontalTextPosition(javax.
    swing.SwingConstants.CENTER);
jButtonOpen.setVerticalTextPosition(javax.
    swing.SwingConstants.BOTTOM);
jButtonOpen.addActionListener(new java.awt.
    event.ActionListener() {
  public void actionPerformed(java.awt.event.
    ActionEvent evt) {
   jButtonOpenActionPerformed(evt);
  }
});
jPanel4.add(jButtonOpen);

jButtonSave.setText(bundle.getString("
    jButtonSave.Text")); // NOI18N
jButtonSave.setToolTipText(bundle.getString(
    "jButtonSave.ToolTipText")); // NOI18N
jButtonSave.setFocusable(false);
jButtonSave.setHorizontalTextPosition(javax.
    swing.SwingConstants.CENTER);
jButtonSave.setVerticalTextPosition(javax.
    swing.SwingConstants.BOTTOM);
jButtonSave.addActionListener(new java.awt.
    event.ActionListener() {
  public void actionPerformed(java.awt.event.
    ActionEvent evt) {
   jButtonSaveActionPerformed(evt);
  }
});
jPanel4.add(jButtonSave);

jButtonReload.setText("Reload");
jButtonReload.setToolTipText("Reload Box
    File");
jButtonReload.setFocusable(false);
jButtonReload.setHorizontalTextPosition(
    javax.swing.SwingConstants.CENTER);
jButtonReload.setVerticalTextPosition(javax.
    swing.SwingConstants.BOTTOM);
jButtonReload.addActionListener(new java.awt
    .event.ActionListener() {
  public void actionPerformed(java.awt.event.
    ActionEvent evt) {
   jButtonReloadActionPerformed(evt);
  }
});
jPanel4.add(jButtonReload);
jPanel4.add(Box.createHorizontalStrut(100));

jButtonMerge.setText(bundle.getString("
    jButtonMerge.Text")); // NOI18N
jButtonMerge.setToolTipText(bundle.getString
    ("jButtonMerge.ToolTipText")); // NOI18N
jButtonMerge.setFocusable(false);
jButtonMerge.setHorizontalTextPosition(javax
    .swing.SwingConstants.CENTER);
jButtonMerge.setVerticalTextPosition(javax.
    swing.SwingConstants.BOTTOM);
jButtonMerge.addActionListener(new java.awt.
    event.ActionListener() {
  public void actionPerformed(java.awt.event.
    ActionEvent evt) {
   jButtonMergeActionPerformed(evt);
  }
});
jPanel4.add(jButtonMerge);

jButtonSplit.setText(bundle.getString("
    jButtonSplit.Text")); // NOI18N
jButtonSplit.setToolTipText(bundle.getString
    ("jButtonSplit.ToolTipText")); // NOI18N
jButtonSplit.setFocusable(false);
jButtonSplit.setHorizontalTextPosition(javax
    .swing.SwingConstants.CENTER);
jButtonSplit.setVerticalTextPosition(javax.
    swing.SwingConstants.BOTTOM);
jButtonSplit.addActionListener(new java.awt.
    event.ActionListener() {
  public void actionPerformed(java.awt.event.
    ActionEvent evt) {
   jButtonSplitActionPerformed(evt);
  }
});
jPanel4.add(jButtonSplit);

jButtonInsert.setText(bundle.getString("
    jButtonInsert.Text")); // NOI18N
jButtonInsert.setToolTipText(bundle.
    getString("jButtonInsert.ToolTipText"));
    // NOI18N
jButtonInsert.setFocusable(false);
jButtonInsert.setHorizontalTextPosition(
    javax.swing.SwingConstants.CENTER);
jButtonInsert.setVerticalTextPosition(javax.
    swing.SwingConstants.BOTTOM);
jButtonInsert.addActionListener(new java.awt
    .event.ActionListener() {
  public void actionPerformed(java.awt.event.
    ActionEvent evt) {
   jButtonInsertActionPerformed(evt);
  }
});
jPanel4.add(jButtonInsert);

jButtonDelete.setText(bundle.getString("
    jButtonDelete.Text")); // NOI18N
jButtonDelete.setToolTipText(bundle.
    getString("jButtonDelete.ToolTipText"));
    // NOI18N
jButtonDelete.setFocusable(false);
jButtonDelete.setHorizontalTextPosition(
    javax.swing.SwingConstants.CENTER);
jButtonDelete.setVerticalTextPosition(javax.
    swing.SwingConstants.BOTTOM);
jButtonDelete.addActionListener(new java.awt
    .event.ActionListener() {
  public void actionPerformed(java.awt.event.
    ActionEvent evt) {
   jButtonDeleteActionPerformed(evt);
  }
});
jPanel4.add(jButtonDelete);

jToolBarEditor.add(jPanel4);

jLabelCharacter.setLabelFor(
    jTextFieldCharacter);
jLabelCharacter.setText("Character");
jPanelSpinner.add(jLabelCharacter);

jTextFieldCharacter.setColumns(4);
jTextFieldCharacter.setEnabled(false);
jTextFieldCharacter.setMargin(new java.awt.
    Insets(0, 2, 0, 2));
jTextFieldCharacter.setPreferredSize(new
    java.awt.Dimension(40, 24));
jTextFieldCharacter.addActionListener(new
    java.awt.event.ActionListener() {
  public void actionPerformed(java.awt.event.
    ActionEvent evt) {
   jTextFieldCharacterActionPerformed(evt);
  }
});
jTextFieldCharacter.addKeyListener(new java.
    awt.event.KeyAdapter() {
  public void keyReleased(java.awt.event.
    KeyEvent evt) {
   jTextFieldCharacterKeyReleased(evt);
  }
});
jPanelSpinner.add(jTextFieldCharacter);

jButtonConvert.setIcon(new javax.swing.
    ImageIcon(getClass().getResource("/net/
    sourceforge/tessboxeditor/icons/tools.png
    ")));  // NOI18N
jButtonConvert.setToolTipText("<html>Convert
    NCR and Escape<br/>Sequence to Unicode</
    html>");
jButtonConvert.setPreferredSize(new java.awt
    .Dimension(20, 20));
jButtonConvert.addActionListener(new java.
    awt.event.ActionListener() {
  public void actionPerformed(java.awt.event.
    ActionEvent evt) {
   jButtonConvertActionPerformed(evt);
  }
});
jPanelSpinner.add(jButtonConvert);
jPanelSpinner.add(Box.createHorizontalStrut
    (10));

jLabelX.setLabelFor(jSpinnerX);
jLabelX.setText("X");
jPanelSpinner.add(jLabelX);

jSpinnerX.setEditor(new javax.swing.JSpinner
    .NumberEditor(jSpinnerX, "#"));
jSpinnerX.setEnabled(false);
jSpinnerX.setPreferredSize(new java.awt.
    Dimension(63, 22));
jSpinnerX.addChangeListener(new javax.swing.
    event.ChangeListener() {
  public void stateChanged(javax.swing.event.
    ChangeEvent evt) {
   jSpinnerXStateChanged(evt);
  }
});
jPanelSpinner.add(jSpinnerX);
```

```
jLabelY.setLabelFor(jSpinnerY);
jLabelY.setText("Y");
jPanelSpinner.add(jLabelY);

jSpinnerY.setEditor(new javax.swing.JSpinner
    .NumberEditor(jSpinnerY, "#"));
jSpinnerY.setEnabled(false);
jSpinnerY.setPreferredSize(new java.awt.
    Dimension(63, 22));
jSpinnerY.addChangeListener(new javax.swing.
    event.ChangeListener() {
  public void stateChanged(javax.swing.event.
    ChangeEvent evt) {
   jSpinnerYStateChanged(evt);
  }
});
jPanelSpinner.add(jSpinnerY);

jLabelW.setLabelFor(jSpinnerW);
jLabelW.setText("W");
jPanelSpinner.add(jLabelW);

jSpinnerW.setModel(new javax.swing.
    SpinnerNumberModel());
jSpinnerW.setEditor(new javax.swing.JSpinner
    .NumberEditor(jSpinnerW, "#"));
jSpinnerW.setEnabled(false);
jSpinnerW.setPreferredSize(new java.awt.
    Dimension(48, 22));
jSpinnerW.addChangeListener(new javax.swing.
    event.ChangeListener() {
  public void stateChanged(javax.swing.event.
    ChangeEvent evt) {
   jSpinnerWStateChanged(evt);
  }
});
jPanelSpinner.add(jSpinnerW);

jLabelH.setLabelFor(jSpinnerH);
jLabelH.setText("H");
jPanelSpinner.add(jLabelH);

jSpinnerH.setModel(new javax.swing.
    SpinnerNumberModel());
jSpinnerH.setEditor(new javax.swing.JSpinner
    .NumberEditor(jSpinnerH, "#"));
jSpinnerH.setEnabled(false);
jSpinnerH.setPreferredSize(new java.awt.
    Dimension(48, 22));
jSpinnerH.addChangeListener(new javax.swing.
    event.ChangeListener() {
  public void stateChanged(javax.swing.event.
    ChangeEvent evt) {
   jSpinnerHStateChanged(evt);
  }
});
jPanelSpinner.add(jSpinnerH);

jToolBarEditor.add(jPanelSpinner);
jToolBarEditor.add(Box.createHorizontalGlue
    ());

jPanelEditor.add(jToolBarEditor, java.awt.
    BorderLayout.PAGE_START);

jSplitPaneEditor.setDividerSize(2);

jPanelCoord.setLayout(new java.awt.
    BorderLayout());

jScrollPaneCoord.setPreferredSize(new java.
    awt.Dimension(200, 275));

jTable.setModel(new javax.swing.table.
    DefaultTableModel(
  new Object [][] {

  },
  new String [] {
   "Char", "X", "Y", "Width", "Height", "Ok"
  }
) {
     private static final long
          serialVersionUID = 1L;
     Class[] types = new Class [] {
   java.lang.String.class, java.lang.Object.
    class, java.lang.Object.class, java.lang.
    Object.class, java.lang.Object.class,
    java.lang.Boolean.class
  };
  boolean[] canEdit = new boolean [] {
   true, false, false, false, false, false
  };

  public Class getColumnClass(int columnIndex
    ) {
   return types [columnIndex];
  }
  public boolean isCellEditable(int rowIndex,
     int columnIndex) {
   return canEdit [columnIndex];
  }
});

jTable.setSelectionMode(javax.swing.
    ListSelectionModel.
    MULTIPLE_INTERVAL_SELECTION);
jScrollPaneCoord.setViewportView(jTable);
tableModel = (DefaultTableModel) this.jTable
    .getModel();
tableModel.addTableModelListener(new
    TableModelListener() {
  public void tableChanged(TableModelEvent e)
     {
   int row = e.getFirstRow();
   int column = e.getColumn();
   // update only if change to column 0 (Char
   )
   if (row != -1 && column == 0) {
    TableModel model = (TableModel)e.
   getSource();
    Object data = model.getValueAt(row,
   column);
    String value = (String) data;
    TessBox box = boxes.toList().get(row);
    box.setChrs(value);
    jTextFieldCharacter.setText(value);
    jTextFieldChar.setText(value);
    jTextFieldCodepointValue.setText(net.
   sourceforge.vietocr.util.Utils.toHex(
   value));
    updateSave(true);
   }
  }
});
ListSelectionModel cellSelectionModel =
    jTable.getSelectionModel();
cellSelectionModel.addListSelectionListener(
    new ListSelectionListener() {
  public void valueChanged(ListSelectionEvent
     e) {

   if (!e.getValueIsAdjusting()) {
    int selectedIndex = jTable.getSelectedRow
   ();
    if (selectedIndex != -1) {
     if (!((JImageLabel) jLabelImage).
   isBoxClickAction()) { // not from image
   block click
      boxes.deselectAll();
     }
     List<TessBox> boxesOfCurPage = boxes.
   toList(); // boxes of current page
     for (int index : jTable.getSelectedRows
   ()) {
      TessBox box = boxesOfCurPage.get(index)
   ;
      // select box
      box.setSelected(true);
      jLabelImage.scrollRectToVisible(box.
   getRect());
     }

     /*
                     * Code added by Anter
                         Aaron Custodio.
                         Highlight selected
                          rows
                    */
     jTable.getModel().setValueAt(true,
   jTable.getSelectedRow(), 5);

     jLabelImage.repaint();
     if (jTable.getSelectedRows().length ==
   1) {
      enableReadout(true);
      // update Character field
      jTextFieldCharacter.setText((String)
   tableModel.getValueAt(selectedIndex, 0));
      jTextFieldChar.setText(
   jTextFieldCharacter.getText());
      jTextFieldCodepointValue.setText(net.
   sourceforge.vietocr.util.Utils.toHex(
   jTextFieldCharacter.getText()));
      // update subimage label
      Icon icon = jLabelImage.getIcon();
      TessBox curBox = boxesOfCurPage.get(
   selectedIndex);
      Rectangle rect = curBox.getRect();
      try {
       Image subImage = getSubimage((
   BufferedImage) ((ImageIcon) icon).
   getImage(), rect);
       ImageIconScalable subIcon = new
   ImageIconScalable(subImage);
       subIcon.setScaledFactor(scaleFactor);
```

```
        jLabelSubimage.setIcon(subIcon);
       } catch (Exception exc) {
        //ignore
       }
       // mark this as table action event to
    prevent cyclic firing of events by
    spinners
       tableSelectAction = true;
       // update spinners
       jSpinnerX.setValue(rect.x);
       jSpinnerY.setValue(rect.y);
       jSpinnerH.setValue(rect.height);
       jSpinnerW.setValue(rect.width);
       tableSelectAction = false;
      } else {
       enableReadout(false);
       resetReadout();
      }
     } else {
      boxes.deselectAll();
      jLabelImage.repaint();
      enableReadout(false);
      tableSelectAction = true;
      resetReadout();
      tableSelectAction = false;

     }



   }
  }
});

TableCellRenderer tcr = this.jTable.
   getDefaultRenderer(String.class);
DefaultTableCellRenderer dtcr = (
   DefaultTableCellRenderer) tcr;
dtcr.setHorizontalAlignment(
   DefaultTableCellRenderer.CENTER);
((JLabel) jTable.getTableHeader().
   getDefaultRenderer()).
   setHorizontalAlignment(SwingConstants.
   CENTER);
((JImageLabel) this.jLabelImage).setTable(
   jTable);
jTable.getInputMap(JComponent.
   WHEN_ANCESTOR_OF_FOCUSED_COMPONENT).put(
   KeyStroke.getKeyStroke("control C"), "
   none");
jTable.getInputMap(JComponent.
   WHEN_ANCESTOR_OF_FOCUSED_COMPONENT).put(
   KeyStroke.getKeyStroke("control X"), "
   none");
jTable.getInputMap(JComponent.
   WHEN_ANCESTOR_OF_FOCUSED_COMPONENT).put(
   KeyStroke.getKeyStroke("control V"), "
   none");
rowHeader = new RowHeaderList(this.jTable);
this.jScrollPaneCoord.setRowHeaderView(
   rowHeader);
this.jTable.putClientProperty("JTable.
   autoStartsEdit", Boolean.FALSE);
this.jTable.putClientProperty("
   terminateEditOnFocusLost", Boolean.TRUE);
this.jTable.setDefaultEditor(String.class,
   new MyTableCellEditor());

jPanelCoord.add(jScrollPaneCoord, java.awt.
   BorderLayout.CENTER);

jTextFieldFind.setPreferredSize(new java.awt
   .Dimension(200, 20));
jTextFieldFind.addActionListener(new java.
   awt.event.ActionListener() {
 public void actionPerformed(java.awt.event.
   ActionEvent evt) {
  jTextFieldFindActionPerformed(evt);
 }
});
jPanelFind.add(jTextFieldFind);

jButtonFind.setText(bundle.getString("
   jButtonFind.Text")); // NOI18N
jButtonFind.setToolTipText(bundle.getString(
   "jButtonFind.ToolTipText")); // NOI18N
jButtonFind.addActionListener(new java.awt.
   event.ActionListener() {
 public void actionPerformed(java.awt.event.
   ActionEvent evt) {
  jButtonFindActionPerformed(evt);
 }
});
jPanelFind.add(jButtonFind);

jPanelCoord.add(jPanelFind, java.awt.
   BorderLayout.SOUTH);

jTabbedPaneBoxData.addTab("Box Coordinates",
    jPanelCoord);

jTextAreaBoxData.setEditable(false);
jTextAreaBoxData.setColumns(20);
jTextAreaBoxData.setRows(5);
jTextAreaBoxData.setMargin(new java.awt.
   Insets(8, 8, 2, 2));
jScrollPaneBoxData.setViewportView(
   jTextAreaBoxData);

jTabbedPaneBoxData.addTab("Box Data",
   jScrollPaneBoxData);

jPanelBoxView.setBackground(java.awt.Color.
   lightGray);
jPanelBoxView.setLayout(new java.awt.
   BorderLayout());

jPanelNorthContainer.setLayout(new javax.
   swing.BoxLayout(jPanelNorthContainer,
   javax.swing.BoxLayout.Y_AXIS));

jPanelChar.setBackground(java.awt.Color.
   lightGray);
jPanelChar.setLayout(new java.awt.FlowLayout
   (java.awt.FlowLayout.LEFT));

jLabelCodepoint.setText("Char/Codepoint:");
jPanelChar.add(jLabelCodepoint);

jTextFieldChar.setEditable(false);
jTextFieldChar.setOpaque(false);
jPanelChar.add(jTextFieldChar);

jTextFieldCodepointValue.setEditable(false);
jTextFieldCodepointValue.setOpaque(false);
jPanelChar.add(jTextFieldCodepointValue);

jPanelNorthContainer.add(jPanelChar);

jPanelControls.setBackground(java.awt.Color.
   lightGray);
jPanelControls.setLayout(new java.awt.
   FlowLayout(java.awt.FlowLayout.LEFT));

jLabelSpinnerMargin.setLabelFor(
   jSpinnerMargin);
jLabelSpinnerMargin.setText("Margins");
jPanelControls.add(jLabelSpinnerMargin);

jSpinnerMargin.setModel(new javax.swing.
   SpinnerNumberModel(3, 0, 20, 1));
jSpinnerMargin.setFocusable(false);
((DefaultEditor) jSpinnerMargin.getEditor())
   .getTextField().setEditable(false);
((DefaultEditor) jSpinnerMargin.getEditor())
   .getTextField().setFocusable(false);
jSpinnerMargin.addChangeListener(new javax.
   swing.event.ChangeListener() {
 public void stateChanged(javax.swing.event.
   ChangeEvent evt) {
  jSpinnerMarginStateChanged(evt);
 }
});
jPanelControls.add(jSpinnerMargin);

jLabelSpinnerScale.setLabelFor(jSpinnerScale
   );
jLabelSpinnerScale.setText("Scale");
jPanelControls.add(Box.createHorizontalStrut
   (10));
jPanelControls.add(jLabelSpinnerScale);

jSpinnerScale.setModel(new javax.swing.
   SpinnerNumberModel(4, 1, 10, 1));
jSpinnerScale.setFocusable(false);
((DefaultEditor) jSpinnerScale.getEditor()).
   getTextField().setEditable(false);
((DefaultEditor) jSpinnerScale.getEditor()).
   getTextField().setFocusable(false);
jSpinnerScale.addChangeListener(new javax.
   swing.event.ChangeListener() {
 public void stateChanged(javax.swing.event.
   ChangeEvent evt) {
  jSpinnerScaleStateChanged(evt);
 }
});
jPanelControls.add(jSpinnerScale);

jPanelNorthContainer.add(jPanelControls);

jPanelBoxView.add(jPanelNorthContainer, java
   .awt.BorderLayout.NORTH);

jLabelSubimage.setHorizontalAlignment(javax.
   swing.SwingConstants.CENTER);
jPanelBoxView.add(jLabelSubimage, java.awt.
   BorderLayout.CENTER);
```

```
jPanelButtons.setBackground(new java.awt.
    Color(192, 192, 192));

jButtonPrev.setText("Prev");
jButtonPrev.addActionListener(new java.awt.
    event.ActionListener() {
  public void actionPerformed(java.awt.event.
    ActionEvent evt) {
   jButtonPrevActionPerformed(evt);
  }
});
jPanelButtons.add(jButtonPrev);

jButtonNext.setText("Next");
jButtonNext.addActionListener(new java.awt.
    event.ActionListener() {
  public void actionPerformed(java.awt.event.
    ActionEvent evt) {
   jButtonNextActionPerformed(evt);
  }
});
jPanelButtons.add(jButtonNext);

jPanelBoxView.add(jPanelButtons, java.awt.
    BorderLayout.SOUTH);

jTabbedPaneBoxData.addTab("Box View",
    jPanelBoxView);

jSplitPaneEditor.setLeftComponent(
    jTabbedPaneBoxData);

jLabelImage.setVerticalAlignment(javax.swing
    .SwingConstants.TOP);
jScrollPaneImage.setViewportView(jLabelImage
    );

jSplitPaneEditor.setRightComponent(
    jScrollPaneImage);

jPanelEditor.add(jSplitPaneEditor, java.awt.
    BorderLayout.CENTER);

jPanelStatus.add(jLabelStatus);
jPanelStatus.add(jLabelPageNbr);
this.jPanelStatus.add(Box.
    createHorizontalStrut(10));

jButtonPrevPage.setIcon(new javax.swing.
    ImageIcon(getClass().getResource("/net/
    sourceforge/tessboxeditor/icons/PrevPage.
    gif"))); // NOI18N
jButtonPrevPage.setToolTipText(bundle.
    getString("jButtonPrevPage.ToolTipText"))
    ; // NOI18N
jButtonPrevPage.setFocusable(false);
jButtonPrevPage.setHorizontalTextPosition(
    javax.swing.SwingConstants.CENTER);
jButtonPrevPage.setVerticalTextPosition(
    javax.swing.SwingConstants.BOTTOM);
jButtonPrevPage.addActionListener(new java.
    awt.event.ActionListener() {
  public void actionPerformed(java.awt.event.
    ActionEvent evt) {
   jButtonPrevPageActionPerformed(evt);
  }
});
jPanelStatus.add(jButtonPrevPage);

jButtonNextPage.setIcon(new javax.swing.
    ImageIcon(getClass().getResource("/net/
    sourceforge/tessboxeditor/icons/NextPage.
    gif"))); // NOI18N
jButtonNextPage.setToolTipText(bundle.
    getString("jButtonNextPage.ToolTipText"))
    ; // NOI18N
jButtonNextPage.setFocusable(false);
jButtonNextPage.setHorizontalTextPosition(
    javax.swing.SwingConstants.CENTER);
jButtonNextPage.setVerticalTextPosition(
    javax.swing.SwingConstants.BOTTOM);
jButtonNextPage.addActionListener(new java.
    awt.event.ActionListener() {
  public void actionPerformed(java.awt.event.
    ActionEvent evt) {
   jButtonNextPageActionPerformed(evt);
  }
});
jPanelStatus.add(jButtonNextPage);

jPanelEditor.add(jPanelStatus, java.awt.
    BorderLayout.SOUTH);

jTabbedPaneMain.addTab("Box Editor",
    jPanelEditor);

jPanelTIFFBox.setLayout(new java.awt.
    BorderLayout());

jToolBarGenerator.setRollover(true);

jPanel3.setLayout(new java.awt.FlowLayout(
    java.awt.FlowLayout.LEFT));

jButtonInput.setText("Input");
jButtonInput.setToolTipText("Load Text File"
    );
jButtonInput.setFocusable(false);
jButtonInput.setHorizontalTextPosition(javax
    .swing.SwingConstants.CENTER);
jButtonInput.setVerticalTextPosition(javax.
    swing.SwingConstants.BOTTOM);
jButtonInput.addActionListener(new java.awt.
    event.ActionListener() {
  public void actionPerformed(java.awt.event.
    ActionEvent evt) {
   jButtonInputActionPerformed(evt);
  }
});
jPanel3.add(jButtonInput);

jLabelOutput.setText("Output");
jPanel3.add(jLabelOutput);

jTextFieldOuputDir.setToolTipText("Ouput
    Directory");
jTextFieldOuputDir.setEnabled(false);
jTextFieldOuputDir.setPreferredSize(new java
    .awt.Dimension(180, 24));
jPanel3.add(jTextFieldOuputDir);

jButtonBrowseOutputDir.setText("...");
jButtonBrowseOutputDir.setToolTipText("
    Browse");
jButtonBrowseOutputDir.setPreferredSize(new
    java.awt.Dimension(24, 23));
jButtonBrowseOutputDir.addActionListener(new
     java.awt.event.ActionListener() {
  public void actionPerformed(java.awt.event.
    ActionEvent evt) {
   jButtonBrowseOutputDirActionPerformed(evt)
    ;
  }
});
jPanel3.add(jButtonBrowseOutputDir);

jTextFieldPrefix.setToolTipText("Prefix (
    Language Code)");
jTextFieldPrefix.setPreferredSize(new java.
    awt.Dimension(38, 24));
jPanel3.add(jTextFieldPrefix);

jTextFieldFileName.setToolTipText("Filename"
    );
jTextFieldFileName.setPreferredSize(new java
    .awt.Dimension(140, 24));
jPanel3.add(jTextFieldFileName);

jButtonFont.setText("Font");
jButtonFont.setToolTipText("Select Font");
jButtonFont.setFocusable(false);
jButtonFont.setHorizontalTextPosition(javax.
    swing.SwingConstants.CENTER);
jButtonFont.setVerticalTextPosition(javax.
    swing.SwingConstants.BOTTOM);
jButtonFont.addActionListener(new java.awt.
    event.ActionListener() {
  public void actionPerformed(java.awt.event.
    ActionEvent evt) {
   jButtonFontActionPerformed(evt);
  }
});
jPanel3.add(jButtonFont);

jCheckBoxAntiAliasing.setText("Anti-Aliasing
    ");
jPanel3.add(jCheckBoxAntiAliasing);

jLabelNoise.setText("Noise");
jPanel3.add(jLabelNoise);

jSpinnerNoise.setModel(new javax.swing.
    SpinnerNumberModel(0, 0, 5, 1));
jSpinnerNoise.setToolTipText("Add Noise to
    Image");
jSpinnerNoise.setName("Noise"); // NOI18N
jSpinnerNoise.setPreferredSize(new java.awt.
    Dimension(47, 22));
jPanel3.add(jSpinnerNoise);

jLabelTracking.setText("Letter Tracking");
jPanel3.add(jLabelTracking);

jSpinnerTracking.setModel(new javax.swing.
    SpinnerNumberModel(Float.valueOf(0.0f),
    Float.valueOf(-0.04f), Float.valueOf(0.1f
    ), Float.valueOf(0.01f)));
```

```
jSpinnerTracking.setToolTipText("Adjust
    Letter Tracking");
jSpinnerTracking.setPreferredSize(new java.
    awt.Dimension(64, 22));
jSpinnerTracking.addChangeListener(new javax
    .swing.event.ChangeListener() {
  public void stateChanged(javax.swing.event.
    ChangeEvent evt) {
   jSpinnerTrackingStateChanged(evt);
  }
});
jPanel3.add(jSpinnerTracking);

jLabelW1.setText("W");
jPanel3.add(jLabelW1);

jSpinnerW1.setModel(new javax.swing.
    SpinnerNumberModel(2550, 600, 2550, 10));
jSpinnerW1.setToolTipText("Image Width");
jSpinnerW1.setEditor(new javax.swing.
    JSpinner.NumberEditor(jSpinnerW1, "#"));
jSpinnerW1.setPreferredSize(new java.awt.
    Dimension(63, 22));
jPanel3.add(jSpinnerW1);

jLabelH1.setText("H");
jPanel3.add(jLabelH1);

jSpinnerH1.setModel(new javax.swing.
    SpinnerNumberModel(3300, 400, 3300, 10));
jSpinnerH1.setToolTipText("Image Height");
jSpinnerH1.setEditor(new javax.swing.
    JSpinner.NumberEditor(jSpinnerH1, "#"));
jSpinnerH1.setPreferredSize(new java.awt.
    Dimension(63, 22));
jPanel3.add(jSpinnerH1);

jButtonGenerate.setText("Generate");
jButtonGenerate.setToolTipText("Generate
    TIFF/Box");
jButtonGenerate.setFocusable(false);
jButtonGenerate.setHorizontalTextPosition(
    javax.swing.SwingConstants.CENTER);
jButtonGenerate.setVerticalTextPosition(
    javax.swing.SwingConstants.BOTTOM);
jButtonGenerate.addActionListener(new java.
    awt.event.ActionListener() {
  public void actionPerformed(java.awt.event.
    ActionEvent evt) {
   jButtonGenerateActionPerformed(evt);
  }
});
jPanel3.add(jButtonGenerate);

jButtonClear.setText("Clear");
jButtonClear.setToolTipText("Clear Textarea"
    );
jButtonClear.setFocusable(false);
jButtonClear.setHorizontalTextPosition(javax
    .swing.SwingConstants.CENTER);
jButtonClear.setVerticalTextPosition(javax.
    swing.SwingConstants.BOTTOM);
jButtonClear.addActionListener(new java.awt.
    event.ActionListener() {
  public void actionPerformed(java.awt.event.
    ActionEvent evt) {
   jButtonClearActionPerformed(evt);
  }
});
jPanel3.add(jButtonClear);

jToolBarGenerator.add(jPanel3);

jPanelTIFFBox.add(jToolBarGenerator, java.
    awt.BorderLayout.PAGE_START);

jTextAreaInput.setColumns(20);
jTextAreaInput.setLineWrap(true);
jTextAreaInput.setRows(5);
jTextAreaInput.setWrapStyleWord(true);
jTextAreaInput.setMargin(new java.awt.Insets
    (5, 5, 2, 2));
jScrollPaneText.setViewportView(
    jTextAreaInput);

jPanelTIFFBox.add(jScrollPaneText, java.awt.
    BorderLayout.CENTER);

//jTabbedPaneMain.addTab("TIFF/Box Generator
    ", jPanelTIFFBox);

getContentPane().add(jTabbedPaneMain, java.
    awt.BorderLayout.CENTER);

jMenuFile.setMnemonic(java.util.
    ResourceBundle.getBundle("net/sourceforge
    /tessboxeditor/Gui").getString("jMenuFile
    .Mnemonic").charAt(0));
jMenuFile.setText(bundle.getString("
    jMenuFile.Text")); // NOI18N

jMenuItemOpen.setAccelerator(javax.swing.
    KeyStroke.getKeyStroke(java.awt.event.
    KeyEvent.VK_O, java.awt.event.InputEvent.
    CTRL_MASK));
jMenuItemOpen.setMnemonic(java.util.
    ResourceBundle.getBundle("net/sourceforge
    /tessboxeditor/Gui").getString("
    jMenuItemOpen.Mnemonic").charAt(0));
jMenuItemOpen.setText(bundle.getString("
    jMenuItemOpen.Text")); // NOI18N
jMenuItemOpen.addActionListener(new java.awt
    .event.ActionListener() {
  public void actionPerformed(java.awt.event.
    ActionEvent evt) {
   jMenuItemOpenActionPerformed(evt);
  }
});
jMenuFile.add(jMenuItemOpen);

jMenuItemSave.setAccelerator(javax.swing.
    KeyStroke.getKeyStroke(java.awt.event.
    KeyEvent.VK_S, java.awt.event.InputEvent.
    CTRL_MASK));
jMenuItemSave.setMnemonic(java.util.
    ResourceBundle.getBundle("net/sourceforge
    /tessboxeditor/Gui").getString("
    jMenuItemSave.Mnemonic").charAt(0));
jMenuItemSave.setText(bundle.getString("
    jMenuItemSave.Text")); // NOI18N
jMenuItemSave.addActionListener(new java.awt
    .event.ActionListener() {
  public void actionPerformed(java.awt.event.
    ActionEvent evt) {
   jMenuItemSaveActionPerformed(evt);
  }
});
jMenuFile.add(jMenuItemSave);

jMenuItemSaveAs.setAccelerator(javax.swing.
    KeyStroke.getKeyStroke(java.awt.event.
    KeyEvent.VK_S, java.awt.event.InputEvent.
    SHIFT_MASK | java.awt.event.InputEvent.
    CTRL_MASK));
jMenuItemSaveAs.setMnemonic(java.util.
    ResourceBundle.getBundle("net/sourceforge
    /tessboxeditor/Gui").getString("
    jMenuItemSaveAs.Mnemonic").charAt(0));
jMenuItemSaveAs.setText(bundle.getString("
    jMenuItemSaveAs.Text")); // NOI18N
jMenuItemSaveAs.addActionListener(new java.
    awt.event.ActionListener() {
  public void actionPerformed(java.awt.event.
    ActionEvent evt) {
   jMenuItemSaveAsActionPerformed(evt);
  }
});
jMenuFile.add(jMenuItemSaveAs);
jMenuFile.add(jSeparatorRecentFiles);

jMenuRecentFiles.setMnemonic(java.util.
    ResourceBundle.getBundle("net/sourceforge
    /tessboxeditor/Gui").getString("
    jMenuRecentFiles.Mnemonic").charAt(0));
jMenuRecentFiles.setText(bundle.getString("
    jMenuRecentFiles.Text")); // NOI18N
jMenuFile.add(jMenuRecentFiles);
jMenuFile.add(jSeparatorExit);

jMenuItemExit.setMnemonic(java.util.
    ResourceBundle.getBundle("net/sourceforge
    /tessboxeditor/Gui").getString("
    jMenuItemExit.Mnemonic").charAt(0));
jMenuItemExit.setText(bundle.getString("
    jMenuItemExit.Text")); // NOI18N
jMenuItemExit.addActionListener(new java.awt
    .event.ActionListener() {
  public void actionPerformed(java.awt.event.
    ActionEvent evt) {
   jMenuItemExitActionPerformed(evt);
  }
});
jMenuFile.add(jMenuItemExit);

jMenuBar.add(jMenuFile);

jMenuEdit.setMnemonic(java.util.
    ResourceBundle.getBundle("net/sourceforge
    /tessboxeditor/Gui").getString("jMenuEdit
    .Mnemonic").charAt(0));
jMenuEdit.setText(bundle.getString("
    jMenuEdit.Text")); // NOI18N

jMenuItemMerge.setAccelerator(javax.swing.
    KeyStroke.getKeyStroke(java.awt.event.
    KeyEvent.VK_C, java.awt.event.InputEvent.
    CTRL_MASK));
jMenuItemMerge.setMnemonic(java.util.
```

```
    ResourceBundle.getBundle("net/sourceforge
    /tessboxeditor/Gui").getString("
    jMenuItemMerge.Mnemonic").charAt(0));
jMenuItemMerge.setText(bundle.getString("
    jMenuItemMerge.Text")); // NOI18N
jMenuItemMerge.addActionListener(new java.
    awt.event.ActionListener() {
  public void actionPerformed(java.awt.event.
    ActionEvent evt) {
   jMenuItemMergeActionPerformed(evt);
  }
});
jMenuEdit.add(jMenuItemMerge);

jMenuItemSplit.setAccelerator(javax.swing.
    KeyStroke.getKeyStroke(java.awt.event.
    KeyEvent.VK_X, java.awt.event.InputEvent.
    CTRL_MASK));
jMenuItemSplit.setMnemonic(java.util.
    ResourceBundle.getBundle("net/sourceforge
    /tessboxeditor/Gui").getString("
    jMenuItemSplit.Mnemonic").charAt(0));
jMenuItemSplit.setText(bundle.getString("
    jMenuItemSplit.Text")); // NOI18N
jMenuItemSplit.addActionListener(new java.
    awt.event.ActionListener() {
  public void actionPerformed(java.awt.event.
    ActionEvent evt) {
   jMenuItemSplitActionPerformed(evt);
  }
});
jMenuEdit.add(jMenuItemSplit);

jMenuItemInsert.setAccelerator(javax.swing.
    KeyStroke.getKeyStroke(java.awt.event.
    KeyEvent.VK_V, java.awt.event.InputEvent.
    CTRL_MASK));
jMenuItemInsert.setMnemonic(java.util.
    ResourceBundle.getBundle("net/sourceforge
    /tessboxeditor/Gui").getString("
    jMenuItemInsert.Mnemonic").charAt(0));
jMenuItemInsert.setText(bundle.getString("
    jMenuItemInsert.Text")); // NOI18N
jMenuItemInsert.addActionListener(new java.
    awt.event.ActionListener() {
  public void actionPerformed(java.awt.event.
    ActionEvent evt) {
   jMenuItemInsertActionPerformed(evt);
  }
});
jMenuEdit.add(jMenuItemInsert);

jMenuItemDelete.setAccelerator(javax.swing.
    KeyStroke.getKeyStroke(java.awt.event.
    KeyEvent.VK_DELETE, 0));
jMenuItemDelete.setMnemonic(java.util.
    ResourceBundle.getBundle("net/sourceforge
    /tessboxeditor/Gui").getString("
    jMenuItemDelete.Mnemonic").charAt(0));
jMenuItemDelete.setText(bundle.getString("
    jMenuItemDelete.Text")); // NOI18N
jMenuItemDelete.addActionListener(new java.
    awt.event.ActionListener() {
  public void actionPerformed(java.awt.event.
    ActionEvent evt) {
   jMenuItemDeleteActionPerformed(evt);
  }
});
jMenuEdit.add(jMenuItemDelete);

jMenuBar.add(jMenuEdit);

jMenuSettings.setMnemonic(java.util.
    ResourceBundle.getBundle("net/sourceforge
    /tessboxeditor/Gui").getString("
    jMenuSettings.Mnemonic").charAt(0));
jMenuSettings.setText("Settings");

jMenuItemFont.setAccelerator(javax.swing.
    KeyStroke.getKeyStroke(java.awt.event.
    KeyEvent.VK_T, java.awt.event.InputEvent.
    CTRL_MASK));
jMenuItemFont.setMnemonic(java.util.
    ResourceBundle.getBundle("net/sourceforge
    /tessboxeditor/Gui").getString("
    jMenuItemFont.Mnemonic").charAt(0));
jMenuItemFont.setText(bundle.getString("
    jMenuItemFont.Text")); // NOI18N
jMenuItemFont.addActionListener(new java.awt
    .event.ActionListener() {
  public void actionPerformed(java.awt.event.
    ActionEvent evt) {
   jMenuItemFontActionPerformed(evt);
  }
});
jMenuSettings.add(jMenuItemFont);
jMenuSettings.add(jSeparatorLAF);

jMenuLookAndFeel.setMnemonic(java.util.
    ResourceBundle.getBundle("net/sourceforge
    /tessboxeditor/Gui").getString("
    jMenuLookAndFeel.Mnemonic").charAt(0));
jMenuLookAndFeel.setText(bundle.getString("
    jMenuLookAndFeel.Text")); // NOI18N
jMenuSettings.add(jMenuLookAndFeel);

jMenuBar.add(jMenuSettings);

jMenuTools.setMnemonic(java.util.
    ResourceBundle.getBundle("net/sourceforge
    /tessboxeditor/Gui").getString("
    jMenuTools.Mnemonic").charAt(0));
jMenuTools.setText(bundle.getString("
    jMenuTools.Text")); // NOI18N

jMenuItemMergeTiff.setAccelerator(javax.
    swing.KeyStroke.getKeyStroke(java.awt.
    event.KeyEvent.VK_M, java.awt.event.
    InputEvent.CTRL_MASK));
jMenuItemMergeTiff.setMnemonic(java.util.
    ResourceBundle.getBundle("net/sourceforge
    /tessboxeditor/Gui").getString("
    jMenuItemMergeTiff.Mnemonic").charAt(0));
jMenuItemMergeTiff.setText(bundle.getString(
    "jMenuItemMergeTiff.Text")); // NOI18N
jMenuItemMergeTiff.addActionListener(new
    java.awt.event.ActionListener() {
  public void actionPerformed(java.awt.event.
    ActionEvent evt) {
   jMenuItemMergeTiffActionPerformed(evt);
  }
});
jMenuTools.add(jMenuItemMergeTiff);

jMenuItemSplitTiff.setAccelerator(javax.
    swing.KeyStroke.getKeyStroke(java.awt.
    event.KeyEvent.VK_L, java.awt.event.
    InputEvent.CTRL_MASK));
jMenuItemSplitTiff.setMnemonic(java.util.
    ResourceBundle.getBundle("net/sourceforge
    /tessboxeditor/Gui").getString("
    jMenuItemSplitTiff.Mnemonic").charAt(0));
jMenuItemSplitTiff.setText(bundle.getString(
    "jMenuItemSplitTiff.Text")); // NOI18N
jMenuItemSplitTiff.addActionListener(new
    java.awt.event.ActionListener() {
  public void actionPerformed(java.awt.event.
    ActionEvent evt) {
   jMenuItemSplitTiffActionPerformed(evt);
  }
});
jMenuTools.add(jMenuItemSplitTiff);

jMenuBar.add(jMenuTools);

jMenuHelp.setMnemonic(java.util.
    ResourceBundle.getBundle("net/sourceforge
    /tessboxeditor/Gui").getString("jMenuHelp
    .Mnemonic").charAt(0));
jMenuHelp.setText(bundle.getString("
    jMenuHelp.Text")); // NOI18N

jMenuItemHelp.setMnemonic(java.util.
    ResourceBundle.getBundle("net/sourceforge
    /tessboxeditor/Gui").getString("
    jMenuItemHelp.Mnemonic").charAt(0));
jMenuItemHelp.setText(bundle.getString("
    jMenuItemHelp.Text")); // NOI18N
jMenuItemHelp.addActionListener(new java.awt
    .event.ActionListener() {
  public void actionPerformed(java.awt.event.
    ActionEvent evt) {
   jMenuItemHelpActionPerformed(evt);
  }
});
jMenuHelp.add(jMenuItemHelp);
jMenuHelp.add(jSeparatorAbout);

jMenuItemAbout.setMnemonic(java.util.
    ResourceBundle.getBundle("net/sourceforge
    /tessboxeditor/Gui").getString("
    jMenuItemAbout.Mnemonic").charAt(0));
jMenuItemAbout.setText(bundle.getString("
    jMenuItemAbout.Text")); // NOI18N
jMenuItemAbout.addActionListener(new java.
    awt.event.ActionListener() {
  public void actionPerformed(java.awt.event.
    ActionEvent evt) {
   jMenuItemAboutActionPerformed(evt);
  }
});
jMenuHelp.add(jMenuItemAbout);

jMenuBar.add(jMenuHelp);

setJMenuBar(jMenuBar);

KeyboardFocusManager manager =
```

```
    KeyboardFocusManager.
    getCurrentKeyboardFocusManager();
manager.addKeyEventDispatcher(new
    KeyEventDispatcher() {

  private void inc(JSpinner s) {
   if (s == jSpinnerX || s == jSpinnerY) {
    if (!invertControls) {
     s.setValue(Math.max(0, ((Integer) s.
    getValue()) - movementMultiplier));
    } else {
     s.setValue(((Integer) s.getValue()) +
    movementMultiplier);
    }
   } else {
    s.setValue(((Integer) s.getValue()) +
    movementMultiplier);
   }
  }

  @SuppressWarnings("unused")
     private void inc(JSpinner s, int max) {
   if ((s == jSpinnerX || s == jSpinnerY) &&
    !invertControls) {
    s.setValue(Math.max(0, ((Integer) s.
    getValue()) - movementMultiplier));
   } else {
    s.setValue(((Integer) s.getValue()) +
    movementMultiplier);
   }
  }

  private void dec(JSpinner s) {
   if ((s == jSpinnerX || s == jSpinnerY) &&
    !invertControls) {
    s.setValue(((Integer) s.getValue()) +
    movementMultiplier);
   } else {
    s.setValue(Math.max(0, ((Integer) s.
    getValue()) - movementMultiplier));
   }
  }

  @Override
  public boolean dispatchKeyEvent(KeyEvent e)
     {

   if (e.getID() != KeyEvent.KEY_TYPED) {
    return false;
   }

   Component focusOwner =
    KeyboardFocusManager.
    getCurrentKeyboardFocusManager().
    getFocusOwner();

   if ( (focusOwner instanceof JSpinner)
    || (focusOwner instanceof JTextField)) {
    return false;
   }

   if (jLabelCharacter.hasFocus()) {
    return false;
   }

   if (!jPanelBoxView.isShowing()) {
    return false;
   }

   if (e.isShiftDown()) {
    movementMultiplier = 10;
   }
   else {
    movementMultiplier = 1;

   }

   char c = Character.toLowerCase(e.
    getKeyChar());

   if (c == 'w') {
    inc(jSpinnerY);
   } else if (c == 's') {
    dec(jSpinnerY);
   } else if (c == 'd') {
    dec(jSpinnerX);
   } else if (c == 'a') {
    inc(jSpinnerX);
   } else if (c == 'q') {
    dec(jSpinnerW);
   } else if (c == 'e') {
    inc(jSpinnerW);
   } else if (c == 'r') {
    dec(jSpinnerH);
   } else if (c == 'f') {
    inc(jSpinnerH);
   } else if (c == ',') {
    jButtonPrev.doClick();
   } else if (c == '.') {
    jButtonNext.doClick();
   } else if (c == 'x') {
    jTextFieldCharacter.requestFocus();
   } else {
    return false;
   }

   return true;
  }
 });
 pack();


}// </editor-fold>//GEN-END:initComponents

 private void jMenuItemOpenActionPerformed(
     java.awt.event.ActionEvent evt) {//GEN-
     FIRST:event_jMenuItemOpenActionPerformed
   //Old source code
  /*
      if (jFileChooserInputImage.
          showOpenDialog(this) ==
          JFileChooser.APPROVE_OPTION) {
           currentDirectory =
                jFileChooserInputImage.
                getCurrentDirectory().getPath
                ();
           openFile(jFileChooserInputImage.
                getSelectedFile());

           for (int i = 0; i < fileFilters.
                length; i++) {
               if (fileFilters[i] ==
                    jFileChooserInputImage.
                    getFileFilter()) {
                   filterIndex = i;
                   break;
               }
           }
      }
     */

  //Code by Aaron
  if (currentIndex == inputTiffs.length) {
    //Train if last index is reached
    saveBoxFile();
    System.exit(0);

  } else {
    openFile(inputTiffs[currentIndex]);
    currentIndex++;

    if (currentIndex == inputTiffs.length) {
      this.jButtonOpen.setText("Finish");
    }

  }
 }//GEN-LAST:
     event_jMenuItemOpenActionPerformed
 public void openFile(final File selectedFile)
     {
  if (!selectedFile.exists()) {
   JOptionPane.showMessageDialog(this, bundle.
     getString("File_not_exist"), APP_NAME,
     JOptionPane.ERROR_MESSAGE);
   return;
  }
  if (!promptToSave()) {
   return;
  }

//         jLabelStatus.setText(bundle.
     getString("Loading_image..."));
  getGlassPane().setCursor(Cursor.
     getPredefinedCursor(Cursor.WAIT_CURSOR));
  getGlassPane().setVisible(true);

  SwingWorker<Void, Void> loadWorker = new
     SwingWorker<Void, Void>() {

   @Override
   protected Void doInBackground() throws
     Exception {
    readImageFile(selectedFile);
    updateMRUList(selectedFile.getPath());
    int lastDot = selectedFile.getName().
     lastIndexOf(".");
    boxFile = new File(selectedFile.
     getParentFile(), selectedFile.getName().
     substring(0, lastDot) + ".box");
    readBoxFile(boxFile);
    return null;
   }

   @Override
   protected void done() {
//                 jLabelStatus.setText(bundle.
     getString("Loading_completed"));
    getGlassPane().setCursor(Cursor.
```

```
    getPredefinedCursor(Cursor.DEFAULT_CURSOR
    ));
   getGlassPane().setVisible(false);
  }
 };

 loadWorker.execute();
}

void readImageFile(File selectedFile) {
 try {
  imageList = ImageIOHelper.getImageList(
    selectedFile);
  if (imageList == null) {
   JOptionPane.showMessageDialog(this, bundle
    .getString("Cannotloadimage"), APP_NAME,
    JOptionPane.ERROR_MESSAGE);
   return;
  }
  imageIndex = 0;
  loadImage();
  this.jScrollPaneImage.getViewport().
    setViewPosition(new Point(0, 0));
  this.setTitle(APP_NAME + " - " +
    selectedFile.getName());
 } catch (OutOfMemoryError oome) {
  JOptionPane.showMessageDialog(this, oome.
    getMessage(), "Out-Of-Memory Exception",
    JOptionPane.ERROR_MESSAGE);
 } catch (IOException | HeadlessException e)
    {
  logger.log(Level.SEVERE, e.getMessage(), e)
    ;
  if (e.getMessage() != null) {
   JOptionPane.showMessageDialog(this, e.
    getMessage(), APP_NAME, JOptionPane.
    ERROR_MESSAGE);
  }
 }
}

void readBoxFile(final File boxFile) {
 if (boxFile.exists()) {
  try {
   boxPages.clear();

   // load into textarea first
   try (BufferedReader in = new
    BufferedReader(new InputStreamReader(new
    FileInputStream(boxFile), "UTF8"))) {
    this.jTextAreaBoxData.read(in, null);
   }

   // load into coordinate tab
   String[] boxdata = this.jTextAreaBoxData.
    getText().split("\\n");
   if (boxdata.length > 0) {
    // if only 5 fields, it's Tess 2.0x
    format
    isTess2_0Format = boxdata[0].split("\\s+"
    ).length == 5;
   }

   int startBoxIndex = 0;

   for (int curPage = 0; curPage < imageList.
    size(); curPage++) {
    TessBoxCollection boxCol = new
    TessBoxCollection();
    // Note that the coordinate system used
    in the box file has (0,0) at the bottom-
    left.
    // On computer graphics device, (0,0) is
    defined as top-left.
    int pageHeight = imageList.get(curPage).
    getHeight();
    for (int i = startBoxIndex; i < boxdata.
    length; i++) {
     String[] items = boxdata[i].split("\\s+"
    );

     // skip invalid data
     if (items.length < 5 || items.length >
    6) {
      continue;
     }

     String chrs = items[0];
     int x = Integer.parseInt(items[1]);
     int y = Integer.parseInt(items[2]);
     int w = Integer.parseInt(items[3]) - x;
     int h = Integer.parseInt(items[4]) - y;
     y = pageHeight - y - h; // flip the y-
    coordinate

     short page;
     if (items.length == 6) {
      page = Short.parseShort(items[5]); //
    Tess 3.0x format
     } else {
      page = 0; // Tess 2.0x format
     }
     if (page > curPage) {
      startBoxIndex = i; // mark begin of
    next page
      break;
     }
     boxCol.add(new TessBox(chrs, new
    Rectangle(x, y, w, h), page));
    }
    boxPages.add(boxCol); // add the last
    page
   }
   loadTable();
   updateSave(false);
  } catch (OutOfMemoryError oome) {
   logger.log(Level.SEVERE, oome.getMessage()
    , oome);
   JOptionPane.showMessageDialog(this, oome.
    getMessage(), "Out-Of-Memory Exception",
    JOptionPane.ERROR_MESSAGE);
  } catch (IOException |
    NumberFormatException e) {
   logger.log(Level.SEVERE, e.getMessage(), e
    );
   if (e.getMessage() != null) {
    JOptionPane.showMessageDialog(this, e.
    getMessage(), APP_NAME, JOptionPane.
    ERROR_MESSAGE);
   }
  }
 } else {
  // clear table and box display
  tableModel.setDataVector((Object[][]) null,
    (Object[]) null);
  ((JImageLabel) this.jLabelImage).setBoxes(
    null);
  jTextAreaBoxData.setText(null);
 }
}

void loadTable() {
 if (!this.boxPages.isEmpty()) {
  boxes = this.boxPages.get(imageIndex);
  tableModel.setDataVector(boxes.
    getTableDataList().toArray(new String
    [0][5]), headers); //edited by Aaron
  ((JImageLabel) this.jLabelImage).setBoxes(
    boxes);
 }
 jTable.getColumnModel().removeColumn(jTable.
    getColumnModel().getColumn(5));
}

/**
     * Displays a dialog to discard changes.
     *
     * @return false if user canceled or
         discard, true else
     */
protected boolean promptToDiscardChanges() {
 if (!boxChanged) {
  return false;
 }
 switch (JOptionPane.showConfirmDialog(this,
   bundle.getString("
    Do_you_want_to_discard_the_changes_to_")
   + boxFile.getName() + "?",
   APP_NAME, JOptionPane.YES_NO_CANCEL_OPTION
    , JOptionPane.INFORMATION_MESSAGE)) {
  case JOptionPane.YES_OPTION:
   return true;
  default:
   return false;
 }
}

/**
     * Displays a dialog to save changes.
     *
     * @return false if user canceled, true
         else
     */
protected boolean promptToSave() {
 if (!boxChanged) {
  return true;
 }
 switch (JOptionPane.showConfirmDialog(this,
   bundle.getString("
    Do_you_want_to_save_the_changes_to_")
   + (boxFile == null ? bundle.getString("
    Untitled") : boxFile.getName()) + "?",
   APP_NAME, JOptionPane.YES_NO_CANCEL_OPTION
    , JOptionPane.INFORMATION_MESSAGE)) {
  case JOptionPane.YES_OPTION:
   return saveAction();
  case JOptionPane.NO_OPTION:
   return true;
```

```
    default:
     return false;
   }
  }

  boolean saveAction() {
   if (boxFile == null || !boxFile.exists()) {
    return saveFileDlg();
   } else {
    return saveBoxFile();
   }
  }

  boolean saveFileDlg() {
   JFileChooser saveChooser = new JFileChooser(
      outputDirectory);
   FileFilter textFilter = new SimpleFilter("
      box", "Box Files");
   saveChooser.addChoosableFileFilter(
      textFilter);
   saveChooser.setFileFilter(textFilter);
   saveChooser.setDialogTitle(bundle.getString(
      "Save_As"));
   if (boxFile != null) {
    saveChooser.setSelectedFile(boxFile);
   }

   if (saveChooser.showSaveDialog(this) ==
      JFileChooser.APPROVE_OPTION) {
    outputDirectory = saveChooser.
      getCurrentDirectory().getPath();
    File f = saveChooser.getSelectedFile();
    if (saveChooser.getFileFilter() ==
      textFilter) {
     if (!f.getName().endsWith(".box")) {
      f = new File(f.getPath() + ".box");
     }
     if (boxFile != null && boxFile.getPath().
      equals(f.getPath())) {
      if (JOptionPane.NO_OPTION == JOptionPane.
      showConfirmDialog(
         Gui.this,
         boxFile.getName() + bundle.getString("
      file_already_exist"),
         bundle.getString("Confirm_Save_As"),
      JOptionPane.YES_NO_OPTION,
         JOptionPane.WARNING_MESSAGE)) {
       return false;
      }
     } else {
      boxFile = f;
     }
    } else {
     boxFile = f;
    }
    return saveBoxFile();
   } else {
    return false;
   }
  }

  boolean saveBoxFile() {
   getGlassPane().setCursor(Cursor.
      getPredefinedCursor(Cursor.WAIT_CURSOR));
   getGlassPane().setVisible(true);

   try {
    try (BufferedWriter out = new
      BufferedWriter(new OutputStreamWriter(new
       FileOutputStream(boxFile), UTF8))) {
     out.write(formatOutputString());
    }
//              updateMRUList(boxFile.getPath())
     ;
    updateSave(false);
   } catch (OutOfMemoryError oome) {
    logger.log(Level.SEVERE, oome.getMessage(),
      oome);
    JOptionPane.showMessageDialog(this, oome.
      getMessage(), bundle.getString("
      OutOfMemoryError"), JOptionPane.
      ERROR_MESSAGE);
   } catch (FileNotFoundException fnfe) {
    logger.log(Level.SEVERE, fnfe.getMessage(),
      fnfe);
   } catch (Exception e) {
    logger.log(Level.SEVERE, e.getMessage(), e)
      ;
   } finally {
    SwingUtilities.invokeLater(new Runnable() {

     @Override
     public void run() {
      getGlassPane().setCursor(Cursor.
      getPredefinedCursor(Cursor.DEFAULT_CURSOR
      ));
      getGlassPane().setVisible(false);
     }
    });
   }

   return true;
  }

  String formatOutputString() {
   StringBuilder sb = new StringBuilder();
   for (short i = 0; i < imageList.size(); i++)
      {
    int pageHeight = ((BufferedImage) imageList
      .get(i)).getHeight(); // each page (in an
       image) can have different height
    for (TessBox box : boxPages.get(i).toList()
     ) {
     Rectangle rect = box.getRect();
     sb.append(String.format("%s %d %d %d %d %d
      ", box.getChrs(), rect.x, pageHeight -
      rect.y - rect.height, rect.x + rect.width
      , pageHeight - rect.y, i)).append(EOL);
    }
   }
   if (isTess2_0Format) {
    return sb.toString().replace(" 0" + EOL,
      EOL); // strip the ending zeroes
   }
   return sb.toString();
  }

  /**
      * Update MRU List.
      *
      * @param fileName
      */
  protected void updateMRUList(String fileName)
      {
   // to be implemented in subclass
  }

  private void jMenuItemSaveActionPerformed(
      java.awt.event.ActionEvent evt) {//GEN-
      FIRST:event_jMenuItemSaveActionPerformed
   saveAction();
  }//GEN-LAST:
      event_jMenuItemSaveActionPerformed

  private void jMenuItemExitActionPerformed(
      java.awt.event.ActionEvent evt) {//GEN-
      FIRST:event_jMenuItemExitActionPerformed
   quit();
  }//GEN-LAST:
      event_jMenuItemExitActionPerformed

  void quit() {
   if (!promptToSave()) {
    return;
   }

   if (currentDirectory != null) {
    prefs.put("currentDirectory",
      currentDirectory);
   }
   if (outputDirectory != null) {
    prefs.put("outputDirectory",
      outputDirectory);
   }

   prefs.putInt("windowState", getExtendedState
      ());

   if (getExtendedState() == NORMAL) {
    prefs.putInt("frameHeight", getHeight());
    prefs.putInt("frameWidth", getWidth());
    prefs.putInt("frameX", getX());
    prefs.putInt("frameY", getY());
   }

   prefs.putInt("filterIndex", filterIndex);

   System.exit(0);
  }

  private void jMenuItemAboutActionPerformed(
      java.awt.event.ActionEvent evt) {//GEN-
      FIRST:event_jMenuItemAboutActionPerformed
   about();
  }//GEN-LAST:
      event_jMenuItemAboutActionPerformed

  void about() {
   try {
    Properties config = new Properties();
    config.loadFromXML(getClass().
      getResourceAsStream("config.xml"));
    String version = config.getProperty("
      Version");
    SimpleDateFormat sdf = new SimpleDateFormat
      ("yyyy/MM/dd");
    Date releaseDate = sdf.parse(config.
      getProperty("ReleaseDate"));
```

```
    JOptionPane.showMessageDialog(this,
      APP_NAME + " " + version + " \u00a9 2011\
      n"
      + "Tesseract Box Editor & Trainer\n"
      + DateFormat.getDateInstance(DateFormat.
      LONG).format(releaseDate)
      + "\nhttp://vietocr.sourceforge.net",
      jMenuItemAbout.getText(), JOptionPane.
      INFORMATION_MESSAGE);
    } catch (IOException | ParseException |
      HeadlessException e) {
     logger.log(Level.SEVERE, e.getMessage(), e)
      ;
    }
  }
  private void jButtonPrevPageActionPerformed(
      java.awt.event.ActionEvent evt) {//GEN-
      FIRST:
      event_jButtonPrevPageActionPerformed
    if (imageList != null && imageIndex > 0) {
     --imageIndex;
     loadImage();
     loadTable();
    }
  }//GEN-LAST:
      event_jButtonPrevPageActionPerformed

  private void jButtonNextPageActionPerformed(
      java.awt.event.ActionEvent evt) {//GEN-
      FIRST:
      event_jButtonNextPageActionPerformed
    if (imageList != null && imageIndex <
      imageList.size() - 1) {
     ++imageIndex;
     loadImage();
     loadTable();
    }
  }//GEN-LAST:
      event_jButtonNextPageActionPerformed

  void loadImage() {
    this.jLabelImage.setIcon(new ImageIcon(
      imageList.get(imageIndex)));
    if (boxes != null) {
     boxes.deselectAll();
    }
    this.jLabelImage.repaint();
    this.jLabelPageNbr.setText(String.format("
      Page: %d of %d", imageIndex + 1,
      imageList.size()));
    setButton();
    tableSelectAction = true;
    resetReadout();
    tableSelectAction = false;
  }

  void setButton() {
    if (imageIndex == 0) {
     this.jButtonPrevPage.setEnabled(false);
    } else {
     this.jButtonPrevPage.setEnabled(true);
    }

    if (imageIndex == imageList.size() - 1) {
     this.jButtonNextPage.setEnabled(false);
    } else {
     this.jButtonNextPage.setEnabled(true);
    }
  }

  void resetReadout() {
    jTextFieldCharacter.setText(null);
    jTextFieldChar.setText(null);
    jTextFieldCodepointValue.setText(null);
    jSpinnerH.setValue(0);
    jSpinnerW.setValue(0);
    jSpinnerX.setValue(0);
    jSpinnerY.setValue(0);
    jLabelSubimage.setIcon(null);
  }

  void enableReadout(boolean enabled) {
    jTextFieldCharacter.setEnabled(enabled);
    jSpinnerX.setEnabled(enabled);
    jSpinnerY.setEnabled(enabled);
    jSpinnerH.setEnabled(enabled);
    jSpinnerW.setEnabled(enabled);
  }

  /**
      * Updates the Save action.
      *
      * @param modified whether file has been
          modified
      */
  void updateSave(boolean modified) {
    if (boxChanged != modified) {
     boxChanged = modified;
     this.jButtonSave.setEnabled(modified);
     this.jMenuItemSave.setEnabled(modified);
     rootPane.putClientProperty("windowModified"
       , modified);
     // see http://developer.apple.com/qa/qa2001
       /qa1146.html
    }
  }

  private void jButtonOpenActionPerformed(java.
      awt.event.ActionEvent evt) {//GEN-FIRST:
      event_jButtonOpenActionPerformed
    jMenuItemOpenActionPerformed(evt);
  }//GEN-LAST:event_jButtonOpenActionPerformed

  private void jButtonSaveActionPerformed(java.
      awt.event.ActionEvent evt) {//GEN-FIRST:
      event_jButtonSaveActionPerformed
    jMenuItemSaveActionPerformed(evt);
  }//GEN-LAST:event_jButtonSaveActionPerformed

  private void jButtonReloadActionPerformed(
      java.awt.event.ActionEvent evt) {//GEN-
      FIRST:event_jButtonReloadActionPerformed
    if (!promptToDiscardChanges()) {
     return;
    }

    if (boxFile != null) {
     jButtonReload.setEnabled(false);
     getGlassPane().setCursor(Cursor.
       getPredefinedCursor(Cursor.WAIT_CURSOR));
     getGlassPane().setVisible(true);

     SwingWorker<Void, Void> loadWorker = new
       SwingWorker<Void, Void>() {

      @Override
      protected Void doInBackground() throws
       Exception {
       readBoxFile(boxFile);
       return null;
      }

      @Override
      protected void done() {
       jButtonReload.setEnabled(true);
       getGlassPane().setCursor(Cursor.
       getPredefinedCursor(Cursor.DEFAULT_CURSOR
       ));
       getGlassPane().setVisible(false);
      }
     };

     loadWorker.execute();
    }
  }//GEN-LAST:
      event_jButtonReloadActionPerformed

  void jMenuItemFontActionPerformed(java.awt.
      event.ActionEvent evt) {//GEN-FIRST:
      event_jMenuItemFontActionPerformed
    JOptionPane.showMessageDialog(this,
      TO_BE_IMPLEMENTED);
  }//GEN-LAST:
      event_jMenuItemFontActionPerformed

  private void jMenuItemHelpActionPerformed(
      java.awt.event.ActionEvent evt) {//GEN-
      FIRST:event_jMenuItemHelpActionPerformed
    final String readme = bundle.getString("
      readme");
    if (MAC_OS_X) {
     try {
      final File supportDir = new File(System.
       getProperty("user.home") + "/Library/
       Application Support/" + APP_NAME);
      if (!supportDir.exists()) {
       supportDir.mkdirs();
      }
      File helpFile = new File(supportDir, "
       readme.html");
      copyFileFromJarToSupportDir(helpFile);
      Runtime.getRuntime().exec(new String[]{"
       open", "-b", "com.apple.helpviewer",
       readme}, null, supportDir);
     } catch (IOException e) {
      logger.log(Level.SEVERE, e.getMessage(), e
       );
     }
    } else {
     if (helptopicsFrame == null) {
      helptopicsFrame = new JFrame(jMenuItemHelp
       .getText());
      helptopicsFrame.getContentPane().setLayout
       (new BorderLayout());
      HtmlPane helpPane = new HtmlPane(readme);
      helptopicsFrame.getContentPane().add(
       helpPane, BorderLayout.CENTER);
```

```
    helptopicsFrame.getContentPane().add(
     helpPane.getStatusBar(), BorderLayout.
     SOUTH);
    helptopicsFrame.pack();
    helptopicsFrame.setLocation((screen.width
     - helptopicsFrame.getWidth()) / 2, 40);
   }
   helptopicsFrame.setVisible(true);
  }
 }//GEN-LAST:
     event_jMenuItemHelpActionPerformed

 private void copyFileFromJarToSupportDir(File
      helpFile) throws IOException {
  if (!helpFile.exists()) {
   try (ReadableByteChannel input = Channels.
     newChannel(ClassLoader.
     getSystemResourceAsStream(helpFile.
     getName()));
      FileChannel output = new FileOutputStream
     (helpFile).getChannel()) {
    output.transferFrom(input, 0, 1000000L);
   }
  }
 }

 private void jMenuItemSaveAsActionPerformed(
     java.awt.event.ActionEvent evt) {//GEN-
     FIRST:
     event_jMenuItemSaveAsActionPerformed
  saveFileDlg();
 }//GEN-LAST:
     event_jMenuItemSaveAsActionPerformed
 private void jSpinnerXStateChanged(javax.
     swing.event.ChangeEvent evt) {//GEN-FIRST
     :event_jSpinnerXStateChanged
  stateChanged(evt);
 }//GEN-LAST:event_jSpinnerXStateChanged
 private void jSpinnerYStateChanged(javax.
     swing.event.ChangeEvent evt) {//GEN-FIRST
     :event_jSpinnerYStateChanged
  stateChanged(evt);
 }//GEN-LAST:event_jSpinnerYStateChanged
 private void jSpinnerWStateChanged(javax.
     swing.event.ChangeEvent evt) {//GEN-FIRST
     :event_jSpinnerWStateChanged
  stateChanged(evt);
 }//GEN-LAST:event_jSpinnerWStateChanged
 private void jSpinnerHStateChanged(javax.
     swing.event.ChangeEvent evt) {//GEN-FIRST
     :event_jSpinnerHStateChanged
  stateChanged(evt);
 }//GEN-LAST:event_jSpinnerHStateChanged
 void stateChanged(javax.swing.event.
     ChangeEvent evt) {
  JOptionPane.showMessageDialog(this,
     TO_BE_IMPLEMENTED);
 }
 private void
     jTextFieldCharacterActionPerformed(java.
     awt.event.ActionEvent evt) {//GEN-FIRST:
     event_jTextFieldCharacterActionPerformed
  if (boxes == null) {
   return;
  }
  List<TessBox> selected = this.boxes.
     getSelectedBoxes();
  if (selected.size() <= 0) {
   return;
  } else if (selected.size() > 1) {
   JOptionPane.showMessageDialog(this, "Please
      select only one box to apply the change.
     ");
   return;
  }

  TessBox box = selected.get(0);
  int index = this.boxes.toList().indexOf(box)
     ;
  // Convert NCR or escape sequence to Unicode
     .
  this.jTextFieldCharacter.setText(
     TextUtilities.convertNCR(this.
     jTextFieldCharacter.getText()));
  String str = this.jTextFieldCharacter.
     getText();
  if (!box.getChrs().equals(str)) {
   box.setChrs(str);
   tableModel.setValueAt(box.getChrs(), index,
      0);
   jTextFieldChar.setText(str);
   jTextFieldCodepointValue.setText(Utils.
     toHex(str));
   updateSave(true);
  }
 }//GEN-LAST:
     event_jTextFieldCharacterActionPerformed
 private void jButtonConvertActionPerformed(
     java.awt.event.ActionEvent evt) {//GEN-
     FIRST:event_jButtonConvertActionPerformed
  String curChar = this.jTextFieldCharacter.
     getText();
  if (curChar.trim().length() == 0) {
   return;
  }
  // Convert NCR or escape sequence to Unicode
     .
  this.jTextFieldCharacter.setText(
     TextUtilities.convertNCR(this.
     jTextFieldCharacter.getText()));
  // Commit the change, if no conversion.
  if (curChar.equals(this.jTextFieldCharacter.
     getText())) {
   jTextFieldCharacterActionPerformed(evt);
  }
 }//GEN-LAST:
     event_jButtonConvertActionPerformed
 void jMenuItemMergeActionPerformed(java.awt.
     event.ActionEvent evt) {//GEN-FIRST:
     event_jMenuItemMergeActionPerformed
  JOptionPane.showMessageDialog(this,
     TO_BE_IMPLEMENTED);
 }//GEN-LAST:
     event_jMenuItemMergeActionPerformed
 void jMenuItemSplitActionPerformed(java.awt.
     event.ActionEvent evt) {//GEN-FIRST:
     event_jMenuItemSplitActionPerformed
  JOptionPane.showMessageDialog(this,
     TO_BE_IMPLEMENTED);
 }//GEN-LAST:
     event_jMenuItemSplitActionPerformed
 void jMenuItemDeleteActionPerformed(java.awt.
     event.ActionEvent evt) {//GEN-FIRST:
     event_jMenuItemDeleteActionPerformed
  JOptionPane.showMessageDialog(this,
     TO_BE_IMPLEMENTED);
 }//GEN-LAST:
     event_jMenuItemDeleteActionPerformed
 void jMenuItemInsertActionPerformed(java.awt.
     event.ActionEvent evt) {//GEN-FIRST:
     event_jMenuItemInsertActionPerformed
  JOptionPane.showMessageDialog(this,
     TO_BE_IMPLEMENTED);
 }//GEN-LAST:
     event_jMenuItemInsertActionPerformed
 private void jButtonMergeActionPerformed(java
     .awt.event.ActionEvent evt) {//GEN-FIRST:
     event_jButtonMergeActionPerformed
  jMenuItemMergeActionPerformed(evt);
 }//GEN-LAST:event_jButtonMergeActionPerformed
 private void jButtonDeleteActionPerformed(
     java.awt.event.ActionEvent evt) {//GEN-
     FIRST:event_jButtonDeleteActionPerformed
  jMenuItemDeleteActionPerformed(evt);
 }//GEN-LAST:
     event_jButtonDeleteActionPerformed
 private void jButtonSplitActionPerformed(java
     .awt.event.ActionEvent evt) {//GEN-FIRST:
     event_jButtonSplitActionPerformed
  jMenuItemSplitActionPerformed(evt);
 }//GEN-LAST:event_jButtonSplitActionPerformed
 private void jButtonInsertActionPerformed(
     java.awt.event.ActionEvent evt) {//GEN-
     FIRST:event_jButtonInsertActionPerformed
  jMenuItemInsertActionPerformed(evt);
 }//GEN-LAST:
     event_jButtonInsertActionPerformed
 private void jButtonFindActionPerformed(java.
     awt.event.ActionEvent evt) {//GEN-FIRST:
     event_jButtonFindActionPerformed
  if (imageList == null) {
   return;
  }
  int pageHeight = imageList.get(imageIndex).
     getHeight();
  String[] items = this.jTextFieldFind.getText
     ().split("\\s+");
  try {
   TessBox findBox;

   if (items.length == 1) {
    String chrs = items[0];
    if (chrs.length() == 0) {
     throw new Exception("Empty search values.
     ");
    }
    // Convert NCR or escape sequence to
     Unicode.
    chrs = TextUtilities.convertNCR(chrs);

    findBox = new TessBox(chrs, new Rectangle
     (), imageIndex);
    findBox = boxes.selectByChars(findBox);
   } else {
    int x = Integer.parseInt(items[0]);
    int y = Integer.parseInt(items[1]);
    int w = Integer.parseInt(items[2]) - x;
    int h = Integer.parseInt(items[3]) - y;
```

```
    y = pageHeight - y - h; // flip the y-
     coordinate
    findBox = new TessBox("", new Rectangle(x,
      y, w, h), imageIndex);
    findBox = boxes.select(findBox);
   }

   if (findBox != null) {
    int index = boxes.toList().indexOf(findBox
     );
    this.jTable.setRowSelectionInterval(index,
      index);
    Rectangle rect = this.jTable.getCellRect(
     index, 0, true);
    this.jTable.scrollRectToVisible(rect);
   } else {
    this.jTable.clearSelection();
    String msg = String.format("No box with
     the specified %s was found.", items.
     length == 1 ? "character(s)" : "
     coordinates");
    JOptionPane.showMessageDialog(this, msg);
   }
  } catch (Exception e) {
   JOptionPane.showMessageDialog(this, "Please
     enter box character(s) or coordinates (
     x1 y1 x2 y2).");
  }
 }//GEN-LAST:event_jButtonFindActionPerformed

 private void jTextFieldFindActionPerformed(
     java.awt.event.ActionEvent evt) {//GEN-
     FIRST:event_jTextFieldFindActionPerformed
  jButtonFindActionPerformed(evt);
 }//GEN-LAST:
     event_jTextFieldFindActionPerformed

 void jMenuItemMergeTiffActionPerformed(java.
     awt.event.ActionEvent evt) {//GEN-FIRST:
     event_jMenuItemMergeTiffActionPerformed
  JOptionPane.showMessageDialog(this,
     TO_BE_IMPLEMENTED);
 }//GEN-LAST:
     event_jMenuItemMergeTiffActionPerformed

  void jMenuItemSplitTiffActionPerformed(java.
      awt.event.ActionEvent evt) {//GEN-FIRST
      :
      event_jMenuItemSplitTiffActionPerformed
  JOptionPane.showMessageDialog(this,
     TO_BE_IMPLEMENTED);
 }//GEN-LAST:
     event_jMenuItemSplitTiffActionPerformed

 private void jButtonPrevActionPerformed(java.
     awt.event.ActionEvent evt) {//GEN-FIRST:
     event_jButtonPrevActionPerformed
  int index = this.jTable.getSelectedRow();
  if (index > 0) {
   boxes.deselectAll();
   this.jTable.clearSelection();
   --index;
   java.util.List<TessBox> boxesOfCurPage =
     boxes.toList(); // boxes of current page
   TessBox selected = boxesOfCurPage.get(index
     );
   selected.setSelected(true);
   this.jTable.addRowSelectionInterval(index,
     index);
   Rectangle rect = this.jTable.getCellRect(
     index, 0, true);
   this.jTable.scrollRectToVisible(rect);
  }
 }//GEN-LAST:event_jButtonPrevActionPerformed

 private void jButtonNextActionPerformed(java.
     awt.event.ActionEvent evt) {//GEN-FIRST:
     event_jButtonNextActionPerformed
  if (boxes == null) {
   return;
  }
  java.util.List<TessBox> boxesOfCurPage =
     boxes.toList(); // boxes of current page
  int index = this.jTable.getSelectedRow();
  if (index < boxesOfCurPage.size() - 1) {
   boxes.deselectAll();
   this.jTable.clearSelection();
   ++index;
   TessBox selected = boxesOfCurPage.get(index
     );
   selected.setSelected(true);
   this.jTable.addRowSelectionInterval(index,
     index);
   Rectangle rect = this.jTable.getCellRect(
     index, 0, true);
   this.jTable.scrollRectToVisible(rect);
  }
 }//GEN-LAST:event_jButtonNextActionPerformed

 void jButtonInputActionPerformed(java.awt.
     event.ActionEvent evt) {//GEN-FIRST:
     event_jButtonInputActionPerformed
  JOptionPane.showMessageDialog(this,
     TO_BE_IMPLEMENTED);
 }//GEN-LAST:event_jButtonInputActionPerformed

 void jButtonFontActionPerformed(java.awt.
     event.ActionEvent evt) {//GEN-FIRST:
     event_jButtonFontActionPerformed
  JOptionPane.showMessageDialog(this,
     TO_BE_IMPLEMENTED);
 }//GEN-LAST:event_jButtonFontActionPerformed

 void jSpinnerTrackingStateChanged(javax.swing
     .event.ChangeEvent evt) {//GEN-FIRST:
     event_jSpinnerTrackingStateChanged
  JOptionPane.showMessageDialog(this,
     TO_BE_IMPLEMENTED);
 }//GEN-LAST:
     event_jSpinnerTrackingStateChanged

 void jButtonGenerateActionPerformed(java.awt.
     event.ActionEvent evt) {//GEN-FIRST:
     event_jButtonGenerateActionPerformed
  JOptionPane.showMessageDialog(this,
     TO_BE_IMPLEMENTED);
 }//GEN-LAST:
     event_jButtonGenerateActionPerformed

 void jButtonClearActionPerformed(java.awt.
     event.ActionEvent evt) {//GEN-FIRST:
     event_jButtonClearActionPerformed
  JOptionPane.showMessageDialog(this,
     TO_BE_IMPLEMENTED);
 }//GEN-LAST:event_jButtonClearActionPerformed

 void jButtonBrowseTessActionPerformed(java.
     awt.event.ActionEvent evt) {//GEN-FIRST:
     event_jButtonBrowseTessActionPerformed
  JOptionPane.showMessageDialog(this,
     TO_BE_IMPLEMENTED);
 }//GEN-LAST:
     event_jButtonBrowseTessActionPerformed

 void jButtonBrowseDataActionPerformed(java.
     awt.event.ActionEvent evt) {//GEN-FIRST:
     event_jButtonBrowseDataActionPerformed
  JOptionPane.showMessageDialog(this,
     TO_BE_IMPLEMENTED);
 }//GEN-LAST:
     event_jButtonBrowseDataActionPerformed

 void jButtonTrainActionPerformed(java.awt.
     event.ActionEvent evt) {//GEN-FIRST:
     event_jButtonTrainActionPerformed
  JOptionPane.showMessageDialog(this,
     TO_BE_IMPLEMENTED);
 }//GEN-LAST:event_jButtonTrainActionPerformed

 void jButtonCancelActionPerformed(java.awt.
     event.ActionEvent evt) {//GEN-FIRST:
     event_jButtonCancelActionPerformed
  JOptionPane.showMessageDialog(this,
     TO_BE_IMPLEMENTED);
 }//GEN-LAST:
     event_jButtonCancelActionPerformed

 private void jTabbedPaneMainStateChanged(
     javax.swing.event.ChangeEvent evt) {//GEN
     -FIRST:event_jTabbedPaneMainStateChanged
  JTabbedPane pane = (JTabbedPane) evt.
     getSource();
  boolean boxEditorActive = pane.
     getSelectedIndex() == 1;
  this.jMenuEdit.setVisible(boxEditorActive);
  this.jMenuItemFont.setVisible(
     boxEditorActive || pane.getSelectedIndex
     () == 0);
  this.jSeparatorLAF.setVisible(this.
     jMenuItemFont.isVisible());
  this.jMenuItemOpen.setVisible(
     boxEditorActive);
  this.jMenuItemSave.setVisible(
     boxEditorActive);
  this.jMenuItemSaveAs.setVisible(
     boxEditorActive);
  this.jMenuRecentFiles.setVisible(
     boxEditorActive);
  this.jSeparatorRecentFiles.setVisible(
     boxEditorActive);
  this.jSeparatorExit.setVisible(
     boxEditorActive);
 }//GEN-LAST:event_jTabbedPaneMainStateChanged

 void jButtonClearLogActionPerformed(java.awt.
     event.ActionEvent evt) {//GEN-FIRST:
     event_jButtonClearLogActionPerformed
  JOptionPane.showMessageDialog(this,
```

```
    TO_BE_IMPLEMENTED);
}//GEN-LAST:
    event_jButtonClearLogActionPerformed

void jButtonSaveLogActionPerformed(java.awt.
    event.ActionEvent evt) {//GEN-FIRST:
    event_jButtonSaveLogActionPerformed
 JOptionPane.showMessageDialog(this,
    TO_BE_IMPLEMENTED);
}//GEN-LAST:
    event_jButtonSaveLogActionPerformed

void jButtonValidateActionPerformed(java.awt.
    event.ActionEvent evt) {//GEN-FIRST:
    event_jButtonValidateActionPerformed
 JOptionPane.showMessageDialog(this,
    TO_BE_IMPLEMENTED);
}//GEN-LAST:
    event_jButtonValidateActionPerformed

void jButtonCloseDialogActionPerformed(java.
    awt.event.ActionEvent evt) {//GEN-FIRST:
    event_jButtonCloseDialogActionPerformed
 JOptionPane.showMessageDialog(this,
    TO_BE_IMPLEMENTED);
}//GEN-LAST:
    event_jButtonCloseDialogActionPerformed

private void jSpinnerMarginStateChanged(javax
    .swing.event.ChangeEvent evt) {//GEN-
    FIRST:event_jSpinnerMarginStateChanged
 iconMargin = (Integer) jSpinnerMargin.
    getValue();
 int index = jTable.getSelectedRow();
 jTable.clearSelection();
 jTable.setRowSelectionInterval(index, index)
    ;
 jLabelSubimage.requestFocus();
}//GEN-LAST:event_jSpinnerMarginStateChanged

private void jSpinnerScaleStateChanged(javax.
    swing.event.ChangeEvent evt) {//GEN-FIRST
    :event_jSpinnerScaleStateChanged
 scaleFactor = (Integer) jSpinnerScale.
    getValue();
 int index = jTable.getSelectedRow();
 jTable.clearSelection();
 jTable.setRowSelectionInterval(index, index)
    ;
 jLabelSubimage.requestFocus();
}//GEN-LAST:event_jSpinnerScaleStateChanged

private void jTextFieldCharacterKeyReleased(
    java.awt.event.KeyEvent evt) {//GEN-FIRST
    :event_jTextFieldCharacterKeyReleased
 if (evt.getKeyCode() == KeyEvent.VK_ESCAPE
    || evt.getKeyCode() == KeyEvent.VK_ENTER)
     {
  jLabelSubimage.requestFocus();
 }
}//GEN-LAST:
    event_jTextFieldCharacterKeyReleased

void jButtonBrowseOutputDirActionPerformed(
    java.awt.event.ActionEvent evt) {//GEN-
    FIRST:
    event_jButtonBrowseOutputDirActionPerformed

 JOptionPane.showMessageDialog(this,
    TO_BE_IMPLEMENTED);
}//GEN-LAST:
    event_jButtonBrowseOutputDirActionPerformed


/**
     * Gets a subimage for display in boxview.
     *
     * @param image
     * @param rect
     * @return
     */
BufferedImage getSubimage(BufferedImage image
    , Rectangle rect) {
 Gui.iconPosX = rect.x;
 Gui.iconPosY = rect.y;

 Gui.imageWidth = image.getWidth();
 Gui.imageHeight = image.getHeight();

 Gui.iconHeight = rect.height;
 Gui.iconWidth = rect.width;

 int height = Gui.iconHeight + Gui.iconMargin
     * 2;
 int width = Gui.iconWidth + Gui.iconMargin *
     2;

 while (width + Gui.iconPosX > image.getWidth
    () + 1) {
  width -= 1;
 }

 while (height + Gui.iconPosY > image.
    getHeight() + 1) {
  height -= 1;
 }

 BufferedImage subImage = image.getSubimage(
    Math.max(0, Math.min(Gui.imageWidth - 1,
    Gui.iconPosX - Gui.iconMargin)),
    Math.max(0, Math.min(Gui.imageHeight - 1,
    Gui.iconPosY - Gui.iconMargin)),
    width,
    height
 );

 return subImage;
}

/**
     * @param args the command line arguments
     */
public static void main(String args[]) {

 java.awt.EventQueue.invokeLater(new Runnable
    () {

  @Override
  public void run() {
   new Gui().setVisible(true);
  }
 });
}
// Variables declaration - do not modify//GEN
    -BEGIN:variables
private javax.swing.Box.Filler filler2;
protected javax.swing.JButton
    jButtonBrowseData;
private javax.swing.JButton
    jButtonBrowseOutputDir;
protected javax.swing.JButton
    jButtonBrowseTess;
protected javax.swing.JButton jButtonCancel;
private javax.swing.JButton jButtonClear;
protected javax.swing.JButton jButtonClearLog
    ;
private javax.swing.JButton
    jButtonCloseDialog;
private javax.swing.JButton jButtonConvert;
private javax.swing.JButton jButtonDelete;
private javax.swing.JButton jButtonFind;
protected javax.swing.JButton jButtonFont;
protected javax.swing.JButton jButtonGenerate
    ;
private javax.swing.JButton jButtonInput;
private javax.swing.JButton jButtonInsert;
private javax.swing.JButton jButtonMerge;
private javax.swing.JButton jButtonNext;
private javax.swing.JButton jButtonNextPage;
private javax.swing.JButton jButtonOpen;
private javax.swing.JButton jButtonPrev;
private javax.swing.JButton jButtonPrevPage;
private javax.swing.JButton jButtonReload;
private javax.swing.JButton jButtonSave;
private javax.swing.JButton jButtonSaveLog;
private javax.swing.JButton jButtonSplit;
protected javax.swing.JButton jButtonTrain;
protected javax.swing.JButton jButtonValidate
    ;
protected javax.swing.JCheckBox
    jCheckBoxAntiAliasing;
protected javax.swing.JCheckBox jCheckBoxRTL;
@SuppressWarnings("rawtypes")
 protected javax.swing.JComboBox jComboBoxOps
     ;
protected javax.swing.JDialog
    jDialogValidationResult;
private javax.swing.JFileChooser
    jFileChooserInputImage;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabelCharacter;
private javax.swing.JLabel jLabelCodepoint;
private javax.swing.JLabel jLabelH;
private javax.swing.JLabel jLabelH1;
protected javax.swing.JLabel jLabelImage;
private javax.swing.JLabel jLabelNoise;
private javax.swing.JLabel jLabelOutput;
private javax.swing.JLabel jLabelPageNbr;
private javax.swing.JLabel
    jLabelSpinnerMargin;
private javax.swing.JLabel jLabelSpinnerScale
    ;
protected javax.swing.JLabel jLabelStatus;
protected javax.swing.JLabel jLabelSubimage;
protected javax.swing.JLabel jLabelTime;
```

```
    private javax.swing.JLabel jLabelTracking;
    private javax.swing.JLabel jLabelW;
    private javax.swing.JLabel jLabelW1;
    private javax.swing.JLabel jLabelX;
    private javax.swing.JLabel jLabelY;
    private javax.swing.JMenuBar jMenuBar;
    private javax.swing.JMenu jMenuEdit;
    private javax.swing.JMenu jMenuFile;
    private javax.swing.JMenu jMenuHelp;
    private javax.swing.JMenuItem jMenuItemAbout;
    private javax.swing.JMenuItem jMenuItemDelete
        ;
    private javax.swing.JMenuItem jMenuItemExit;
    private javax.swing.JMenuItem jMenuItemFont;
    private javax.swing.JMenuItem jMenuItemHelp;
    private javax.swing.JMenuItem jMenuItemInsert
        ;
    private javax.swing.JMenuItem jMenuItemMerge;
    private javax.swing.JMenuItem
        jMenuItemMergeTiff;
    private javax.swing.JMenuItem jMenuItemOpen;
    private javax.swing.JMenuItem jMenuItemSave;
    private javax.swing.JMenuItem jMenuItemSaveAs
        ;
    private javax.swing.JMenuItem jMenuItemSplit;
    private javax.swing.JMenuItem
        jMenuItemSplitTiff;
    protected javax.swing.JMenu jMenuLookAndFeel;
    protected javax.swing.JMenu jMenuRecentFiles;
    private javax.swing.JMenu jMenuSettings;
    private javax.swing.JMenu jMenuTools;
    private javax.swing.JPanel jPanel3;
    private javax.swing.JPanel jPanel4;
    private javax.swing.JPanel jPanelBoxView;
    private javax.swing.JPanel jPanelButtons;
    private javax.swing.JPanel jPanelChar;
    private javax.swing.JPanel jPanelCommand;
    private javax.swing.JPanel jPanelControls;
    protected javax.swing.JPanel jPanelCoord;
    private javax.swing.JPanel jPanelEditor;
    private javax.swing.JPanel jPanelFind;
    private javax.swing.JPanel jPanelMain;
    private javax.swing.JPanel
        jPanelNorthContainer;
    private javax.swing.JPanel jPanelSpinner;
    private javax.swing.JPanel jPanelStatus;
    private javax.swing.JPanel jPanelStatus1;
    private javax.swing.JPanel jPanelTIFFBox;
    private javax.swing.JPanel jPanelTrainer;
    protected javax.swing.JProgressBar
        jProgressBar1;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JScrollPane jScrollPane2;
    private javax.swing.JScrollPane
        jScrollPaneBoxData;
    private javax.swing.JScrollPane
        jScrollPaneCoord;
    private javax.swing.JScrollPane
        jScrollPaneImage;
    private javax.swing.JScrollPane
        jScrollPaneText;
    private javax.swing.JPopupMenu.Separator
        jSeparatorAbout;
    private javax.swing.JPopupMenu.Separator
        jSeparatorExit;
    private javax.swing.JPopupMenu.Separator
        jSeparatorLAF;
    private javax.swing.JPopupMenu.Separator
        jSeparatorRecentFiles;
    protected javax.swing.JSpinner jSpinnerH;
    protected javax.swing.JSpinner jSpinnerH1;
    private javax.swing.JSpinner jSpinnerMargin;
    protected javax.swing.JSpinner jSpinnerNoise;
    private javax.swing.JSpinner jSpinnerScale;
    protected javax.swing.JSpinner
        jSpinnerTracking;
    protected javax.swing.JSpinner jSpinnerW;
    protected javax.swing.JSpinner jSpinnerW1;
    protected javax.swing.JSpinner jSpinnerX;
    protected javax.swing.JSpinner jSpinnerY;
    private javax.swing.JSplitPane
        jSplitPaneEditor;
    private javax.swing.JTabbedPane
        jTabbedPaneBoxData;
    private javax.swing.JTabbedPane
        jTabbedPaneMain;
    protected javax.swing.JTable jTable;
    protected javax.swing.JTextArea
        jTextAreaBoxData;
    protected javax.swing.JTextArea
        jTextAreaInput;
    protected javax.swing.JTextArea
        jTextAreaOutput;
    protected javax.swing.JTextArea
        jTextAreaValidationResult;
    protected javax.swing.JTextField
        jTextFieldBootstrapLang;
    protected javax.swing.JTextField
        jTextFieldChar;
    protected javax.swing.JTextField
        jTextFieldCharacter;
    private javax.swing.JTextField
        jTextFieldCodepointValue;
    protected javax.swing.JTextField
        jTextFieldDataDir;
    protected javax.swing.JTextField
        jTextFieldFileName;
    protected javax.swing.JTextField
        jTextFieldFind;
    protected javax.swing.JTextField
        jTextFieldLang;
    protected javax.swing.JTextField
        jTextFieldOuputDir;
    protected javax.swing.JTextField
        jTextFieldPrefix;
    protected javax.swing.JTextField
        jTextFieldTessDir;
    private javax.swing.JToolBar jToolBarEditor;
    private javax.swing.JToolBar
        jToolBarGenerator;
    private javax.swing.JToolBar jToolBarTrainer;
    // End of variables declaration//GEN-END:
        variables
    private JFrame helptopicsFrame;
}
```

Path = Tess2SpeechTrainer/src/jTessBoxEditor/src/net/sourceforge/tessboxeditor/GuiWithEdit.java

```
/**
 * Copyright @ 2011 Quan Nguyen
 *
 * Licensed under the Apache License, Version
     2.0 (the "License"); you may not
 * use this file except in compliance with the
     License. You may obtain a copy of
 * the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed
     to in writing, software
 * distributed under the License is
     distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND,
     either express or implied. See the
 * License for the specific language governing
     permissions and limitations under
 * the License.
 */
package net.sourceforge.tessboxeditor;

import java.awt.Rectangle;
import java.awt.event.ActionEvent;
import java.util.ArrayList;
import java.util.List;
import javax.swing.JOptionPane;
import net.sourceforge.tessboxeditor.datamodel
    .TessBox;

public class GuiWithEdit extends GuiWithMRU {

 @Override
 void jMenuItemMergeActionPerformed(java.awt.
     event.ActionEvent evt) {
  if (boxes == null) {
   return;
  }

  List<TessBox> selected = boxes.
     getSelectedBoxes();
  if (selected.size() <= 1) {
   JOptionPane.showMessageDialog(this, "Please
      select more than one box for Merge
     operation.");
```

```
        return;
    }
    /*
            * Added by Anter Aaron Custodio to
                save the highlighted character
            */
    for (int i = 0; i < jTable.getRowCount(); i
        ++) {
        if (jTable.getModel().getValueAt(i, 5) !=
            null && (boolean)jTable.getModel().
            getValueAt(i, 5) == true) {
            boxes.toList().get(i).setSelected(true);
        }
    }

    int minX = Integer.MAX_VALUE, minY = Integer
        .MAX_VALUE, maxX = 0, maxY = 0;

    String chrs = "";
    short page = 0;
    int index = 0;

    for (TessBox box : selected) {
        chrs += box.getChrs();
        page = box.getPage();
        index = this.boxes.toList().indexOf(box);
        Rectangle rect = box.getRect();
        minX = Math.min(minX, rect.x);
        minY = Math.min(minY, rect.y);
        maxX = Math.max(maxX, rect.x + rect.width);
        maxY = Math.max(maxY, rect.y + rect.height)
            ;
        this.boxes.remove(box);
    }

    ArrayList<Integer> selectedIndex = new
        ArrayList<>(); //Added by Anter Aaron
        Custodio

    if (chrs.length() > 0) {
        TessBox newBox = new TessBox(chrs, new
            Rectangle(minX, minY, maxX - minX, maxY -
            minY), page);
        newBox.setSelected(true);
        boxes.add(index, newBox);
        int tableIndex = this.boxes.toList().
            indexOf(newBox);

        /*
                * Added by Anter Aaron Custodio
                */
        for (int i = 0; i < boxes.toList().size();
            i++) {
            if (boxes.toList().get(i).isSelected()) {
                selectedIndex.add(i);
            }
        }


        tableModel.setDataVector(boxes.
            getTableDataList().toArray(new String
            [0][5]), headers);
        this.jTable.setRowSelectionInterval(
            tableIndex, tableIndex);
        Rectangle rect = this.jTable.getCellRect(
            tableIndex, 0, true);
        this.jTable.scrollRectToVisible(rect);
    }

    /*
            * Added by Anter Aaron Custodio Get
                all the highlighted boxes after
                removing merged boxes
            */
    for (int i = 0; i < selectedIndex.size(); i
        ++) {
        jTable.getModel().setValueAt(true,
            selectedIndex.get(i), 5);
    }

    this.jLabelImage.repaint();
    updateSave(true);


    jTable.getColumnModel().removeColumn(jTable.
        getColumnModel().getColumn(5));
}

@Override
void jMenuItemSplitActionPerformed(java.awt.
    event.ActionEvent evt) {
    if (boxes == null) {
        return;
    }
    List<TessBox> selected = boxes.
        getSelectedBoxes();
    if (selected.size() <= 0) {
        JOptionPane.showMessageDialog(this, "Please
            select a box to split.");
        return;
    } else if (selected.size() > 1) {
        JOptionPane.showMessageDialog(this, "Please
            select only one box for Split operation.
            ");
        return;
    }

    boolean modifierKeyPressed = false;
    int modifiers = evt.getModifiers();
    if ((modifiers & ActionEvent.CTRL_MASK) ==
        ActionEvent.CTRL_MASK
        || (modifiers & ActionEvent.ALT_MASK) ==
        ActionEvent.ALT_MASK
        || (modifiers & ActionEvent.META_MASK) ==
        ActionEvent.META_MASK) {
        modifierKeyPressed = true;
    }

    TessBox box = selected.get(0);
    int index = this.boxes.toList().indexOf(box)
        ;
    Rectangle rect = box.getRect();
    if (!modifierKeyPressed) {
        rect.width /= 2;
        tableModel.setValueAt(String.valueOf(rect.
            width), index, 3);
    } else {
        rect.height /= 2;
        tableModel.setValueAt(String.valueOf(rect.
            height), index, 4);
    }

    TessBox newBox = new TessBox(box.getChrs(),
        new Rectangle(rect), box.getPage());
    newBox.setSelected(true);
    boxes.add(index + 1, newBox);
    Rectangle newRect = newBox.getRect();
    if (!modifierKeyPressed) {
        newRect.x += newRect.width;
    } else {
        newRect.y += newRect.height;
    }

    Object[] newRow = {newBox.getChrs(), newRect
        .x, newRect.y, newRect.width, newRect.
        height};
    tableModel.insertRow(index + 1, newRow);
    jTable.setRowSelectionInterval(index, index
        + 1);
    resetReadout();
    this.jLabelImage.repaint();
    updateSave(true);
}

@Override
void jMenuItemInsertActionPerformed(java.awt.
    event.ActionEvent evt) {
    if (boxes == null) {
        return;
    }
    List<TessBox> selected = boxes.
        getSelectedBoxes();
    if (selected.size() <= 0) {
        JOptionPane.showMessageDialog(this, "Please
            select the box to insert after.");
        return;
    } else if (selected.size() > 1) {
        JOptionPane.showMessageDialog(this, "Please
            select only one box for Insert operation
            .");
        return;
    }

    TessBox box = selected.get(0);
    int index = this.boxes.toList().indexOf(box)
        ;
    index++;
    TessBox newBox = new TessBox(box.getChrs(),
        new Rectangle(box.getRect()), box.getPage
        ());
    newBox.setSelected(true);
    boxes.add(index, newBox);
    Rectangle newRect = newBox.getRect();
    newRect.x += 15; // offset the new box 15
        pixel from the base one
    Object[] newRow = {newBox.getChrs(), newRect
        .x, newRect.y, newRect.width, newRect.
        height};
    tableModel.insertRow(index, newRow);
    jTable.setRowSelectionInterval(index, index)
        ;
    this.jLabelImage.repaint();
    updateSave(true);
}

@Override
void jMenuItemDeleteActionPerformed(java.awt.
    event.ActionEvent evt) {
```

```
  if (boxes == null) {
   return;
  }
  List<TessBox> selected = boxes.
     getSelectedBoxes();
  if (selected.size() <= 0) {
   JOptionPane.showMessageDialog(this, "Please
      select a box or more to delete.");
   return;
  }

  for (TessBox box : selected) {
   int index = this.boxes.toList().indexOf(box
     );
   this.boxes.remove(box);
   tableModel.removeRow(index);
  }

  resetReadout();
   this.jLabelImage.repaint();
   updateSave(true);
  }

  /**
      * @param args the command line arguments
      */
  public static void main(String args[]) {
   java.awt.EventQueue.invokeLater(new Runnable
      () {

    @Override
    public void run() {
     new GuiWithEdit().setVisible(true);
    }
   });
  }
}
```

Path = Tess2SpeechTrainer/jTessBoxEditor/src/net/sourceforge/tessboxeditor/GuiWithSpinner.java

```
/**
 * Copyright @ 20011 Quan Nguyen
 *
 * Licensed under the Apache License, Version
     2.0 (the "License");
 * you may not use this file except in
     compliance with the License.
 * You may obtain a copy of the License at
 *
 *  http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed
     to in writing, software
 * distributed under the License is
     distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY
     KIND, either express or implied.
 * See the License for the specific language
     governing permissions and
 * limitations under the License.
 */
package net.sourceforge.tessboxeditor;

import java.awt.Image;
import java.awt.Rectangle;
import java.awt.image.BufferedImage;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.Icon;
import javax.swing.ImageIcon;
import javax.swing.JSpinner;
import net.sourceforge.tessboxeditor.tessboxeditor.
    components.ImageIconScalable;
import net.sourceforge.tessboxeditor.datamodel
    .TessBox;

public class GuiWithSpinner extends
    GuiWithEdit {
 private final static Logger logger = Logger.
    getLogger(GuiWithSpinner.class.getName())
    ;

 @Override
 void stateChanged(javax.swing.event.
    ChangeEvent evt) {
  if (tableSelectAction) {
   return;
  }
  List<TessBox> selected = boxes.
     getSelectedBoxes();
  if (selected.size() <= 0) {
   return;
  } else if (selected.size() > 1) {
//            JOptionPane.showMessageDialog(
    this, "Select only one box for Spinner
    operation.");
   return;
  }

  TessBox box = selected.get(0);
  int index = this.boxes.toList().indexOf(box)
     ;

  box.setChrs(this.jTextFieldCharacter.getText
     ());
  tableModel.setValueAt(box.getChrs(), index,
     0);
  Rectangle rect = box.getRect();
  JSpinner sp = (JSpinner) evt.getSource();
  if (sp == this.jSpinnerX) {
   rect.x = (Integer) this.jSpinnerX.getValue
     ();
   tableModel.setValueAt(String.valueOf(rect.x
     ), index, 1);
  } else if (sp == this.jSpinnerY) {
   rect.y = (Integer) this.jSpinnerY.getValue
     ();
   tableModel.setValueAt(String.valueOf(rect.y
     ), index, 2);
  } else if (sp == this.jSpinnerW) {
   rect.width = (Integer) this.jSpinnerW.
     getValue();
   tableModel.setValueAt(String.valueOf(rect.
     width), index, 3);
  } else if (sp == this.jSpinnerH) {
   rect.height = (Integer) this.jSpinnerH.
     getValue();
   tableModel.setValueAt(String.valueOf(rect.
     height), index, 4);
  }

  Icon icon = jLabelImage.getIcon();
  try {
   Image subImage = getSubimage((BufferedImage
     ) ((ImageIcon) icon).getImage(), rect);
   ImageIconScalable subIcon = new
     ImageIconScalable(subImage);
   subIcon.setScaledFactor(scaleFactor);
   jLabelSubimage.setIcon(subIcon);
  } catch (Exception e) {
   logger.log(Level.WARNING, e.getMessage(), e
     );
  }
  this.jLabelImage.repaint();
  updateSave(true);
 }

 /**
     * @param args the command line arguments
     */
 public static void main(String args[]) {
  java.awt.EventQueue.invokeLater(new Runnable
     () {

   @Override
   public void run() {
    new GuiWithSpinner().setVisible(true);
   }
  });
 }
}
```

Path = Tess2SpeechTrainer/jTessBoxEditor/src/net/sourceforge/tessboxeditor/

GuiWithValidator.java

```
/**
 * Copyright @ 2013 Quan Nguyen
 *
 * Licensed under the Apache License, Version
     2.0 (the "License"); you may not
 * use this file except in compliance with the
     License. You may obtain a copy of
 * the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed
     to in writing, software
 * distributed under the License is
     distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND,
     either express or implied. See the
 * License for the specific language governing
     permissions and limitations under
 * the License.
 */
package net.sourceforge.tessboxeditor;

import java.awt.Cursor;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;
import javax.swing.SwingWorker;
import javax.swing.filechooser.FileFilter;
import static net.sourceforge.tessboxeditor.
    GuiWithTrainer.DIALOG_TITLE;
import net.sourceforge.vietocr.OCR;
import net.sourceforge.vietocr.OCRFiles;
import net.sourceforge.vietpad.components.
    SimpleFilter;

public class GuiWithValidator extends
    GuiWithTrainer {

 private JFileChooser
    jFileChooserValidatingImage;
 private OcrWorker ocrWorker;

 private final static Logger logger = Logger.
    getLogger(GuiWithValidator.class.getName
    ());

 public GuiWithValidator() {
  initComponents();
 }

 private void initComponents() {

  jFileChooserValidatingImage = new
    JFileChooser();
  jFileChooserValidatingImage.
    setAcceptAllFileFilterUsed(false);
  jFileChooserValidatingImage.
    setApproveButtonText("Select");
  jFileChooserValidatingImage.setDialogTitle("
    Select Image File");
  FileFilter allImagesFilter = new
    SimpleFilter("bmp;jpg;jpeg;png;tif;tiff",
    "All Images");
  jFileChooserValidatingImage.setFileFilter(
    allImagesFilter);

 }

 @Override
 void jButtonValidateActionPerformed(java.awt.
    event.ActionEvent evt) {
  String lang = this.jTextFieldLang.getText();
  File tessdata = new File(trainDataDirectory,
    "tessdata");
  File traineddata = new File(tessdata, lang +
    ".traineddata");
  if (!traineddata.exists()) {
   String message = String.format("%s.
    traineddata does not exist in %s. Be sure
    to run training first.", lang, tessdata.
    getPath());
   JOptionPane.showMessageDialog(this, message
    , DIALOG_TITLE, JOptionPane.ERROR_MESSAGE
    );
   return;
  }

  jFileChooserValidatingImage.
    setCurrentDirectory(trainDataDirectory ==
     null ? null : new File(
    trainDataDirectory));

  // perform OCR on the training image
  if (jFileChooserValidatingImage.
    showOpenDialog(this) == JFileChooser.
    APPROVE_OPTION) {
   jButtonValidate.setEnabled(false);
   this.jTextAreaValidationResult.setText(null
    );
   jLabelStatus.setText(bundle.getString("
    OCR_running..."));
   jProgressBar1.setIndeterminate(true);
   jProgressBar1.setString(bundle.getString("
    OCR_running..."));
   jProgressBar1.setVisible(true);
   getGlassPane().setCursor(Cursor.
    getPredefinedCursor(Cursor.WAIT_CURSOR));
   getGlassPane().setVisible(true);

   File imageFile =
    jFileChooserValidatingImage.
    getSelectedFile();
   List<File> files = new ArrayList<File>();
   files.add(imageFile);

   // instantiate SwingWorker for OCR
   ocrWorker = new OcrWorker(files);
   ocrWorker.execute();
  }
 }

 @Override
 void jButtonCloseDialogActionPerformed(java.
    awt.event.ActionEvent evt) {
  this.jDialogValidationResult.setVisible(
    false);
 }

 /**
     * A worker class for managing OCR process
         .
     */
 class OcrWorker extends SwingWorker<Void,
    String> {

  List<File> files;

  OcrWorker(List<File> files) {
   this.files = files;
  }

  @Override
  protected Void doInBackground() throws
    Exception {
   OCR<File> ocrEngine = new OCRFiles(
    tessDirectory);
   ocrEngine.setDatapath(trainDataDirectory);
   ocrEngine.setLanguage(jTextFieldLang.
    getText());

   for (int i = 0; i < files.size(); i++) {
    if (!isCancelled()) {
     String result = ocrEngine.recognizeText(
    files.subList(i, i + 1));
     publish(result); // interim result
    }
   }

   return null;
  }

  @Override
  protected void process(List<String> results)
     {
   for (String str : results) {
    jTextAreaValidationResult.append(str);
    jTextAreaValidationResult.setCaretPosition
    (jTextAreaValidationResult.getDocument().
    getLength());
   }
  }

  @Override
  protected void done() {
   jProgressBar1.setIndeterminate(false);

   try {
    get(); // dummy method
    jLabelStatus.setText(bundle.getString("
    OCR_completed."));
    jProgressBar1.setString(bundle.getString("
```

```
 OCR_completed."));
} catch (InterruptedException ignore) {
 logger.log(Level.WARNING, ignore.
  getMessage(), ignore);
} catch (java.util.concurrent.
  ExecutionException e) {
 String why;
 Throwable cause = e.getCause();
 if (cause != null) {
  if (cause instanceof IOException) {
   why = bundle.getString("
  Cannot_find_Tesseract.
  _Please_set_its_path.");
  } else if (cause instanceof
  FileNotFoundException) {
   why = bundle.getString("
  An_exception_occurred_in_Tesseract_engine_while_recognizing_this_image
  .");
  } else {
   why = cause.getMessage();
  }
 } else {
  why = e.getMessage();
 }

 logger.log(Level.SEVERE, why, e);
 jLabelStatus.setText(null);
 jProgressBar1.setString(null);
 JOptionPane.showMessageDialog(null, why,
  DIALOG_TITLE, JOptionPane.ERROR_MESSAGE);
} catch (java.util.concurrent.
  CancellationException e) {
 logger.log(Level.WARNING, e.getMessage(),
  e);
 jLabelStatus.setText("OCR " + bundle.
  getString("canceled"));
 jProgressBar1.setString("OCR " + bundle.
  getString("canceled"));
} finally {
 getGlassPane().setCursor(Cursor.
  getPredefinedCursor(Cursor.DEFAULT_CURSOR
  ));
 getGlassPane().setVisible(false);
 jButtonValidate.setEnabled(true);
 jDialogValidationResult.setVisible(true);
}
}
}

/**
 * @param args the command line arguments
 */
public static void main(String[] args) {

 java.awt.EventQueue.invokeLater(new Runnable
    () {

  @Override
  public void run() {
   new GuiWithValidator().setVisible(true);
   Gui.finished = true;
  }
 });
}
}
```

XI. Acknowledgment

Woooh! Sa wakas! Ito na ang part na inaabangan kong isulat dahil bukod sa pwede na akong mag-tagalog dito, ibigsabihin na rin nito ay natapos ko na "Special Problem" ko.

Unang-una sa lahat, gusto kong pasalamatan ang aking pamilya dahil sa walang sawang pagsuporta nila sa akin habang ginagawa ko ang thesis ko. Lahat ng mga bagay na kailangan ko ay ibinibigay nila kahit medyo marami at medyo magastos. Gusto ko rin magpasalamat sa kanila dahil sa tiwalang ibinigay nila sa akin. Kahit isang beses ay hindi nila pinagdudahan ang kakayahan kong tapusin ang thesis ko, kahit minsan ako mismo ang nagdududa sa sarili ko kung kaya ko ba talagang tapusin 'to.

Gusto ko rin pasalamatan si Sir Greg Baes dahil siya ang nag-guide sa akin upang maging maayos at magkaroon ng direksyon ang thesis ko. Kahit medyo hassle pumunta sa school niya kapag magpapaconsult ka, medyo harsh magsalita (lol), at maraming pinapagawa, alam ko namang ginagawa niya ang lahat ng iyon para sa ikabubuti ng thesis ko. Gusto ko rin magpasalamat sa pagbibigay niya ng tiwala sa akin na kahit next week pa dapat talaga ako magde-defense, pinag-defense na niya ako ng araw na iyon. Medyo nakakabigla dahil hindi pa talaga ako prepared noon at hindi ko pa talaga nade-debug yung app ko, pero nagtiwala siya na okay na akong magdefense. Maraming salamat po at sorry din po sir sa abala sa schedule ninyo minsan.

Salamat din kay deep learning lord Sir Marvin Ignacio na kahit hindi namin siya adviser, hinahayaan pa rin niya kami na magpa-consult sa kanya. Siya ang napapagtanungan ko ng mga bagay na related sa Machine Learning at training algorithms. Maraming salamat ulit sir! LEARNING!

Siyempre gusto ko rin pasalamatan ang 2012 Block 12 B.S. Computer Science students dahil kahit na shiftee ako, tinanggap pa rin nila ako. Medyo late ko na nga lang sila nakakausap at nakakabiruan dahil mahiyain talaga ako kapag hindi ko pa kilala ang mga kasama ko. Sorry medyo late ko na kayo naka-close

pero kahit ganoon, naging masaya pa rin naman ako sa mga kalokohan natin sa maiksing panahon na 'yon. Gusto kong sabihin na masayang kasama ang block niyo. Salamat din sa pagbigay niyo ng tiwala sa akin na kaya kong tapusin ang thesis ko. Salamat sa pagsabi sa akin palagi ng "Kaya mo 'yan, ikaw pa!" at pag-pressure sakin na i-meet ang expectations niyo. Dahil doon, nagawa ko talagang ibigay ang lahat ng kakayahan ko upang matapos ang thesis na ito.

Salamat U.P. Manila sa limang taon na pagpapa-realize sa akin na marami pa akong dapat matutunan. Salamat sa pag-challenge sa akin at pagpapakita sa akin na hindi sapat na matalino ka noong highschool, dahil pagdating mo sa U.P., doon masusubok kung gaano ka talagang katalino. Pero 'di pala talaga ako ganoong katalino tulad ng iniisip ko. Looool!

Syempre makakalimutan ko ba na pasalamatan ang pinaka-tumulong sa akin? Siya ang nandiyan sa tuwing may problema ako. Maraming salamat Stack Over-flow!! Hindi ko ito matatapos ng wala ka. You da real MVP!

Inaalay ko sa inyong lahat ang thesis na ito. Kung wala kayo, hindi ko ito matatapos. Maraming maraming salamat talaga sa pagbibigay ninyo ng tiwala at suporta sa akin. Good Bye U.P.?