Smart Glasses for the Visually Impaired People

By

Esra Ali Hassan

17879

Dissertation submitted in partial fulfilment of

the requirements for the

Bachelor of Engineering (Hons)

(Electrical & Electronics)

JANUARY 2016

Universiti Teknologi PETRONAS

Bandar Seri Iskandar

32610 Tronoh

Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Smart Glasses for the Visually Impaired People

By

Esra Ali Hassan

17874

A project dissertation submitted to the
Electrical & Electronic Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
BACHELOR OF ENGINEERING (Hons)
(Electrical & Electronics)

Approved By,

____

(AP. Dr. Tang Tong Boon)

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
January 2016

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

ESRA ALI HASSAN

ABSTRACT

Engineering solutions have involved in everyone's life, most importantly are those aiming to help people with disabilities, however, the modern assistance devices with their current prices are not meeting the requirements of the market. This project is mainly focusing on people with visual impairments and more specifically their education life. It is presenting a concept of smart glasses to provide assistance in multiple tasks represented as modes to be chosen by the user. To prove the concept, this project implements only one mode which is reading using text detection techniques. Taking into consideration the cost, this project is using the single board computer raspberry pi 2 as the heart of the processing and the raspberry pi camera for image capturing and video recording. The video taken is processed using MATLAB, and the description of the live scenes for the text recognition mode or any other future implemented modes will be provided to the user in an audio format. Finally the results of this project show the design of the smart glasses prototype. The implementation process and techniques of the text recognition mode is discussed, an experiment is also conducted to compute the accuracy of the system when changing the font size and style, the results showed that the accuracy has a dependency on the font style (Arial giving the best results), and most importantly the font size; the bigger the size the better results are obtained. This project is using the advantage of the wearable devices and the capabilities of the new raspberry pi therefore it can be taken further to provide assistance in more tasks.

ACKNOWLEDGMENT

I would like to express my deepest appreciation to all those who provided me the possibility to complete this project. I take this opportunity to express my profound gratitude and deep regards to my supervisor Dr. Tang Tong boon for his exemplary guidance, monitoring and constant encouragement throughout this project.

I am obliged to Mr. Mohaned Essam for the valuable information provided by him, am grateful for his cooperation during the implementation of this project.

Last but not least, I thank almighty, my parents, brother, sisters and friends for their constant encouragement without which this project would not be possible.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Background of the project

Nowadays, engineering solutions are improving our daily life in many ways some of which we do not even notice. Of the most important improvements are those aiming to help people with disabilities to overcome their challenges, and to cope with the changing environment around them.

As quoted from the medical dictionary "Total blindness is the inability to tell light from dark, or the total inability to see. Visual impairment or low vision is a severe reduction in vision that cannot be corrected with standard glasses or contact lenses and reduces a person's ability to function at certain or all tasks " [1]. Visual impairment as classified according to [2] includes low vision acuity, blindness, legal blindness which means having a very low vision that is legally considered as blindness, and moderate visual impairment. Visually impaired people are growing over the past decades with different age groups. As reported by the world health organization WHO at August 2014, 285 million people worldwide are estimated to be visually impaired, 39 million of them are blind and 246 million have low vision [3]. Losing the sight or vision means a lot of difficulties communicating with others as well as developing the level of knowledge and experience. Visually impaired tend to be suited into special classes and they are treated in special ways which in many cases have resulted in isolating them from the society and preventing them from interacting with others and accessing the amount of information accessed by a normal person.

The impact of the visual impairments on the affected people differs from one person to another, generally, it reduces the life quality and independence which in return might cause depression and isolation in some cases [4]. The abilities of the visually impaired people to do many tasks are limited. As a result many schools and jobs cannot accommodate them because of the high cost associated with their special needs; this contributes to the high unemployment rate among them. In the United States for example the blind unemployment rate reaches 75% [5]. Unemployment results in low level of income as quoted from WHO statistics in 2014 “ About 90% of the world's visually impaired live in low-income settings “ [3].

On the other hand the new aids and technologies developed for the visually impaired are considered expensive at an average price range from $200 - $3000 or higher as found on EnableMart online website. The price depends on the task the device or the software is performing while most of them are designed for single task usage, such as the software programs specialized on only PDF reading ($200) [6]. Another example is Esight which is one of the new commercialized multiple tasks with a price of $15000 as quoted from their website [7].

For all the information stated, visually impaired people are a subject of study to many researchers who are trying to ease their lives in many different ways. This project presents a new concept of glasses that is believed to help the visually impaired people and widen their scope using a description of the live scenes in front of them, the scenes are classified to a mode format that can be changed when needed by the users, for example navigation, reading, face recognition and so on. To prove the concept and because of the time limitation, the project only focuses on implementing one mode which is text recognition to provide assistance in reading physical materials, the glasses is designed with the low cost single board computer raspberry pi 2, and the image processing techniques are implemented using SIMULINK models.

1.1.1 Raspberry pi 2:

Raspberry pi is a low cost single board computer that is capable of doing everything expected from a desktop computer, the idea behind creating the raspberry pi came in 2006 with the intention to encourage children to learn programming and how computers work [8]. Years later with the advanced versions of the raspberry pi, not only children but also adults, researchers and students started to realize and to engage this tiny computer into bigger projects. Because of the affordability and the capability of the raspberry pi, it has been the suitable device for many applications.

Raspberry pi 2 B is the second generation of the raspberry pi, it is used in this project as the heart of the processing because of the affordable price, and the acceptable capabilities with the quad core ARM7 processor and the 1GB RAM [8], this in return gives a better performance in image processing applications compared to previous versions.

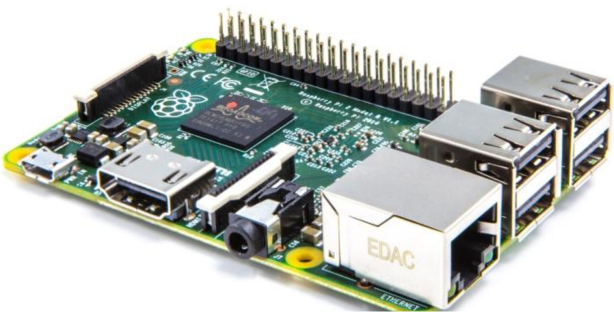


Figure 1: Raspberry pi 2

1.1.2 Simulink modelling:

Simulink, software owned by MathWork, is a block diagram and model based design software suitable for system design, modelling and simulation. Simulink has a great range of capabilities that enable the user to translate the algorithm into a model and focus on the overall system. Simulink provides automatic code generation, blocks libraries and real time continuous testing and verification methods [9].

Simulink support package for raspberry pi enables the user to build models that can run as a standalone in the raspberry pi, the package includes blocks to interact with the camera, audio port, GPIO, and other features of the raspberry pi.

Simulink is also integrated with Matlab to enable more analysis and more functions capabilities, in this project both Simulink and Matlab are used to complete the system.

1.2 Problem Statement

- Visual impairments limit the ways the person can interact with others, access information or develop his/her own knowledge and experience, therefore a need for an assisting aids with multitasks feature to cope with different situations is an important issue.
- The available aids and technologies in the market nowadays are expensive to the normal or low level of income people which are the majority of the users, therefore new devices with similar tasks and cheaper prices are needed.

1.3 Objectives and scope of study:

The scope of this project is to help visually impaired students. The concept of the glasses is to assist in multiple tasks. Because of the time limitation this project only demonstrates one mode which is text recognition as a prove of concept. The project prototype includes a custom designed border platform for text recognition, a pair of glasses with a camera and earpiece, connected to the single board computer the raspberry pi 2. Matlab package for raspberry pi is used to implement the modelling and coding of the image processing.

The objectives of the project:

- To design and implement smart glasses that could be used by the visually impaired people easily, with an emphasis on cost-effectiveness.
- To prove the feasibility of the image processing techniques with the audio description as a tool helping the visually impaired people and giving them more independence in their education life.

The following chapters in this report include:

- Chapter 2: Literature review, and technical methods theory.
- Chapter 3: Includes the research methodology, the proposed technical steps, and the tools and components.
- Chapter 4: Discusses the technical results and analysis.
- Chapter 5: Concludes all of the previous sections with suggested recommendations and future work.

CHAPTER 2

LITERATURE REVIEW

In this chapter, a review of past work and research papers related to the project is examined. The project looks at many areas, at first the needs and expectation of the visually impaired people are reviewed, followed by a discussion about some examples of the available and suggested aids and technologies to assist the visually impaired in their lives, after that a technical section that is important for the scope of the project is reviewed which is text recognition.

1- The needs of the visually impaired people

   In order to design an assistive device for the visually impaired, information about their needs and their expectations should be gathered for a better design, it is easy to understand that the main problem is the sight loss, but to explore the situation without sight and to identify their needs is not an easy task for the sighted people. Many needs have been identified by many papers including papers written by visually impaired people to help the researchers to understand the issues better. In [10] The needs of the visually impaired are discussed by observing the type of information a person would require in the daily life, generally people do not ask for information but they absorb it from the surroundings and use it to play certain rule, as difficult as it seems to classify the information needs for visually impaired; the paper has identified some types of information needs for a defined task which are: the function of the task, the form in which we find things related to the task, the clusters they belong to, the agents who initiate it, the users and finally the information about the mechanism used to find things. The depth of the details required by the visually impaired is also important, a paper written by one of the visually impaired people called pictures into words [11] discussed the blind imagination, and that the amount of information needed actually depends on the personality of the person and his history, people who were having normal vision for a long time might be eager to know more details than those who were born without vision.

The need of education is a major need for every human being; it is a basic human right that is contributing toward the development of each and every one of us. Inclusive education is a case study that has been researched a lot lately it means that “all students attend and are welcomed by their neighbourhood schools in age-appropriate, regular classes and are supported to learn, contribute and participate in all aspects of the life of the school.” [12]. According to [13] UNESCO has counted some of the problems that are blocking inclusive education from being applied nowadays, the cultural and social background that affect the perception about people with disabilities and therefore negatively affecting the visually impaired and leading to isolation, also classifying students depending on their type of disability in certain groups with certain amount of information and tools which limit their education level and therefore their ability to develop their best in term of career and social life which result in low level of income. Those problems can be eliminated by the use of suitable technology in the modern education and the inclusive education techniques which until now are not sufficiently applied.

2- Assisting aids and Technologies

Technology has changed how people live nowadays, it is almost in everything around us, developing an assisting aids for people with disabilities were introduced long time ago, it is fast growing with the recent technology available nowadays, for the visually impaired it starts from the simple glasses, contact lenses going up to magnifiers and more advanced readers or navigation systems.

Some of the recent technology aids as listed by [4] are the audio books which is a saved audio format for certain books or newspapers provided by certain suppliers, e-book reader and screen readers are used to read the digital format of books or website contents from a screen, electronic magnifiers for those who are not completely blind, and the digital recorders to help in taking notes.

Other research papers have focused on some basic needs one of them is the navigation. Performing a routine or normal task for people with no sight is something difficult without the assistance of others in navigation and obstacle avoidance, PERCEPT system discussed in [14] is an invented system for navigation which mainly integrating gloves with a smart phone for the user side, and a building that is equipped with percept system which is the R-tags or the RFID tags distributed in the building and helping the user to navigate to his destination in that specific building. Another system also used the Infrared tags installed on the buildings walls to help in the indoor navigation as discussed by Jain [15] ; the system is mainly developed as a path finder for the visually impaired by making use of their smart phones, an application is used along with the user module that is consisting of an IR receiver sensor and accelerometer to accurately locate the user and update the path, the system also used text to speech synthesizer to communicate the information to the user ear.

Other non-commercialized products are also available under testing for other purposes. As listed by [16] the touch colour which is a device that use the temperature to reflect the colours to the user was introduced, each colour has abbreviation version of Braille, it helps in creating the colour wanted by the user through painting the selected colour by selecting its temperature or in identifying the colours captured by the equipped camera.

Finally through all those new and recent technologies, the Eyewear based wearable devices arises as the best substitutions, the potential of this new design and how they can best serve the visually impaired people was discussed by Jafri and Ali [17] They mentioned some of their advantages which are: the weight of the device is light with natural look, it offers a hands-free access to information through the connected computer, it is naturally looking at the viewing direction to eliminate the direction instructions used in other devices and the most important fact is that it can implement many tasks of many devices in a single wearable device.

One of the integrated systems supporting the eyewear based devices is called expression [18], it is using google glasses as a design base, and implementing an algorithm to identify the social expression captured by the camera installed on the glasses and comparing it with the expressions on its database, therefore it can tell the user if you are smiling to him, yawning or any other stored expressions. OpenShades and Memento are two other developed applications for google glass, and OrCam are other commercially released glasses for multiple tasks [17].

a

b

c

Figure 2: eyewear based wearable assisting aid [17] , [7] : a: Google glass , b: esight , c: Orcam

Figure 3: Screenreader

For the purpose of this project, the devices concerned about the reading tasks are compared in table 1 from the previous literature review discussion. The comparison includes two categories depending on the number of tasks performed, the first category Braille readers, audio books ad screen readers for a single or few tasks, the second category includes OrCam and Esight as representation of multiple tasks wearable devices.

Table 1. Comparison between proposed design and available assistive devices

Device	Functionality & No. of tasks	Price	Remarks
Braille readers [5]	Reading and writing. (Two tasks)	$1000-$3000	Only support tactile materials
Audio books [4]	Reading (One task)	$25 per month	Only for certain available books.
Screen reader [4]	Reading digital format (One task)	$150 - $1000	Only for digital content
OrCam [19]	Multi task- includes reading.	$2,500	Unaffordable price
Esight [7]	Re-display the live scenes for visually impaired to see. (Multitask)	$15,000	Unaffordable price, and only for people with low vision, totally blind people can't benefit from it.
Proposed design	Proposed as reading. Has the capacity to be Multitask	$100-$150	Limited by the performance and accuracy of the hardware.

3- Text Recognition:

Text recognition is a critical research problem that is under the study until today. For humans it is easy to recognize text but for computer based system it is a complicated process, text recognition can be classified into: Hand written texts and typed text. Some research have classified the hand written text recognition into online and offline methods, online methods are meant to recognize the text while it is written in real time which is more difficult because it is considering the time of the process as a major factor with the accuracy, some of the online methods are: the direction based algorithm which identify between different characters by observing the direction of writing them, and the KNN classifier which compare between the input and pre trained data. The offline methods are for the static images that include a text where time is not considered as major factor, many methods are introduced for this type some of them named as clustering, Future extraction, Pattern matching, and artificial neural network method [20].

The most popular and the one needed for this project is to recognize the typed text format, optical character recognition OCR is a very famous technique defined as “the mechanical or electronic translation of images in written or printed text“ [21]. OCR technique is used by many and each has his different method of implementing it, one of the methods is to extract the text from the image and save it in separate file, the separated text will then go under pre-processing step which contains binarization to convert the image into gray-scale then the isolated specs and holes into characters are removed followed by segmentation process, after that the features will be extracted from the processed data such as the skewness and kurtosis, those features will help in classifying the data [22]. Two famous methods mentioned by other research paper to implement the OCR techniques are the correlation approach and neural net approach, for the correlation approach we can use either the Fourier domain for filtering the matched pattern or the spatial domain, on the other hand for the neural net approach this approach is a very interesting method trying to simulate the way the human brain is functioning, as for humans the cognitive tasks are easy compared to its difficulty for a computer because of the brain neurons function, following the same technique the ANN method is implementing a network of neurons taking many inputs and connecting their outputs with a learning rule [23].

Before implementing the OCR usually a pre-processing steps are carried out as discussed in [24], when digital cameras are used to acquire images in preparation for text recognition, they introduce some noise as well as uncertainty in the form of the images, in [24] three important pre-processing steps were discussed which are the autorotation, perspective correction as some images might be taken while the camera is not orthogonally to the paper sheet. And finally the non-linear image transformation that appears in the images taken from large books where the text at the middle appears smaller and sometimes not clear to the OCR to identify it.

A text to speech synthesizer is a system that is capable of reading the text loud as an audio format when it is received by the computer [21]. There are many types of TTS techniques each one has its advantages and disadvantages all of them are trying to deliver the speech in a native sound as close as possible. Many steps are required in to convert the text into speech which can be divided into two, the front end steps which are related to the language processing, the second part is the back end steps which related to speech signal processing [25].

Summary

This chapter has reviewed the past work related to the scope of this project. The first part discussed the needs of the visually impaired people, it can been concluded that it is difficult to specify certain type of needs with certain way of solutions, because losing the sight is more complicated to be analyzed, however, when going more specifically at the end of this part, the education need was specially discussed with the importance of inclusive education concept which is not yet implemented, one of the problems is the absence of suitable assisting aids that will enable the visually impaired to learn at the same environment with the sighted people, this can be supported by the facts and results discussed earlier in the first point of the problem statement and the background study about the schools and jobs accommodation problems that affect their economic life. The second part discussed the up to date assisting aids and technologies developed for the visually impaired, many aids are now available in the market, however, from the discussion a conclusion can be made that some limitations are still there, one of them is the fact that each device perform one specialized task making it inconvenient for the users when combined with its price, this strengthen the second point of the problem statement specially when reviewing the comparison table that shows examples of the available aids, their current prices and most importantly the number of tasks they are providing. On another side the potential of the wearable devices and specially the eyewear based devices seems to be promising for the overall advantages over the existing devices, taking this into consideration supports the idea of developing more wearable devices. The last part has discussed about text recognition, a very popular topic that is successfully implemented with different error percentages, the popular technique used is the OCR, a review of the OCR implementation methods, steps, and pre-processing techniques have been analyzed from different papers, each technique has its advantage and disadvantages, the advantages however made each technique better than the others in certain situations. Since this project is focusing on developing a reading mode using text recognition, those steps and techniques discussed are combined and analyzed to come up with a suitable implementation methodology based on the requirements of this project which is illustrated in more details in the following chapter.

CHAPTER 3

METHODOLOGY

3.1 Research Methodology

The following steps show the project methodology in order to reach its objectives.

1- Problem Identification: to define the problem preciously a lot of pre-searching was done on the original idea before reaching to the scope of study and the objectives to be achieved that is defined in the objectives part.
2- Literature review: After identifying the exact problem to be solved, a searching process is done to the past papers, conferences, books, articles and any other material that was studying either the whole problem or a part of the problem, by that the needs of the visually impaired is more defined to be considered in the design of the project and to define the problem statement more clearly, the previous aids and technologies are revised, and the area of text recognition is studied to help in implementing the specified reading mode.
3- Setting up requirements and components: After doing the literature review the requirements of the system are chosen depending on the finding of the previous work and the chosen scope. Setting the requirements lead to identifying the suitable components which are explained in the tools and components part.
4- Project technical Methodology: The components and software are identified, the algorithm and methods are formalized from the previous research and the information about the tools and components used, and new improvements in the method are implemented from time to time for better performance.
5- Design phase: started from the simple execution of the first testing codes to the simulation process using MATLAB, followed by designing the same algorithm model in SIMULINK, finally the design phase ends with the final prototype design and code implementation.

6- Testing phase: After completing the prototype, it will be under a testing procedure to test if it functions as the same expected ways, if any errors found we return back to the design phase, troubleshoot the problem and modify accordingly.

7- Verification and final results: after successfully passing through the testing phase, the design is verified by experiments to test its functionality and accuracy for further analysis, the final results are displayed and it is then verified by the supervisor.

3.2 Project Components

The project main components are shown in figure 4:

1- Raspberry pi 2 B single board computer: the heart of the processing, executing the program and integrating the inputs and outputs of the system.

2- Raspberry pi camera: for image acquisition, it is a 5 megapixel camera with resolution 2592 x 1944 pixels connected to the raspberry pi via ribbon cable.

3- Matlab & simulink for programming and modelling.

4- Earpiece: to deliver the audio output to the user.

Figure 4: Initial components

3.3 Overall system design

As mentioned previously, the main objective of the design is to prove the concept of the smart glasses as a multi-task assistive tool for the visually impaired people by utilizing the benefits of the eyewear design as discussed in the literature review. The idea to implement this design is to use the raspberry pi to include all the models of the different modes that is available, the user chooses the desired mode to run through an input to the raspberry pi GPIO port which in this case are switches as shown in figure 5, the main program in the raspberry pi will then run the respective simulink model according to the chosen input, the model will interact with the user through the audio output to give feedback or instructions, and to provide the final description output, the audio is received by the user using an earpiece, figure 5 shows a simplified sketch of the design concept.

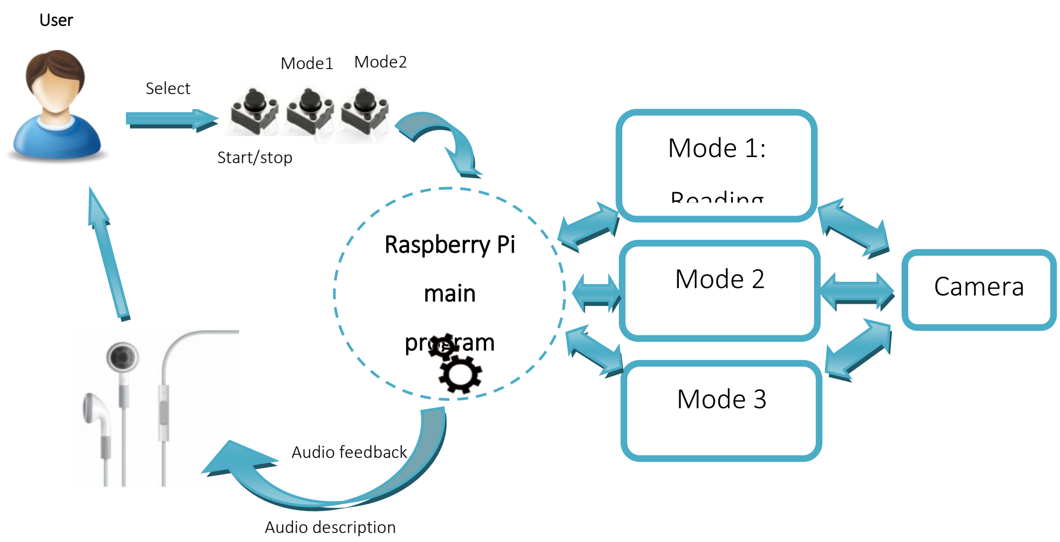


Figure 5: Overall concept

From the discussed overall design concept, this project is implementing the reading mode which is designed to read a physical text material; the specification of the model is the following:

1- The model is implemented to detect text within red boundaries; therefore the material should be enclosed in a custom designed red borders.
2- The material is assumed to have clear content for the model to provide good quality output.
3- The material orientation is important; therefore the model will try to help the user to correct the position if it couldn't detect one or more of the border lines, also if the angle of rotation is very big.

The flow chart in figure 6 is explaining the overall process and the working mechanism.

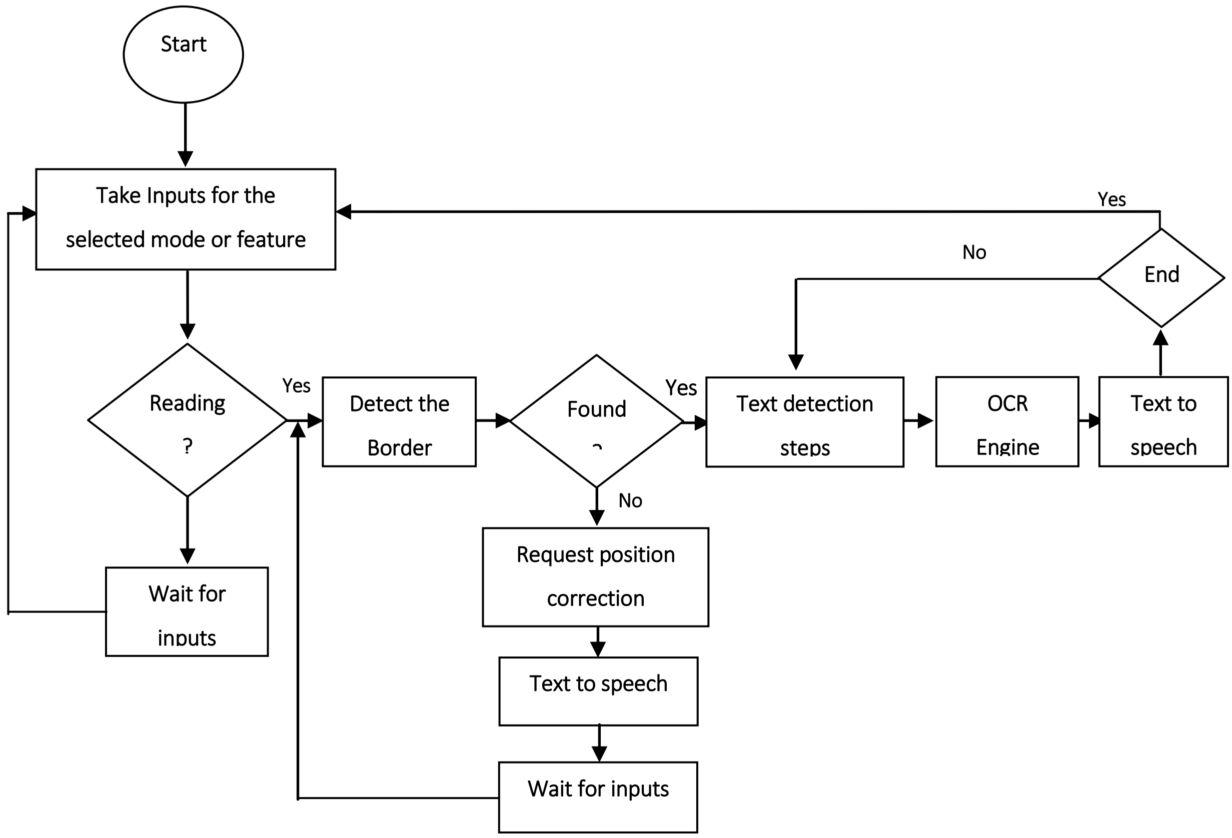


Figure 6: Working Mechanism flow chart

3.4 Design implementation

The reading mode has some major implementation check points; the following discusses these check points in steps:

1- The reading mode starts when the user press of the corresponding push button, the main program will confirm with the user first that it is going to perform text reading, and asks the user to prepare the material and to press another push button when it is ready to make sure a more stabilized video is taken. The interaction is done using the audio output

2- Image acquisition: The main program call the Simulink reading model, image acquisition is done using the raspberry pi camera in the form of video using the V4L2 video capture block from Simulink support package for raspberry pi. The image size, sampling time, and output format are configured in this block.

3- As specified earlier the text localization is performed with the help of red borders, the following steps are used to detect the border:

    a- Convert the image to grayscale image for better processing.

    b- Compute the red component, a colour thresholding procedure is implemented using image subtraction as followed:

       Assuming (Original) is the original RGB image. And (Gray) is the grayscale image.

       First subtracting (Gray) from the red dimension of Original

       IRed=Original(:,:,1)-Gray

       The same subtraction is repeated to the Green and blue dimensions.

       Finally to find only the red component we subtract all the other components from the IRed:

       I=IRed-Igreen.

       Ifinal=I-Iblue.

       The final image will only show the red colour component of the original image.

    c- Image binarization as a preparation for hough transform.

    d- Perform hough transform to find the lines in the image: hough transform is an image processing technique used to find shapes including lines and circles in the binary images. It is depending on the following equation:

$$x\cos\theta + y\sin\theta = r \tag{1}$$

Where r (Rho) is the distance from the reference point, Theta is the angle with the respect to X-axis. Hough transform find the lines depending on the concept that for any given (x,y) point on the same line the values of r and theta will be equal.

Figure 7 shows the general concept, the first graph is showing the image lines, the points identified and the theta and rho for line 1,2 and 3, the second graph is the rho values against the theta values, the intersection shows the points in the same line.

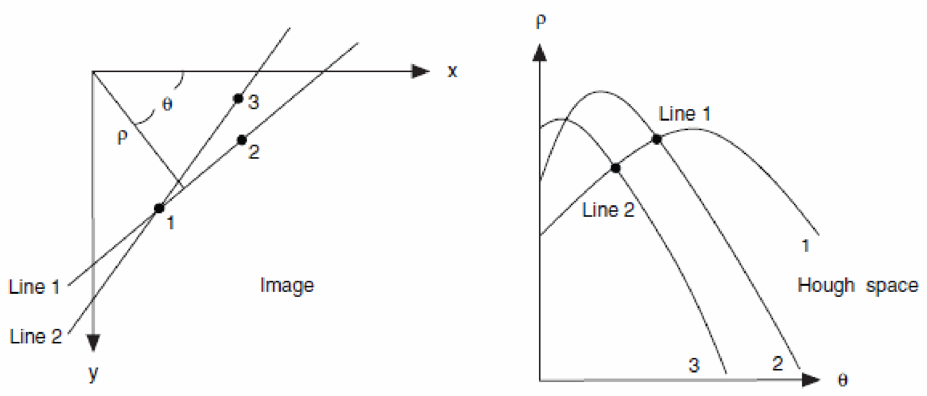


Figure 7: Hough transform concept.

e- To identify the four borders, the lines found in hough transform are used, for that a user defined matlab function is included in the simulink model, the function takes the rho and theta values from the hough transform and compute the intersection points between the lines using the line equation parameters by taking advantage of the starting and ending points in from the Hough lines input. The function confirms if the four lines are found, and compute the angle of rotation if required. The output will be the four lines flags, the points of intersection defining the image corners, and the angle of rotation.

f- The second user defined matlab function takes the inputs of the position from the later function, if the position is not confirmed; it will stop the simulation and gives a feedback to correct the position of the material.

4- Image cropping: if the position and angle are confirmed to be correct, the image will be cropped according to the intersection points found.

5- Image enhancement step: as the image acquisition is using a digital camera, some noise will be present, the noise type and procedures differ from camera to another and from one environment to another environment, matlab analysis is done to the images taken from the raspberry pi camera to find the best procedure that will reduce the noise and improve the image. In the discussion chapter the analysis is further explained.

6- Background subtraction: this step is to subtract the text from the background by binarizing the image using a threshold.

7- Saving the final image: Since the procedure is all done in simulink model, and the OCR engine will be called from the main program, the simulink model final image has to be written back to a file in the raspberry pi in order to be converted to text, the conversion from image matrix to image format required a format encoding, a function is implemented to create the image file and to encode the image using the specification of the ppm image format.

8- The main program use the system command line to call the tesseract OCR software to convert the final image to text and save it, tesseract is an open source optical character recognition that has been installed in the raspberry pi, the command line to perform this step is:

Tesseract finalimage.ppm Textout

9- Finally, this step is to deliver the text read by the OCR to the user ear using a text to speech technique and an earpiece, the main program takes the result of the OCR and passes it to Espeak software to convert the text to an audio format to be delivered to the user. The command line to perform this step is:

espeak -v english-us --stdout -f textout.txt -s 140 | aplay -f cd

The OCR steps are fairly similar in most of the applications. The difference is found in the techniques used to perform each step. For this application, figure 8 shows the discussed steps in simplified blocks for a better understanding of the process.

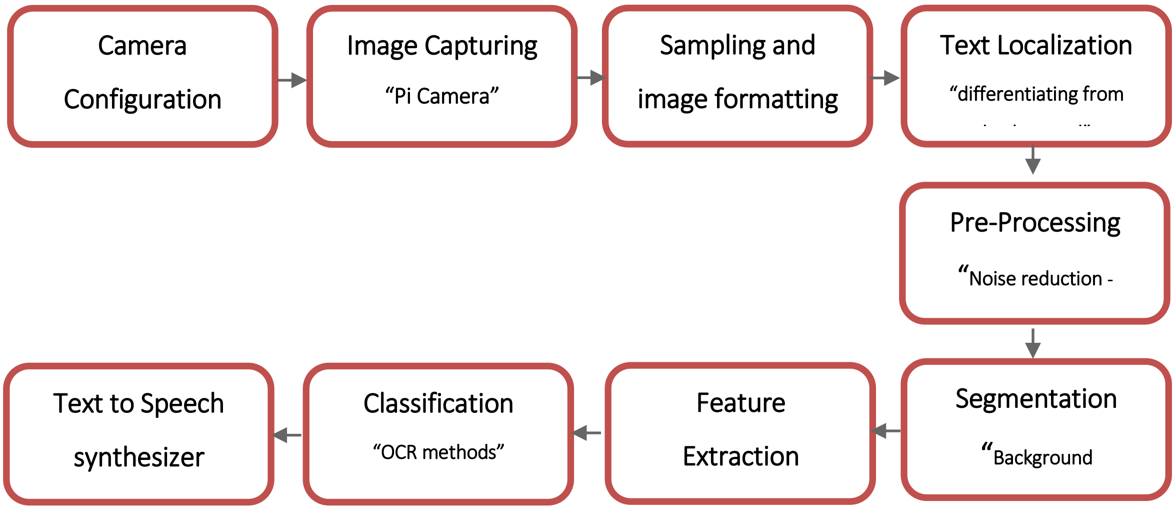


Figure 8: Text Recognition Steps

Summary

In this chapter, the research methodology has been discussed, and the overall system design and process has been illustrated to show the big picture of the system and the glasses design concept. The text recognition design implementation has been explained in details to show the process of the designed system.

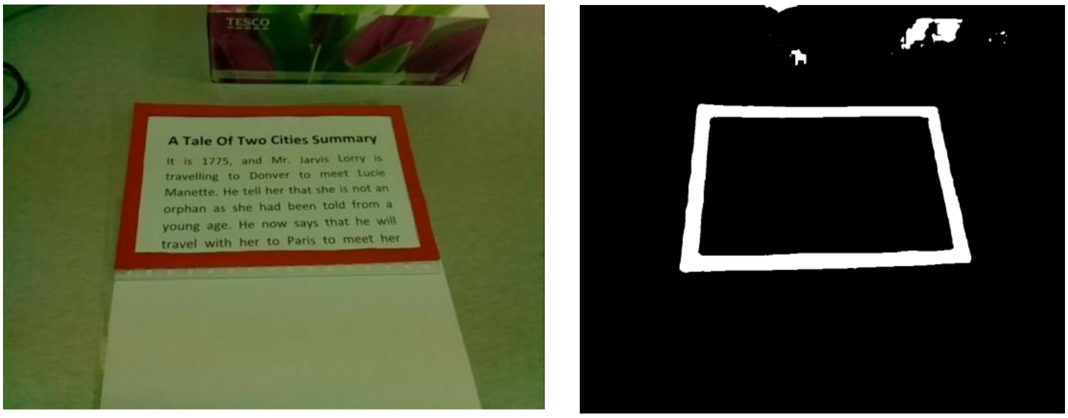
CHAPTER 4

RESULTS AND DISCUSSION

This chapter presents the project results, the discussion and analysis associated with them according to the steps mentioned in the methodology part. The chapter is partitioned into separate points to demonstrate the implementation and analysis of the steps mentioned in the methodology part. The last section of the chapter shows the accuracy test of the system performance and an analysis on the results obtained.

1- Colour thresholding

Colour thresholding is a segmentation method based on the colours present in the image, since we already specified the colour of the border to be red, the colour thresholding method is used to identify the red components so that we can identify the border from other background objects that will help in localizing the text area, as shown in figure 9 (a), the original picture contains the text material with its border and the background, the result of colour thresholding followed by binarization as shown in figure 9 (b) gives a better focus on the text area only.



a b

Figure 9: Colour thresholding. (a) the original image, and (b) red component found by color thresholding

2- Hough transform

Hough transform function takes a binary image as an input and gives the output of the rho and theta values in a matrix format. The number of lines to be found is then configured in hough line block, to find the lines with maximum length, a local maxima block is configured after the hough lines. Figure 10 (a) shows the lines found by the hough transform according to the lines having the peak values of similar rho and theta which means the longer lines as shown in white dots in figure 10 (b).

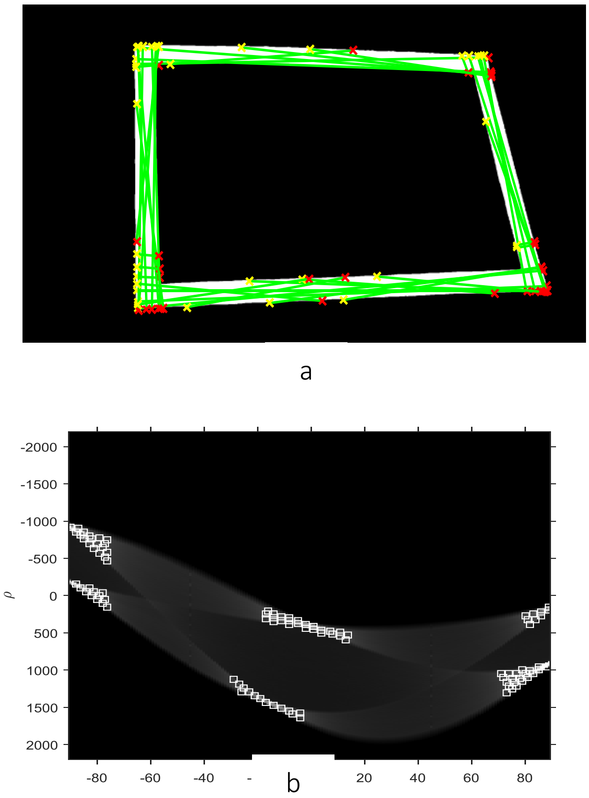
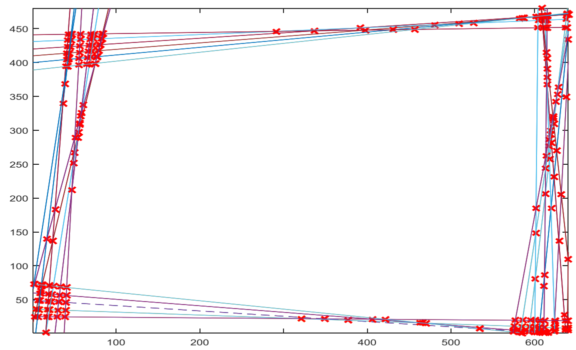


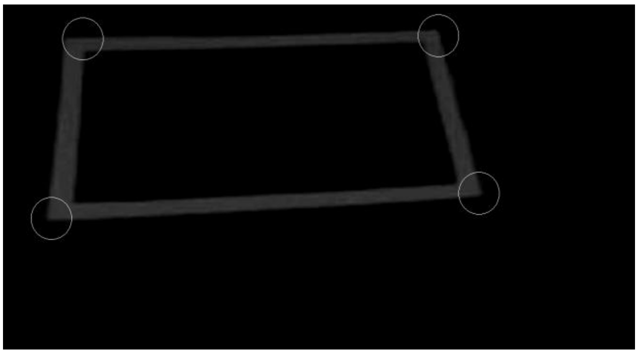
Figure 10: the Hough transform: (a) the lines with starting and ending points, (b): Rho vs Theta

3- Border detection

From the hough transform block, the function created for border detection uses each line start and end (x, y) values to find its equation, using this equation the intersection between the lines can be found, the image is divided into four parts and the intersection found in each part is compared to the expected specification to find the corner of the border, figure 11 (a) shows the intersection points, figure 11(b) shows the corner points identified.



a



b

Figure 11: border detection, (a): the intersection between the lines drawn in red x symbols, (b): the final 4 intersection points found by the border detection

4- Position confirmation

The position confirmation step is added to confirm that all borders are present in the image, the border detection step is responsible of finding the corners for the image cropping, however before proceeding with the rest of the steps, a confirmation of the presence of all the 4 lines in the border is important, the function uses the lines input from the hough transform to confirm the find out if there are lines in the four direction of the image as expected from the border, this function output is a an array of four elements each element is corresponding to one of the border lines, if all of them are detected the function return four ones array, if one of them is missing a feedback text will be written to the raspberry pi file to be converted to audio so that the user can correct the position, figure 12 (a) shows the result of the border detection without position confirmation function, the corner are still identified although the bottom line of the border doesn’t appear in the image. Figure 12 (b) shows the text output sample when there are unfound border lines. Figure 12

(c) shows the image division, the corner flags, and the lines flags, this figure works as reference shape for the image border.

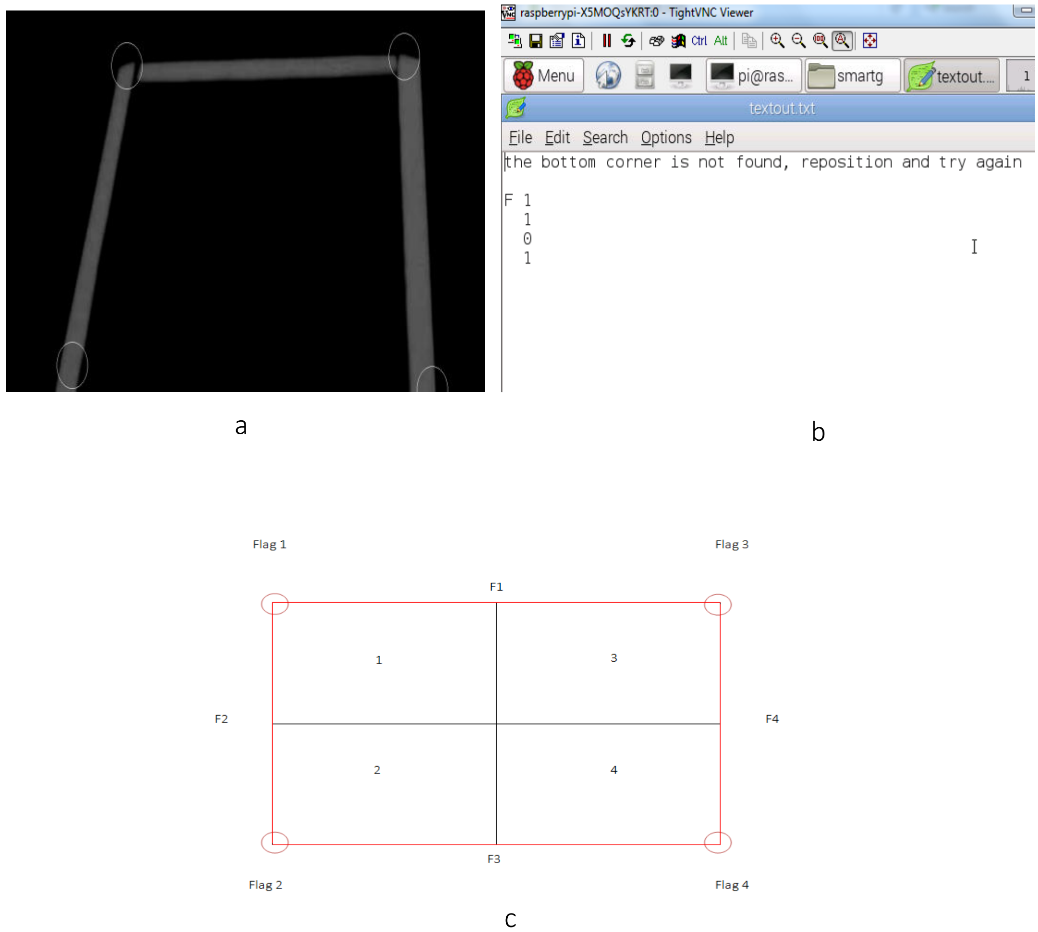


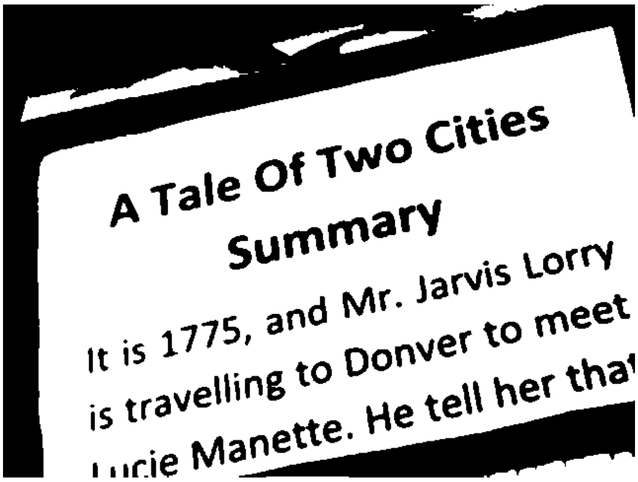
Figure 12: Position detection. (a): the result of border detection without position confirmation, (b) the output after position confirmation function when the bottom line is missing, (c) the reference image for the border functions

5- Angle detection

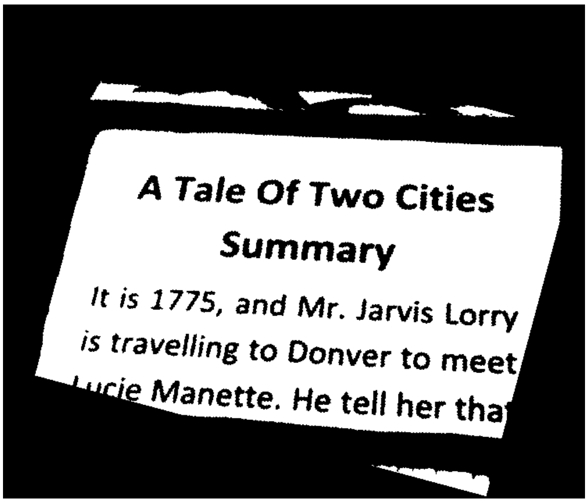
The OCR software used assume no or little rotation in the text image, if the image is significantly rotated, the OCR software will return an empty text file.

The angle detection functions take the theta values found in the hough transform matrix and process it to find the suitable angle, however if the angle of rotation is big, the rotation of the image will not be efficient and therefore the user will be asked to reposition the material.

Figure 13 (a) shows output image before adjustments, figure 13(b) shows the corrected image rotation.



a



b

Figure 13: image rotation example. (a): the output before rotation, (b): results after rotation.

6- Image enhancement

Different noise elimination procedures and filters were applied to estimate the best noise reduction filtering procedure for a better text recognition results, the following is the procedures chosen to improve the images taken by the raspberry pi camera:

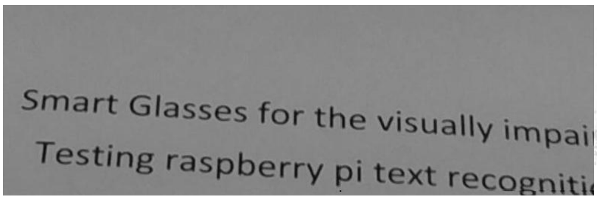
1- Local median filter: this filter is best used to eliminate the salt and paper noise, which was found by observing the images taken using raspberry pi camera.
2- Histogram matching: the histogram of the final image is known, as the image is only text image the histogram can be easily matched with a pure text image reference histogram to get better results by

making the text pixels more clear and eliminating any unwanted gray level.

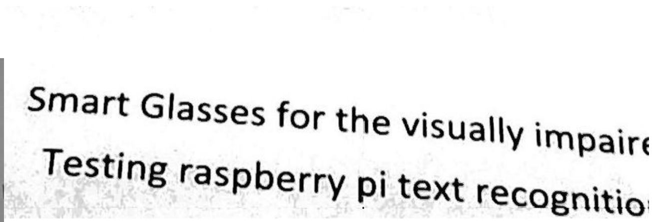
3- Binarization is done using thresholding method to localize the text from its background using two binary values.

4- The filtering process usually leave the text thinner, this could disturb the text recognition in the following step, therefore a dilation morphological operation is done to fill the gaps for better character recognition. Dilation is one of the basic mathematical morphology operators, the input to the dilation operator is a binary image and a structuring element which in this case 2X2 ones matrix, The effect of this procedure is enlarging the boundary regions of the foreground pixels which result in growing their size, by that we eliminate the holes or make them smaller, the foreground pixels in this case are the characters.

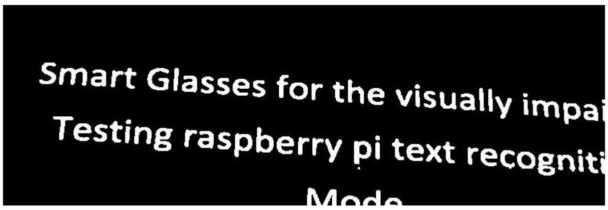
The following images are showing the steps mentioned in this point.



a



b



c

Figure 14: Noise reduction and quality improvement: (a): after the median filter, (b): image after histogram matching and binarization, (c): final image after dilation process.

7- Simulink model: In order to let the raspberry pi run as a standalone, a Simulink model was designed for the previous algorithm and functions. Then the model execution file was deployed to the raspberry pi to run it without the use of the host computer. The model is shown in figure 15, it is consisting of the main subsystems running in the main model which are the hough transform, border detection functions, and image enhancement, is the model implemented so far until the border lines detection. the final image is displayed using the SDL display block and is saved to a file in the raspberry pi for the OCR input.

To run the model as a standalone, a code written in C++ is created to call the executable simulink model, and to call the tesseract ocr, the code can be added to the startup commands to run immediately after powering the raspberry pi.

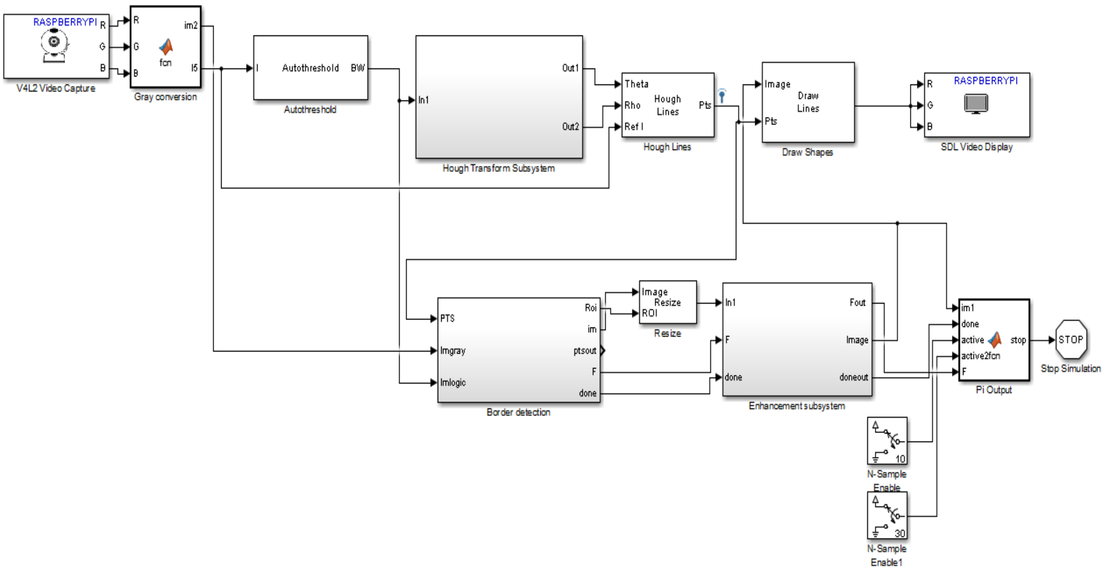
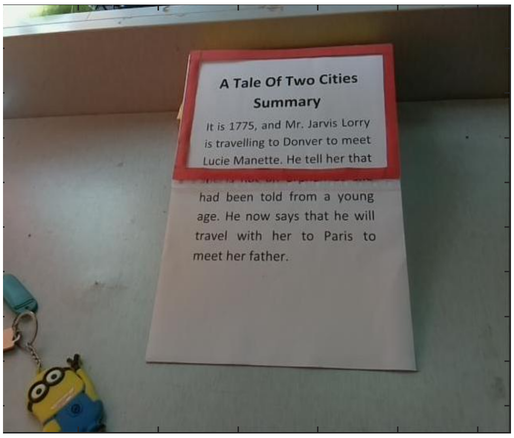


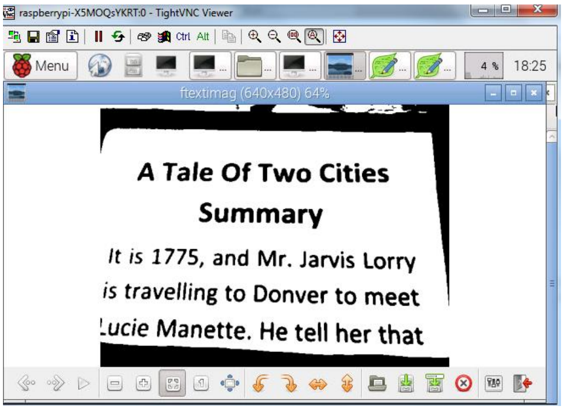
Figure 15: Simulink text recognition model

8- Overall system performance

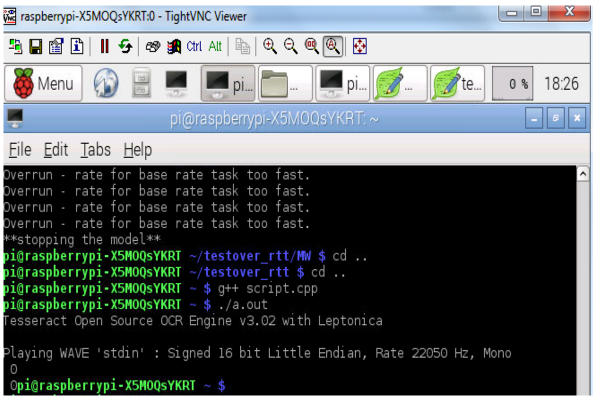
The following figures shows the running model on the raspberry pi results, figure 16 (a) is the original view in front of the camera, figure 16 (b) is the final image saved, the later image is sent to the OCR as shown in the command lines execution in figure 16 (c), and finally figure 16 (d) shows the final text output that is sent to the earpiece. Figure 17 shows the final prototype.



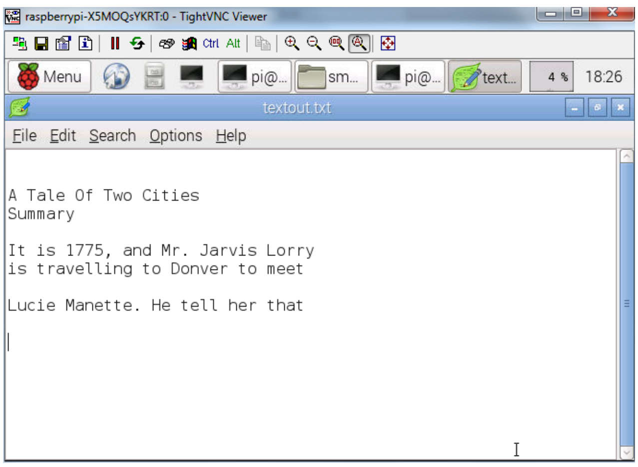
a



b



c



d

Figure 16: The running model results. (a): the original image, (b): the final image to the OCR, (c): C++ program call for the OCR in the terminal, and (d): the final text sent to the audio port.

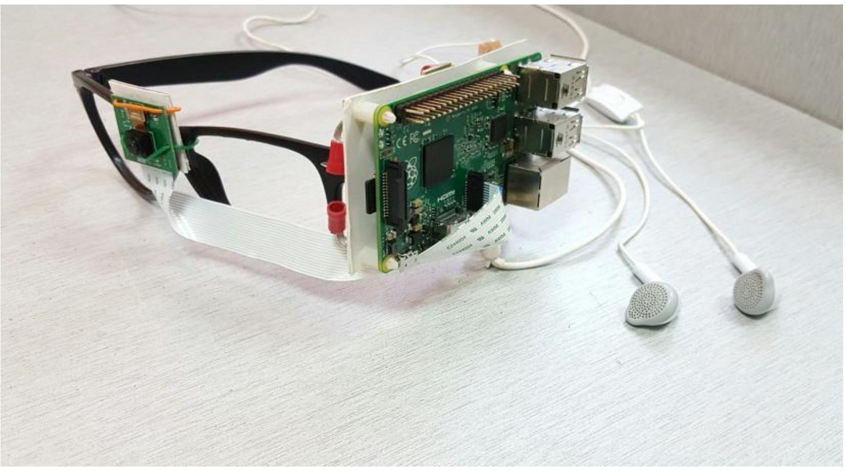


Figure 17: The prototype

Accuracy test

An experiment has been conducted to verify the accuracy of the text recognition mode, five different font styles have been chosen for a 100 word text and three different font sizes have been applied to them. The system was tested to read those 15 materials from a distance of 25cm, and the number of correct letters has been counted to compute the accuracy of the system at each case, figure 18 shows the result of the experiment. With font size 22 the system accuracy is considerably high especially for font style Arial that was around 88% accurate, the lowest accuracy results was using font style Bodoni. For font size 20 the accuracy decreased as expected for all font styles, however, Arial and Times New Roman still give good results which are 85% and 80% respectively. The last font size tested was 18, in this level the system accuracy decreased sharply; font style Arial in this test was 35% accurate while Bodoni was 39% accurate, the lowest was Garamond MT with accuracy 14% only.

From this experiment, it can be concluded that different font styles can result in different accuracy levels; therefore, the font style selection for the materials is important to ensure better results. Arial and Times New Roman are known by their sharp edges and clear letters which resulted in better accuracy percentage when tested against other font styles.

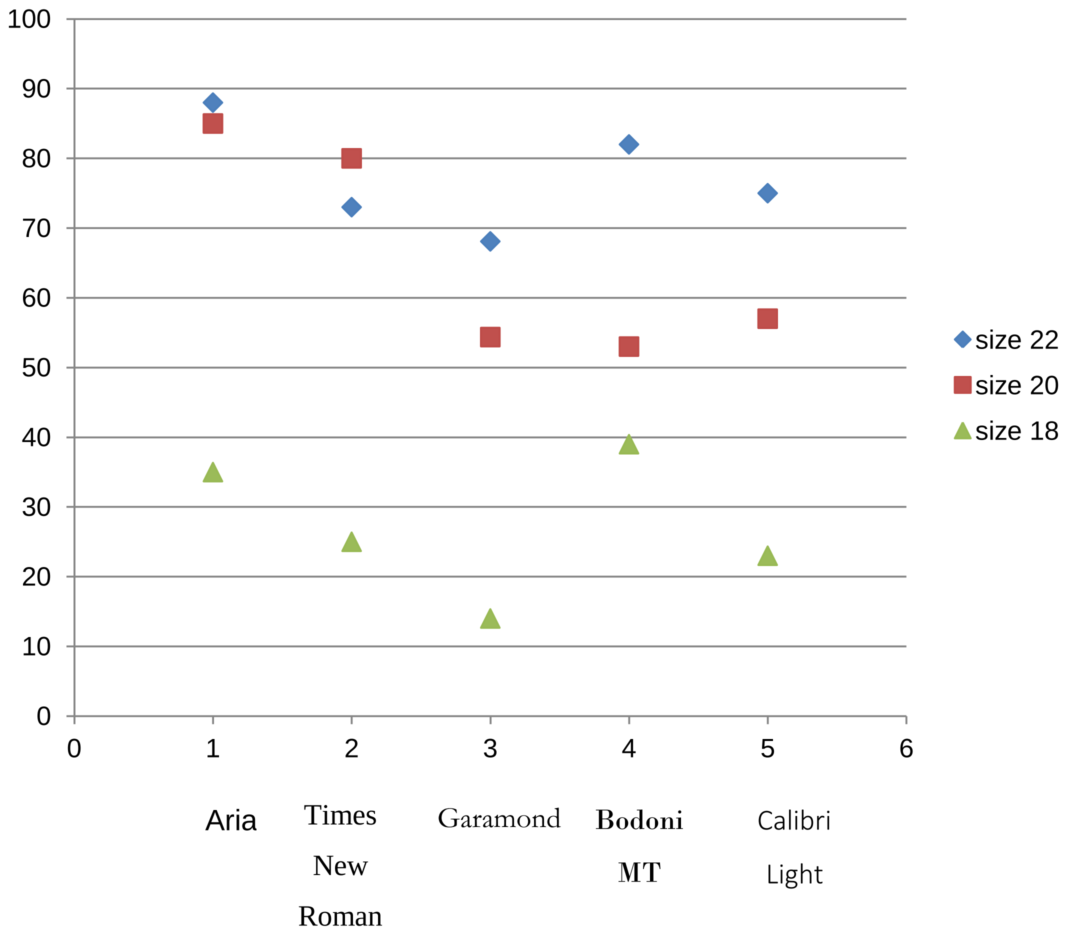


Figure 18: the experiment results, x axis are the font styles, y axis is the accuracy percentages

Apart from improving the algorithm, some reasons have been identified for the decreased accuracy results in smaller font sizes:

1- The camera quality: is the main reason. The OCR pre-processing for the images taken by a camera is usually very complex to compensate the noise and distortion introduced by the camera. However, not all the images taken can be retrieved by the normal techniques. Figure 19 shows an example of text size 14 material taken by the raspberry pi camera; the font in this text is highly distorted.
2- The video stability: In order to enable the camera for image acquisition in simulink, a video capturing block is used, this introduce a blurring effect that becomes significant when the font size is small

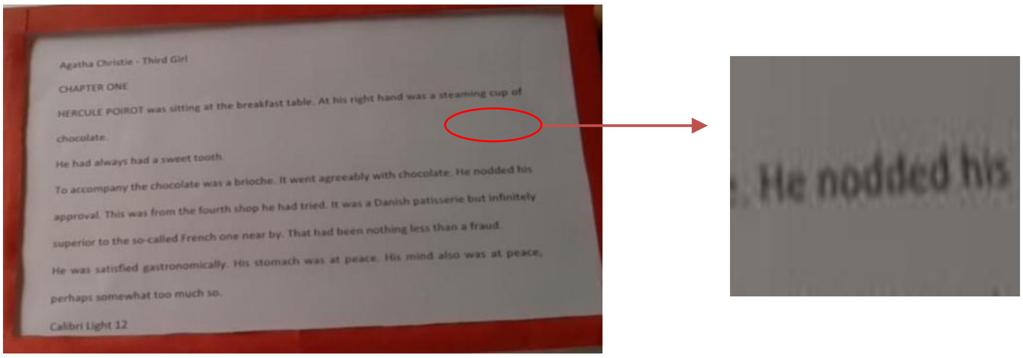


Figure 19: Example of small text (font size 14)

Summary

This chapter discussed about the results obtained from the reading mode implementation. First the results of the methods and techniques are illustrated with examples from the working model, followed by the overall system result and the simulink model explanation for the text recognition mode. Finally the results of the experiment conducted to compute the accuracy of the system is discussed, the results showed that the accuracy has a dependency on the font style (Arial giving the best results), and most importantly the font size, the bigger the font size the better the results for all font styles.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

Visually impaired people are those who are either totally blind or having a very low vision that is legally considered as blindness. The number of the visually impaired people has increased in the recent decades and the difficulties they face in their everyday life are becoming more and more serious with the new technologies, buildings, population and so on. This project is intended to help this type of people to widen their scope of independence by giving them a description of the live scenes delivered in an audio format using an earpiece. The project is implemented using Matlab as a main software program, and the single board computer the raspberry pi 2 as a platform. The project is also using the raspberry pi camera to capture real time videos and an earpiece to voice out the descriptions. The implementation of one mode in the education scope is presented in this report, which is reading mode utilizing the text recognition techniques.

The implementation of the text recognition techniques specified in the methodology was first done using Matlab code and functions, this helps in formalizing the algorithm before modelling it using Simulink. Simulink model and its results are shown in the results and discussion chapter, the results obtained are satisfying for a perfect defined situation, and for a clear text. The system is able to interact with the user through its inputs and the audio output, which prove the feasibility of the concept, also the image processing techniques show good starting results which can be improved by implementing more optimized procedures. The system is tested for different font style and sizes to show the relationship between the system accuracy and the font characteristics of the material, the results obtained from this experiment shows a high dependency in both tested characteristics, font style Arial gives the best results compared to the other 4 styles, also the bigger the font size the better the results obtained from the system.

The project is implemented with emphasize on cost effectiveness, which is kept as low as possible, this appears in the following cost analysis table, the final prototype cost only $75.

Table 2. Cost analysis

Item	Raspberry pi 2	Camera	Earpiece	Accessories	**Total**
Cost	$35	$20	$15	$10	**$75**

As a recommendation for future work, more optimization functions should be added, and user testing results and feedback can be included for better improvements.

As has been concluded from the experiment results, the camera quality is a very important component, therefore, as a recommendation a better camera can be added to the system to improve the results accuracy.

From the simulink progress and procedures, it is recommended to implement the code in higher level language for better results and faster implementation for this model and other future models.

REFERENCES

[1] "Medical dictionary," Farlex Inc, 5 November 2012. [Online]. Available: http://medical dictionary.thefreedictionary.com/Visual+Impairment. [Accessed October 2015].

[2] M. A. Mandal, "Types of visual impairment," AZoM.com Limited trading, 2000. [Online]. Available: http://www.news-medical.net/health/Types-of-visual-impairment.aspx. [Accessed December 2015].

[3] "WHO | Visual impairment and blindness," World Health Organization, 7 April 1948. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs282/en/. [Accessed October 2015].

[4] The Macular Degeneration Foundation, Low Vision Aids & Technology, Sydney , Australia: The Macular Degeneration Foundation, July 2012.

[5] R.Velázquez, "Wearable Assistive Devices for the Blind," in *Wearable and Autonomous Biomedical Devices and Systems for Smart Environment*, vol. 75, A. Lay-Ekuakille and S. C. Mukhopadhyay, Eds., Aguascalientes, Mexico, Universidad Panamericana, 2010, pp. 331-349.

[6] "low vision assistance," EnableMart, 1957. [Online]. Available: https://www.enablemart.com/vision/low-vision-assistance. [Accessed October 2015].

[7] "esight," [Online]. Available: http://esighteyewear.com/.

[8] "What is a raspberry pi," Raspberry Pi Foundation, [Online]. Available: https://www.raspberrypi.org. [Accessed 5 2 2016].

[9] MathWorks, *Simulink and Raspberry Pi Workshop Manual,* MathWorks, Inc, 2015, pp. 6-8.

[10] N. Moore, "The Information Needs of Visually Impaired People," ACUMEN, Leeds, West Yorkshire, England, March 2000.

[11] G. Kleege, "Blind Imagination: Pictures into Words," *Writing Spotlight,* pp. 1-5.

[12] "What is Inclusive Education?," InclusionBC, 1955. [Online]. Available: http://www.inclusionbc.org/our-priority-areas/inclusive-education/what-inclusive-education. [Accessed September 2015].

[13] Unisco, "Modern Stage of SNE Development:Implementation of Inclusive Education," in *Icts In Education For People With Special Needs*, Moscow, Kedrova: Institute For

Information Technologies In Education UNESCO, 2006, pp. 12-14.

[14] A. Ganz, J. Schafer, E. Puleo, C. Wilson and M. Robertson, "Quantitative and Qualitative Evaluation of PERCEPT Indoor Navigation System for Visually Impaired Users," in *Engineering in Medicine and Biology Society (EMBC)*, San Diego, California USA, Aug. 28 2012-Sept. 1 2012.

[15] D. Jain, "Path-Guided Indoor Navigation for the Visually Impaired," in *16th international ACM SIGACCESS conference on Computers & accessibility*, New York, 2014.

[16] M. Ebling, "Virtual Senses," *New Products,* pp. 4-5, 2009.

[17] R. Jafri and S. A. Ali, "Exploring the Potential of Eyewear-Based Wearable Display Devices for Use by the Visually Impaired," in *International Conference on User Science and Engineering*, Shah Alam, 2-5 Sept. 2014 .

[18] I. Anam, S. Alam and M. Yeasin, "Expression: A Dyadic Conversation Aid using Google Glass for People who are Blind or Visually Impaired," in *International Conference on Mobile Computing, Applications and Services*, Austin, TX , 6-7 Nov. 2014 .

[19] OrCam, OrCam, [Online]. Available: http://www.orcam.com. [Accessed December 2015].

[20] S. G. Dedgaonkar, A. A. Chandavale and A. M. Sapkal, "Survey of Methods for Character Recognition," *International Journal of Engineering and Innovative Technology (IJEIT),* vol. 1, no. 5, pp. 180-182, May 2012.

[21] P. S. Shatake, S.A.Patil and P.M.Jadhav, "Review of Text to Speech Conversion Methods," in *IRF International Conference*, Pune,India, 2014.

[22] J. Hansen, *A Matlab Project in Optical Character Recognition (OCR).*

[23] F. YU. and D. GREGORY, "Optical Pattern Recognition:Architectures and Techniques," 1996.

[24] W. Bieniecki, S. Grabowski and W. Rozenberg, "Image Preprocessing for Improving OCR Accuracy," in *Perspective Technologies and Methods in MEMS Design*, Lviv-Polyana , 23-26 May 2007.

[25] E. Vanitha, P. K. Kasarla and E. Kumaraswamy, "Implementation of Text-To-Speech for Real Time Embedded System Using Raspberry Processor," *International Journal and Magazine of Engineering,Technology, Management and research,* vol. 2, no. 7, pp. 1995-1998, July 2015.

APPENDIX I

FYP2 Gantt Chart

Detail / week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Previous research study	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Integrating the camera with raspberry pi	■	■													
Implementing The Text recognition Algorithm		■	■	■	■	■	■								
Preparation of progress report						■	■								
Pre-Sedex preparation								■	■	■					
GPIO configuration and audio output								■	■	■	■	■			
Implementing text to speech										■	■				
Apply improvements											■	■			
Dissertation submission and preparation (soft copy)											■	■	■		
Submission of technical paper												■	■		
Viva															■
Dissertation submission and preparation (hard bound)														■	■

FYP2 Key Milestones

- Text recognition implementation with raspberry pi
- Text to speech implementation
- Progress Report submission
- Pre-Sedex
- Draft Final report
- Image enhancement applied. (Enhancing quality)
- Dissertation submission (soft copy)
- Viva
- Dissertation submission (hard bound)

APPENDIX II

The Matlab codes

1- Image subtraction

```
function [im2,I5]  = redcomponent(R,G,B)
%#codegen

%------------------------ define the image size ------------------------%
 im2=uint8(zeros(640,480));
 I5=uint8(zeros(640,480));
 R1=uint8(zeros(640,480));
 G1=uint8(zeros(640,480));
 B1=uint8(zeros(640,480));
%----------------------------------------------------------------------%


%----------- the RGB image, the blue, green and red assignment -----------%
 R1=R; G1=G; B1=B;
 im=uint8(zeros(640,480,3));
 im(:,:,1)=R1;
 im(:,:,2)=G1;
 im(:,:,3)=B1;
%----------------------------------------------------------------------%


%------------ image subtraction to find the red component only -----------%
im2=rgb2gray(im);
I1=im(:,:,1)-im2;
I2=im(:,:,2)-im2;
I3=im(:,:,3)-im2;
I4=I1-I2;
I5=I4-I3; % final red only image
%----------------------------------------------------------------------%

end
```

2- Border detection code

```
function [Roi,ptsout,done,im2,F,rotation,tosimout] = last(pts,imag,imlogic)
%#codegen
im=logical(zeros(640,480)); %binary image
im=imlogic;
im2=uint8(zeros(640,480));  %grayscale image
im2=imag;
PTS=int32(zeros(30,4));    %hough transform end and start points output
done=0;
PTS=pts;

inter=int32(zeros(1,1));
yinter=int32(zeros(1,1));

%------------ define the intersection points of interest arrays-----------%
m1ymin=int32(zeros(1,30));m1xmin=int32(zeros(1,30));m2ymax=int32(zeros(1,30));m2xmin=int3
2(zeros(1,30));
m3ymin=int32(zeros(1,30));
m3xmax=int32(zeros(1,30));m4xmax=int32(zeros(1,30));m4ymax=int32(zeros(1,30));
%-----------------------------------------------------------------------%

f1=0;f2=0;f3=0;f4=0; % four lines flags
F=double(zeros(1,4));

for i=1:30

%------------- set the four lines flags----------------------------------%
     if(PTS(i,1)==1 || PTS(i,3)==1)
       if(PTS(i,1)==480 || PTS(i,3)==480)
          if(PTS(i,2)<(640/2) || PTS(i,4)<(640/2))
             f1=1;
          else
             f3=1;
          end
       end
    elseif(PTS(i,2)==1 || PTS(i,4)==1)
       if(PTS(i,2)==640 || PTS(i,4)==640)
          if(PTS(i,1)<(480/2) || PTS(i,3)<(480/2))
             f2=1;
          else
             f4=1;
          end
       end
     end
%-----------------------------------------------------------------------%


%----------------------- find intersections ----------------------------- %
    m1=double((PTS(i,2)-PTS(i,4))/(PTS(i,1)-PTS(i,3)));
    a1=PTS(i,2)-(m1*PTS(i,1));
    for j=i+1:30
       m2=double((PTS(j,2)-PTS(j,4))/(PTS(j,1)-PTS(j,3)));
       a2=PTS(j,2)-(m2*PTS(j,1));
       interd=(a2-a1)/(m1-m2);
       yinterd=m2*inter+a2;
```

```
%----------------------- find intersections ----------------------------- %
   m1=double((PTS(i,2)-PTS(i,4))/(PTS(i,1)-PTS(i,3)));
   a1=PTS(i,2)-(m1*PTS(i,1));

   for j=i+1:30
      m2=double((PTS(j,2)-PTS(j,4))/(PTS(j,1)-PTS(j,3)));
      a2=PTS(j,2)-(m2*PTS(j,1));
      interd=(a2-a1)/(m1-m2);
      yinterd=m2*inter+a2;
      inter=(a2-a1)/(m1-m2);
      yinter=m2*inter+a2;
       if (inter>0 && yinter>0 && m1~=m2)
      if(yinter>0 && inter>0 && yinter<(641/2) && inter<(481/2))
         if (m1ymin(1)==0 || (yinter<m1ymin(1)&& im(yinter,inter)==1)  )
               m1ymin=[yinter,m1ymin(1:29)];
               m1xmin=[m1xmin(1),inter,m1xmin(2:29)];

         end

         if (m1xmin(1)==0 || (inter<m1xmin(1) && im(yinter,inter)==1))
              m1xmin=[inter,m1xmin(1:29)];
              m1ymin=[m1ymin(1),yinter,m1ymin(2:29)];
         end

      elseif(yinter>((641/2)+1)  && inter>0 && yinter<641 && inter<(481/2))

        if (yinter>m2ymax(1) && im(yinter,inter)==1 )
               m2ymax=[yinter,m2ymax(1:29)];
               m2xmin=[m2xmin(1),inter,m2xmin(2:29)];

        end
        if (m2xmin(1)==0 || (inter<m2xmin(1) && im(yinter,inter)==1))
               m2xmin=[inter,m2xmin(1:29)];
               m2ymax=[m2ymax(1),yinter,m2ymax(2:29)];

        end
       elseif (yinter>0 && inter>((481/2)+1) && yinter<(641/2) && inter<481)

         if (m3ymin(1)==0 || (yinter<m3ymin(1) && im(yinter,inter)==1) )
               m3ymin=[yinter,m3ymin(1:29)];
                 m3xmax=[m3xmax(1),inter,m3xmax(2:29)];
         end
         if ( inter> m3xmax(1)&& im(yinter,inter)==1)
               m3xmax=[inter,m3xmax(1:29)];
               m3ymin=[m3ymin(1),yinter,m3ymin(2:29)];
         end

        elseif (yinter>((641/2)+1) && inter>((481/2)+1) && yinter<641 && inter<481)
         if (yinter>m4ymax(1) && im(yinter,inter)==1 )
               m4ymax=[yinter,m4ymax(1:29)];
               m4xmax=[m4xmax(1),inter,m4xmax(2:29)];
         end
```

```
        if (  inter>m4xmax(1)&& im(yinter,inter)==1)
             m4xmax=[inter,m4xmax(1:29)];
              m4ymax=[m4ymax(1),yinter,m4ymax(2:29)];
        end

      end
      end
   end

end

%------------------------------------------------------------------------%

F=[f1 f2 f3 f4];

X=int32(zeros(1,8));
flag=double(zeros(1,4));


%---- define and find the four intersection points and their flags -------%
m1x=int32(1);m1y=int32(1);m2x=int32(1);
m2y=int32(657);m3x=int32(1246);m3y=int32(1);m4x=int32(1246);m4y=int32(657);
flag1=0;flag2=0;flag3=0;flag4=0;

for j=1:30
   if((flag1*flag2*flag3*flag4)==1)
      break;
   else
   for i=1:30
      if((flag1*flag2*flag3*flag4)==1)
      break;
      else
         if(m1ymin(j)*m1xmin(i)>0 && im(m1ymin(j),m1xmin(i))==1 && flag1==0)
            m1y=m1ymin(j); m1x=m1xmin(i);
            flag1=1;
         end
         if(m2ymax(j)*m2xmin(i)>0 && im(m2ymax(j),m2xmin(i))==1 && flag2==0)
            m2y=m2ymax(j); m2x=m2xmin(i);
            flag2=1;
         end
         if(m3ymin(j)*m3xmax(i)>0 && im(m3ymin(j),m3xmax(i))==1 && flag3==0)
            m3y=m3ymin(j); m3x=m3xmax(i);
            flag3=1;
         end
         if(m4ymax(j)*m4xmax(i)>0 && im(m4ymax(j),m4xmax(i))==1 && flag4==0)
            m4y=m4ymax(j); m4x=m4xmax(i);
            flag4=1;
         end
      end
   end
   end
end

%------------------------------------------------------------------------%
```

```
X=[m1x,m1y,m2x,m2y,m3x,m3y,m4x,m4y];
flag=[flag1,flag2,flag3,flag4];


%---------------- find the angle of rotation ---------------------------%
distance=sqrt(double(((m2x-m1x)^2 + (m2y-m1y)^2)));
 midpointx=(double(m2x+m1x))/2;
 ang1=atand(((midpointx-1)/(distance/2)));
 if (ang1>10)
  rotation=((ang1-45)/180);
 else
    rotation=0;
 end
  tosimout=[distance,midpointx,ang1,rotation];
%--------------------------------------------------------------------%



%---------- find the length of vertical and horizantal lines -------------%
y1m=abs(m2y-m1y);
y2m=abs(m4y-m3y);
x1m=abs(m3x-m1x);
x2m=abs(m4x-m2x);

VertY=[y1m,y2m];
HoriX=[x1m,x2m];
Vmax=round(max(VertY));
Hmax=round(max(HoriX));
%--------------------------------------------------------------------%


%------------------ find the minimum y and x ---------------------------%
ymin=round(min([m1y,m3y]));
xmin=round(min([m1x,m2x]));
newy=[ymin (Vmax-ymin)];
newx=[xmin (Hmax-xmin)];
%--------------------------------------------------------------------%

matla
%--------- define and assign the traingle for the border cropping --------%
Roi=int32(zeros(1,4));
Roi=[int32(xmin) int32(ymin) int32(Hmax) int32(Vmax)];
ptsout=[m1x m1y; m2x m2y;m3x m3y;m4x m4y];
%--------------------------------------------------------------------%

done=1; % indication for the end of this function

end
```

APPENDIX III

The Main C++ Code

```
#ifndef GPIO_CLASS_H
#define GPIO_CLASS_H
#include <fstream>
#include <iostream>
#include <sstream>
#include "GPIOClass.h"
#include <string>
using namespace std;
/* GPIO Class

 * Purpose: Each object instantiated from this class will control a GPIO pin
 * The GPIO pin number must be passed to the overloaded class constructor
 */
class GPIOClass
{
public:
    GPIOClass();  // create a GPIO object that controls GPIO4 (default
    GPIOClass(string x); // create a GPIO object that controls GPIOx, where x is passed to this
constructor
    int export_gpio(); // exports GPIO
    int unexport_gpio(); // unexport GPIO
    int setdir_gpio(string dir); // Set GPIO Direction
    int setval_gpio(string val); // Set GPIO Value (putput pins)
    int getval_gpio(string& val); // Get GPIO Value (input/ output pins)
    string get_gpionum(); // return the GPIO number associated with the instance of an object
private:
    string gpionum; // GPIO number associated with the instance of an object
};
#endif

using namespace std;

GPIOClass::GPIOClass()
{
    this->gpionum = "4"; //GPIO4 is default
}
```

```
GPIOClass::GPIOClass(string gnum)
{
   this->gpionum = gnum;  //Instatiate GPIOClass object for GPIO pin number "gnum"
}

int GPIOClass::export_gpio()
{
   string export_str = "/sys/class/gpio/export";
   ofstream exportgpio(export_str.c_str()); // Open "export" file. Convert C++ string to C string.
Required for all Linux pathnames
   if (exportgpio < 0){
      cout << " OPERATION FAILED: Unable to export GPIO"<< this->gpionum <<" ."<< endl;
      return -1;
   }

   exportgpio << this->gpionum ; //write GPIO number to export
   exportgpio.close(); //close export file
   return 0;
}

int GPIOClass::unexport_gpio()
{
   string unexport_str = "/sys/class/gpio/unexport";
   ofstream unexportgpio(unexport_str.c_str()); //Open unexport file
   if (unexportgpio < 0){
      cout << " OPERATION FAILED: Unable to unexport GPIO"<< this->gpionum <<" ."<<
endl;
      return -1;
   }

   unexportgpio << this->gpionum ; //write GPIO number to unexport
   unexportgpio.close(); //close unexport file
   return 0;
}

int GPIOClass::setdir_gpio(string dir)
{
```

```
    string setdir_str ="/sys/class/gpio/gpio" + this->gpionum + "/direction";
    ofstream setdirgpio(setdir_str.c_str()); // open direction file for gpio
        if (setdirgpio < 0){
            cout << " OPERATION FAILED: Unable to set direction of GPIO"<< this->gpionum <<"
."<< endl;
            return -1;
        }

        setdirgpio << dir; //write direction to direction file
        setdirgpio.close(); // close direction file
        return 0;
}

int GPIOClass::setval_gpio(string val)
{

    string setval_str = "/sys/class/gpio/gpio" + this->gpionum + "/value";
    ofstream setvalgpio(setval_str.c_str()); // open value file for gpio
        if (setvalgpio < 0){
            cout << " OPERATION FAILED: Unable to set the value of GPIO"<< this->gpionum <<"
."<< endl;
            return -1;
        }

        setvalgpio << val ;//write value to value file
        setvalgpio.close();// close value file
        return 0;
}

int GPIOClass::getval_gpio(string& val){

    string getval_str = "/sys/class/gpio/gpio" + this->gpionum + "/value";
    ifstream getvalgpio(getval_str.c_str());// open value file for gpio
    if (getvalgpio < 0){
        cout << " OPERATION FAILED: Unable to get value of GPIO"<< this->gpionum <<" ."<<
endl;
        return -1;
            }
```

```
    getvalgpio >> val ;  //read gpio value

    if(val != "0")
        val = "1";
    else
        val = "0";

    getvalgpio.close(); //close the value file
    return 0;
}

string GPIOClass::get_gpionum(){

return this->gpionum;

}

#include <iostream>
#include <unistd.h>
#include <errno.h>
#include <stdio.h>
#include <stdlib.h>
#include <signal.h>
#include "GPIOClass.h"

int main (void)
{
    int flag17=0;
    int i;
    int flagout=0;
    int flag4=0;
    int flagfun=0;
    string inputstate;
    string inputstate2;
    GPIOClass* gpio4 = new GPIOClass("4");
    GPIOClass* gpio17 = new GPIOClass("17");

    gpio4->export_gpio();
    gpio17->export_gpio();
```

```
    cout << " GPIO pins exported" << endl;

    gpio17->setdir_gpio("in");
    gpio4->setdir_gpio("in");

    cout << " Set GPIO pin directions" << endl;
while(1)
{
while(!flagfun)
{
    //while(flag17==0)
        //usleep(500000);
        gpio17->getval_gpio(inputstate);
        gpio4->getval_gpio(inputstate2);
        cout << "Current input pin state is " << inputstate  << inputstate2 <<endl;
        if(inputstate == "0" && flag4==0)
        {
            flag17=1;
        }
    if(inputstate2== "0" && flag4==1 )
    {
        flagfun=1;
        }
    else if(inputstate2== "0" && flag4==0 )
    {
            i=system("espeak -v english-us+f4 --stdout -f /home/pi/smartg/textout4.txt -s 140  | aplay
-f cd");
    }

    if(flag17==1) {
            flag4=1;
                cout << "input pin state is ""Pressed"" ." << flag4 <<endl;
        //i=system("espeak -v english-us --stdout -f /home/pi/smartg/textout3.txt -s 140  | aplay -f cd");
        i=system("espeak -v english-us+f4 --stdout -f /home/pi/smartg/textout3.txt -s 140  | aplay -f
cd");
    flag17=0;    }
}

system("sudo ./testover_rtt/MW/testover");
```

```
system("sudo ./testover_rtt/MW/testover");

i=system("tesseract /home/pi/smartg/ftextimag /home/pi/smartg/textout2");
printf("\n %i",i);

i=system("espeak -v english-us+f4 --stdout -f /home/pi/smartg/textout2.txt -s 140  | aplay -f cd");
//printf("\n %i",i);

i=system("tesseract /home/pi/smartg/ftextimag2 /home/pi/smartg/textout2");
printf("\n %i",i);

i=system("espeak -v english-us+f4 --stdout -f /home/pi/smartg/textout2.txt -s 140  | aplay -f cd");
printf("\n %i",i);

i=system("espeak -v english-us+f4 --stdout -f /home/pi/smartg/textout.txt -s 140  | aplay -f cd");
//printf("\n %i",i);

i=system("espeak -v english-us+f4 --stdout -f /home/pi/smartg/textout5.txt -s 140  | aplay -f cd");
while(flagout==0 && flagfun==1)
{
       gpio17->getval_gpio(inputstate);
        gpio4->getval_gpio(inputstate2);
    if (inputstate== "0")
    {
    flagout=1;
    break;
    }
    if (inputstate2== "0")
    {
    flagfun=0;
    break;
    }
}

if (flagout==1) {
i=system("espeak -v english-us+f4 --stdout -f /home/pi/smartg/textout6.txt -s 140  | aplay -f cd");
```

```
        cout << "unexporting pins" << endl;
              gpio4->unexport_gpio();
              gpio17->unexport_gpio();
              cout << "deallocating GPIO Objects" << endl;
              delete gpio4;
              gpio4 = 0;
              delete gpio17;
              gpio17 =0;
break;}}
  cout << "Exiting....." << endl;
  return 0;
}
```

APPENDIX IV

Simulink Model Subsystems

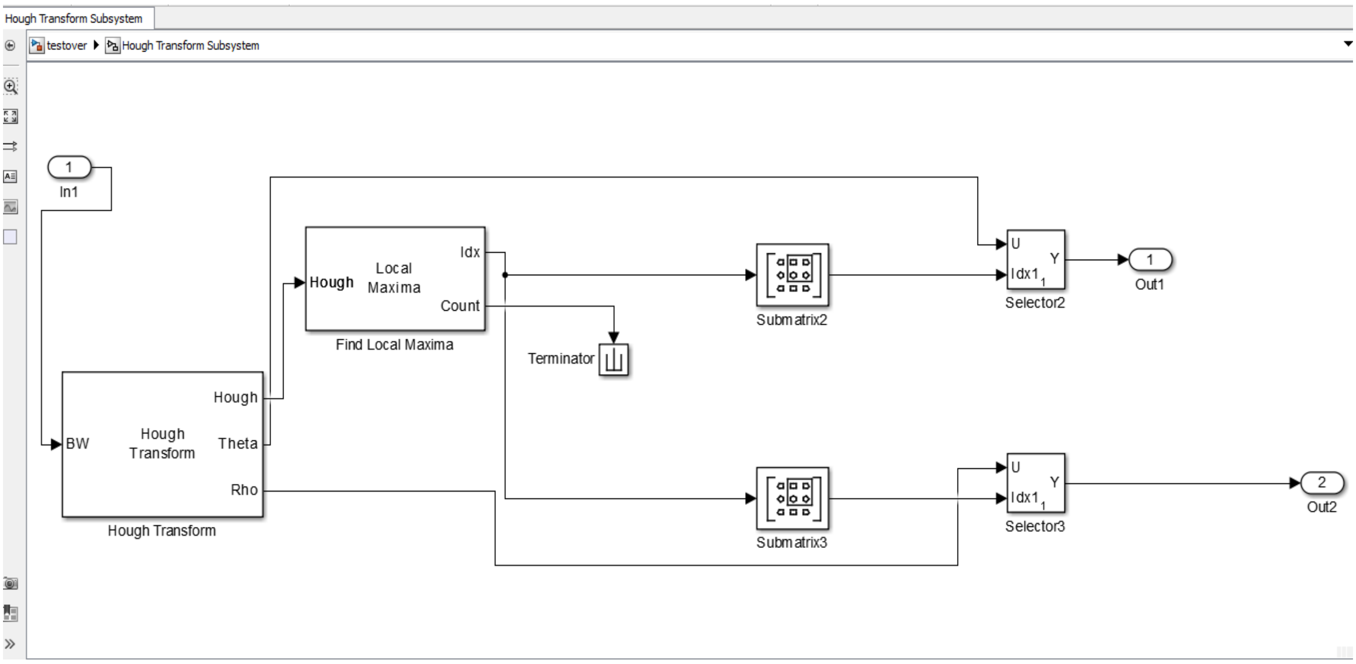


Figure 20: Hough transform subsystem

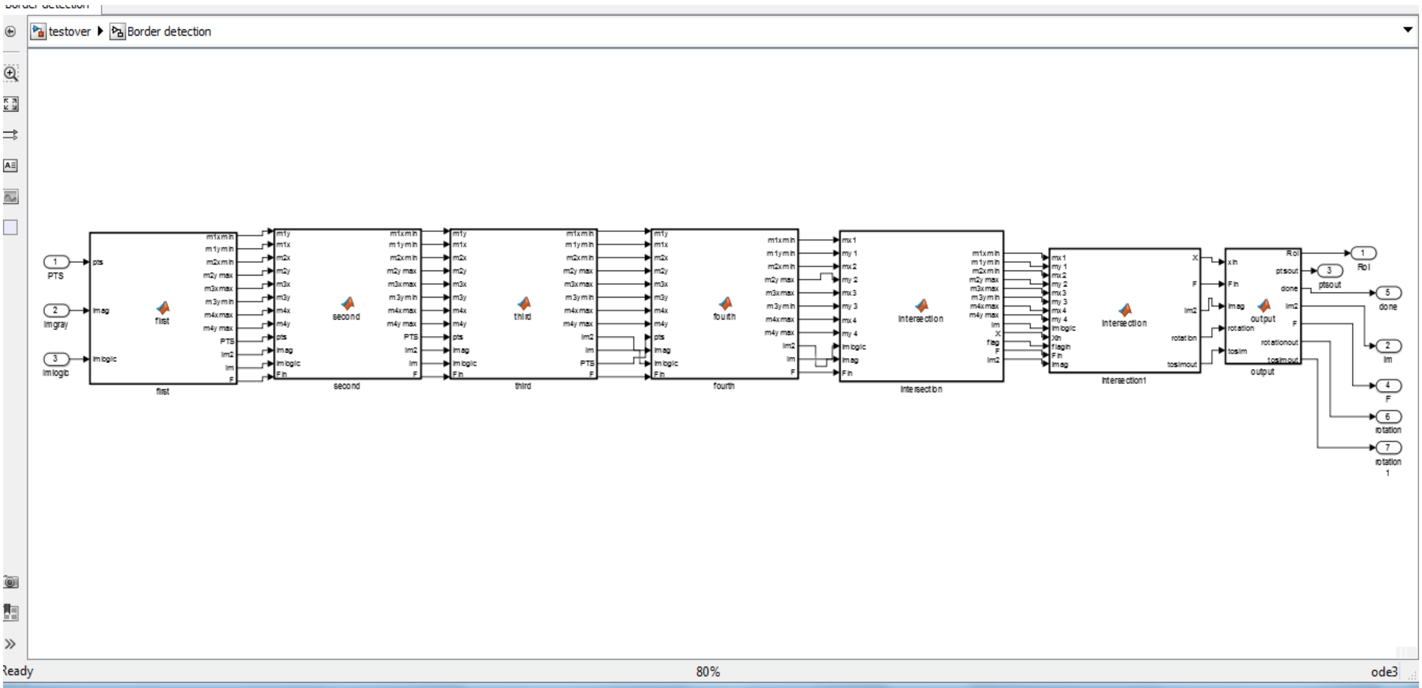


Figure 21: Border detection subsystem

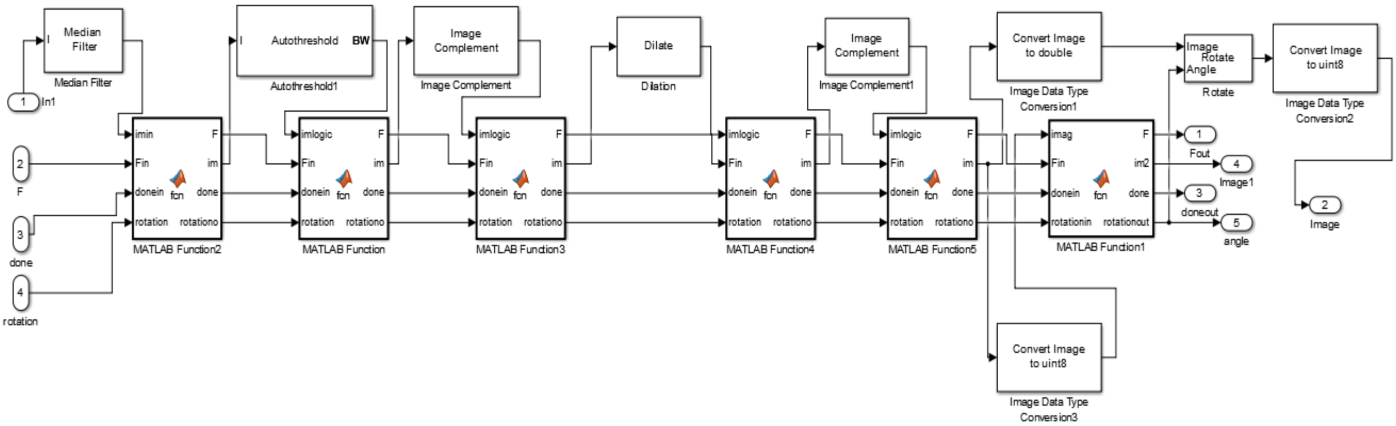
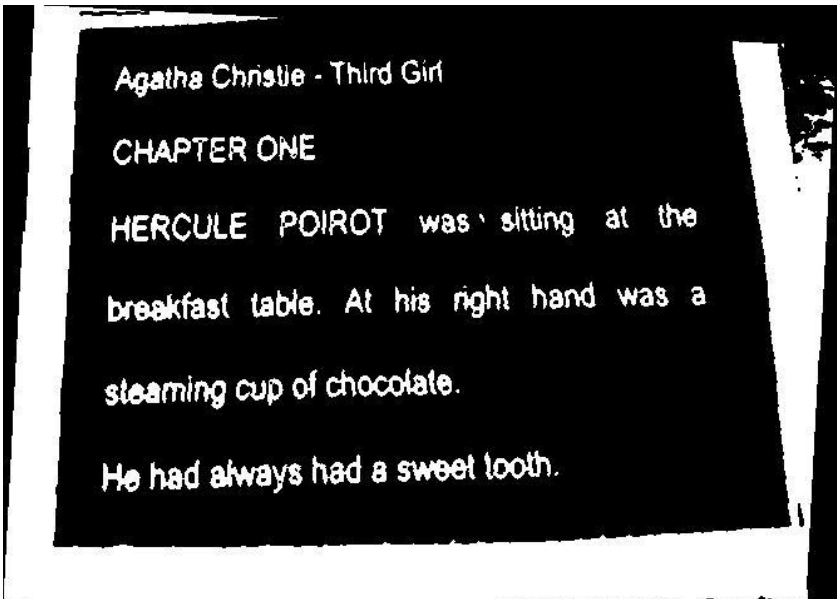


Figure 22: Image enhancement subsystem

APPENDIX V

Accuracy test output samples

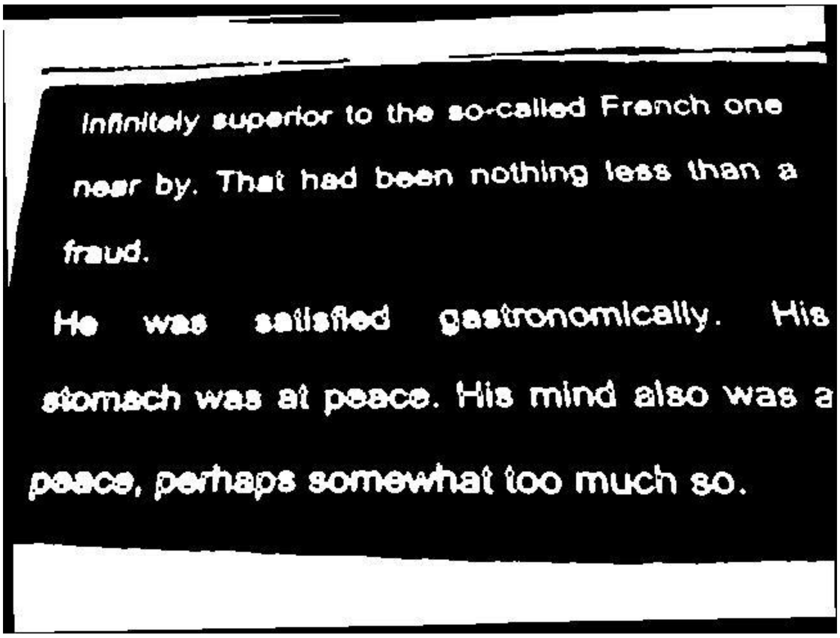

Agatha Christie - Third Girl

CHAPTER ONE

HERCULE POIROT was sitting at the breakfast table. At his right hand was a steaming cup of chocolate.

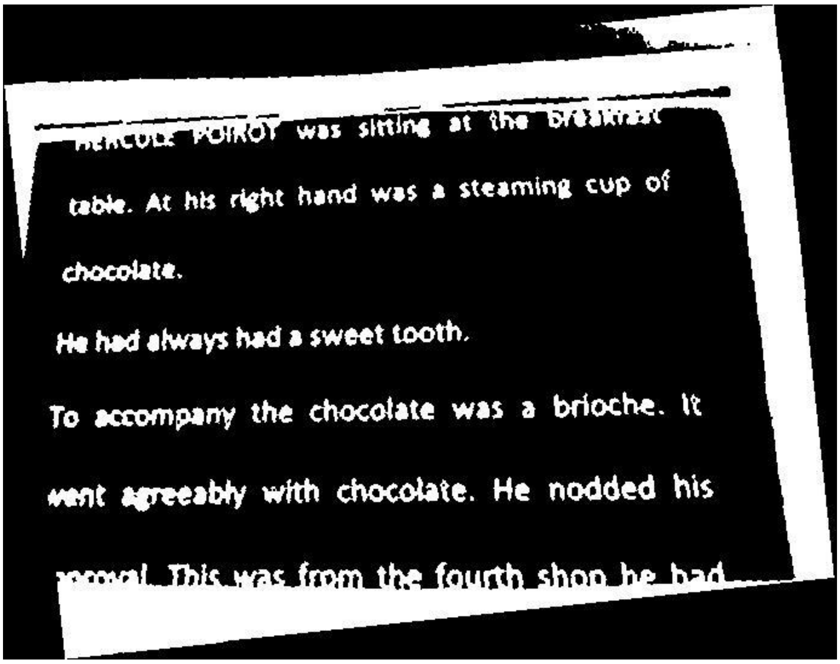
He had always had a sweet tooth.

Figure 23: Arial, size 22


infinitely superior to the so-called French one near by. That had been nothing less than a fraud.

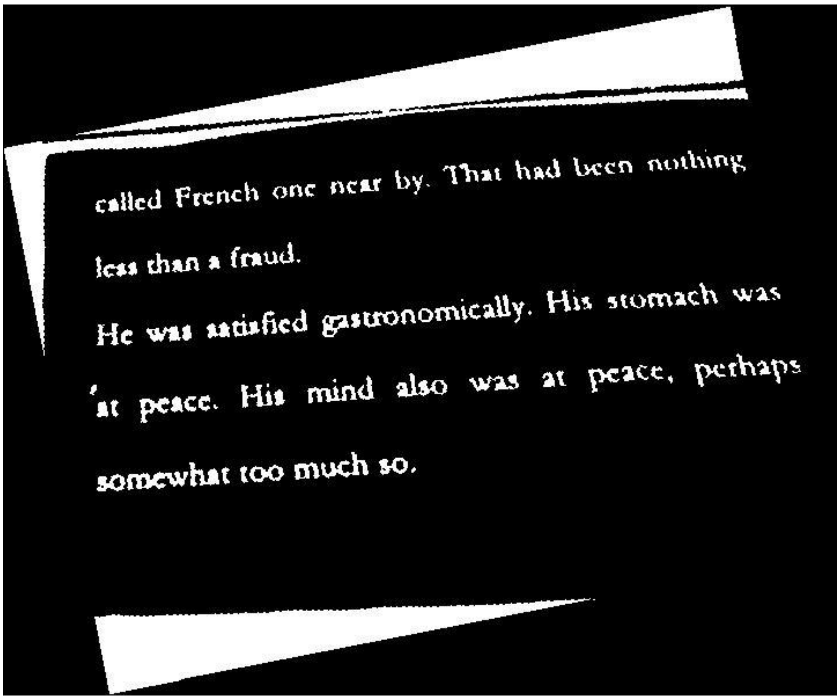
He was satisfied gastronomically. His stomach was at peace. His mind also was a peace, perhaps somewhat too much so.

Figure 24: bodoni MT, size 22



HERCULE POIROT was sitting at the breakfast
table. At his right hand was a steaming cup of
chocolate.
He had always had a sweet tooth.
To accompany the chocolate was a brioche. It
went agreeably with chocolate. He nodded his
approval. This was from the fourth shop he had

Figure 25: Calibri Light, size 22



called French one near by. That had been nothing
less than a fraud.
He was satisfied gastronomically. His stomach was
at peace. His mind also was at peace, perhaps
somewhat too much so.

Figure 26: Garamond, size 22

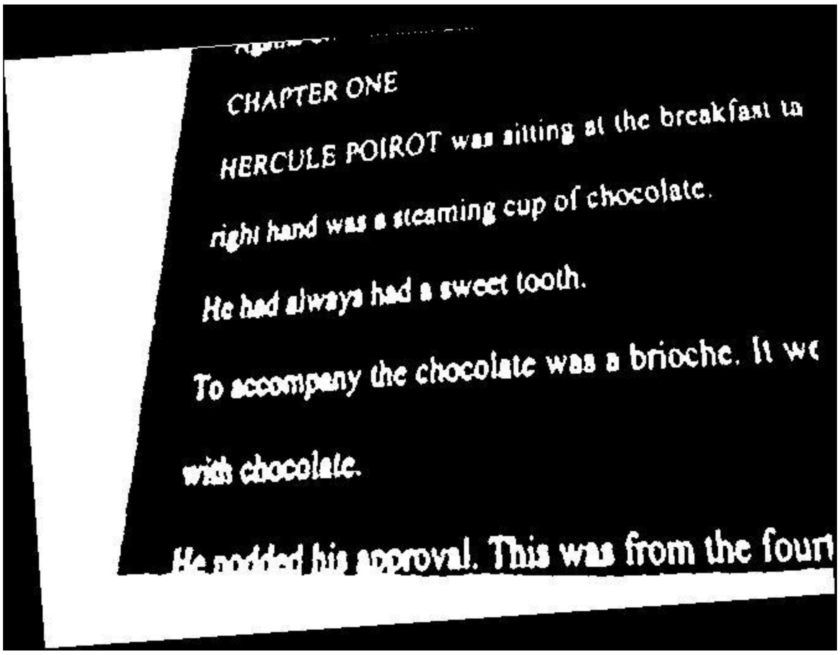

CHAPTER ONE
HERCULE POIROT was sitting at the breakfast ta
right hand was a steaming cup of chocolate.
He had always had a sweet tooth.
To accompany the chocolate was a brioche. It we
with chocolate.
He nodded his approval. This was from the fourt

Figure 27: Times New Roman, size 18

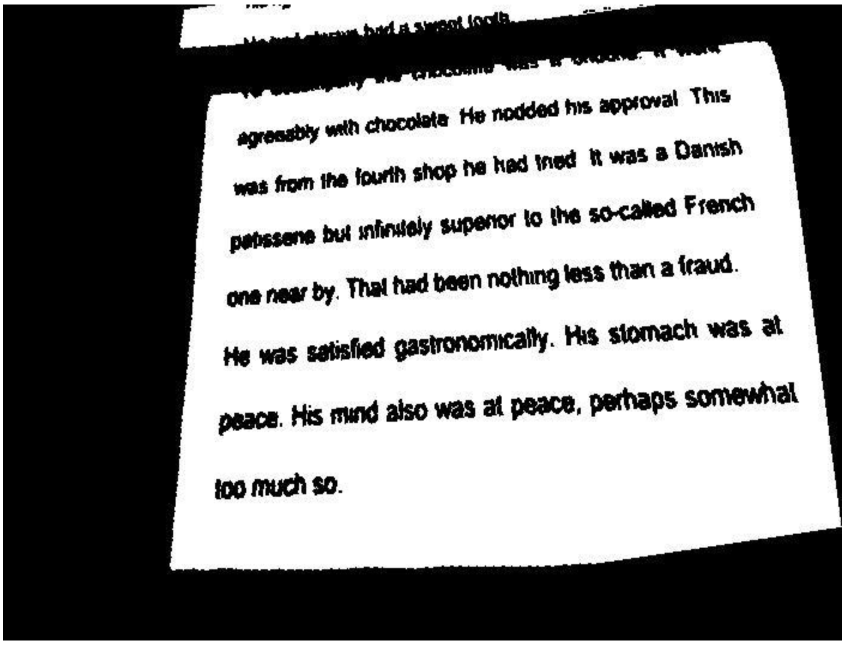

agreeably with chocolate He nodded his approval This
was from the fourth shop he had tried It was a Danish
patisserie but infinitely superior to the so-called French
one near by. That had been nothing less than a fraud.
He was satisfied gastronomically. His stomach was at
peace. His mind also was at peace, perhaps somewhat
too much so.

Figure 28: Arial, size 18

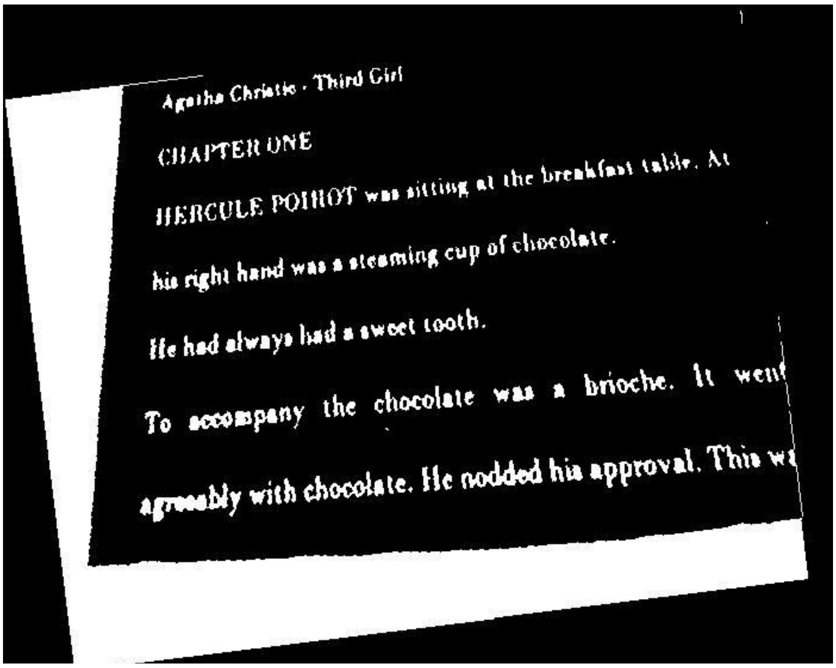


Figure 29: Bodoni MT, size 18

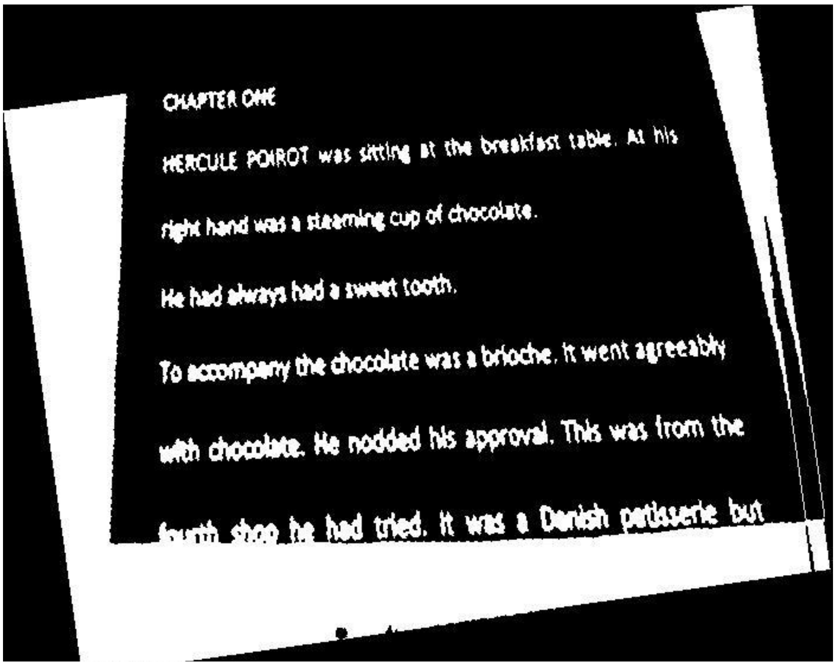


Figure 30: Calibri Light, size 18