International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 – 5161

Available at http://inpressco.com/category/ijcet

Research Article

Reading Assistant for the Visually Impaired

Anusha Bhargava†, Karthik V. Nath† , Pritish Sachdeva†* and Monil Samel†

†Electronics and Telecommunication Engineering, Xavier Institute of Engineering (Mumbai University), Mahim Causeway, Mumbai-400016, India

Accepted 07 April 2015, Available online 10 April 2015, **Vol.5, No.2 (April 2015)**

Abstract

A Majority of the visually impaired use Braille for reading documents and books which are difficult to make and less readily available. This gives rise to the need for the development of devices that could bring relief to the agonizing tasks that the visually impaired has to go through. Due to digitization of books there are many excellent attempts at building a robust document analysis system in industries, academia and research labs, but this is only for those who are able to see. This project aims to study the image recognition technology with speech synthesis and to develop a cost effective, user friendly image to speech conversion system with help of Raspberry Pi. The project has a small inbuilt camera that scans the text printed on a paper, converts it to audio format using a synthesized voice for reading out the scanned text quickly translating books, documents and other materials for daily living, especially away from home or office. Not only does this save time and energy, but also makes life better for the visually impaired as it increases their independency.

Keywords: *Raspberry Pi, OCR, Reading Assistant.*

1. Introduction

A key fact by WHO, 285 million people are estimated to be visually impaired worldwide: 39 million are blind and 246 have low vision. With all the various problems faced by blind people, the problem of reading is our field of interest. We are advancing in technology and science, but in spite of such development the techniques used by the blind are old fashioned. Most of the reading materials available for the blind are in the form of Braille. A person has to learn using Braille just for reading, and if a person is unable to learn Braille then he will be unable to read. Another disadvantage is that an error in understanding will result in reading wrong data. The last disadvantage is that the books, documents etc. have to be converted into a form of raised dots for the blind to read. The books and papers available for the blind in Braille format are quite less in comparison to the vast pool of books which are printed daily. Hence a device to help the blind in reading is a necessity. This document provides details about using raspberry pi as the main unit which has an inbuilt camera that is used to scan any written document and uses Optical character recognition (OCR) to convert the image into a digital text. We then use a text to audio system that will enable us to convert the digital text into a synthesized voice. We are using a raspberry pi which is a credit card-sized single-board computer. It is a complete computer in itself with a working operating system. The operating system can differ as per the use of the device. In our document we have used Raspbian operating system within raspberry pi and have written the code in python language. The camera in the pi is used for scanning any written document (books, bank statements, menus etc.).The scanned image undergoes image processing for obtaining the region of interest and segmentation of the letters from the word. The segmented letters undergo OCR. The output is combined to obtain the individual words as it was present originally in the document. The words obtained are given to a text to speech convertor that allows us to obtain the voice converted output according to the written document.

2. Existing Technology

The most basic and widely used method is Braille. Apart from that the other technology used is Talking Computer Terminal, Computer Driven Braille Printer, Paperless Braille Machines, Optacon etc. These technologies use different techniques and methods allowing the person to read or convert document to Braille. Nowadays Computers are designed to interact by reading the books or documents. Synthesized voice is used to read the content by the computers. We also have devices that scan the documents and use interfaced screen to allow the blind to sense the scanned documents on the screen either in Braille or

*Corresponding author: **Pritish Sachdeva**

by using the shape of the letters itself with help of vibrating pegs. Various phone applications are also developed to help in reading or helping the blind in other ways**.**

3. Literature Survey

3.1 Raspberry Pi

The engineers with a pragmatic approach are the biggest boon to a society. The application of ideas, theories, and new innovations is what drives them. For years the work was done on Arduino boards but with the launch of the very cheap Raspberry Pi it all changed. Raspberry Pi's inception began in 2006 it was finally released on 19 February 2012 as two models: Model A and Model B. After the sale of 3 million units in May 2014, the latest Model B+ was announced in July 2014. It contains many minor improvements based on the user suggestions without any increase in price. Raspberry Pi board costs only $35 and does the work of a computer costing hundreds of dollars. Though its purpose is not to replace computers, laptops etc. but to work in supplement with them. Boot it up, and you have a got a fully functional powerhouse. Grab a four-gigabyte SD card and flash it with the free Linux-based operating system on the Raspberry Pi Foundation's website. Put the SD card into the slot, apply power, and you've got a 700 megahertz workstation with hardware accelerated 3-D graphics—something that would have been state-of-the-art in 2001 and set you back several thousand dollars. The Raspberry Pi offers another path: encouraging experimentation by lowering the cost of accidentally breaking when you're trying to learn.

Technical Specifications

The following are specifications for Model B

- Broadcom BCM2835 SoC processor with 700 MHz ARM1176JZF-S cores
- 512MB RAM
- Videocore 4 GPU supports up to 1920x1200 Resolution
- SD card slot
- 10/100Mbps Ethernet port
- 2 x USB 2.0 ports
- HDMI, audio/video jack
- GPIO header containing 40 pins
- Micro USB power port providing 2A current supply
- DSI and CSI ports
- Dimensions: 85.6x56mm

3.2 Camera Module

The Raspberry Pi camera module size is 25mm square, 5MP sensor much smaller than the Raspberry Pi computer, to which it connects by a flat flex cable (FFC, 1mm pitch, 15 conductor, type B.

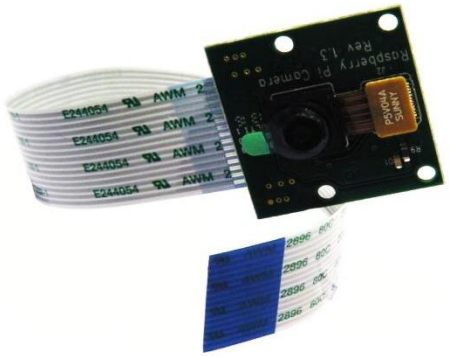

Fig. 1 Raspberry Pi Camera Module

The Raspberry Pi camera module offers a unique new capability for optical instrumentation with critical capabilities as follows:

- 1080p video recording to SD flash memory cards.
- Simultaneous output of 1080p live video via HDMI, while recording.
- Sensor type: OmniVision OV5647 Colour CMOS QSXGA (5-megapixel)
- Sensor size: 3.67 x 2.74 mm
- Pixel Count: 2592 x 1944
- Pixel Size: 1.4 x 1.4 um
- Lens: f=3.6 mm, f/2.9
- Angle of View: 54 x 41 degrees
- Field of View: 2.0 x 1.33 m at 2 m
- Full-frame SLR lens equivalent: 35 mm
- Fixed Focus: 1 m to infinity
- Removable lens.
- Adapters for M12, C-mount, Canon EF, and Nikon F mount lens interchange.
- In-camera image mirroring.
- Higher-resolution still-image capture (2592 x 1944 native resolution, 5 megapixels)
- Low cost ($25 for the add-on camera module).

Open-source, modifiable software for certain aspects of camera control, image capture, and image processing Underlying Linux-based microcontroller with capabilities for HTTP service of images, Ethernet and WiFi connectivity, etc.

3.3 Image Processing

Books and papers have letters. Our aim is to extract these letters and convert them into digital form and then recite it accordingly. Image processing is used to obtain the letters. Image processing is basically a set of functions that is used upon an image format to deduce some information from it. The input is an image while the output can be an image or set of parameters obtained from the image. Once the image is being loaded, we can convert it into gray scale image. The image which we get is now in the form of pixels within a specific range. This range is used to determine the letters. In gray scale, the image has either white or

black content; the white will mostly be the spacing between words or blank space.

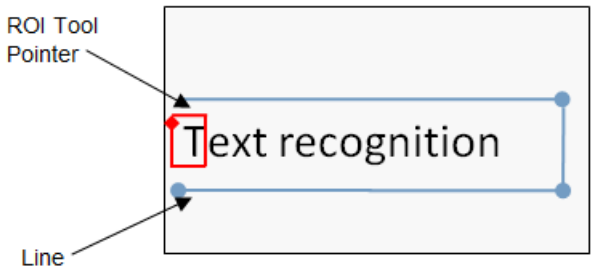

Fig. 2 ROI Tool Pointer Method

The black content will mostly be words. By using proper functions and techniques we can obtain the words. Once words are obtained, they have to be segmented into alphabets. Hence image processing will have 2 parts, (a) getting the region of interest and obtaining words (b) segmentation of words into alphabets.

3.4 Feature Extraction

In this stage we gather the essential features of the image called feature maps. One such method is to detect the edges in the image, as they will contain the required text. For this we can use various axes detecting techniques like: Sobel, Kirsch, Canny, Prewitt etc. The most accurate in finding the four directional axes: horizontal, vertical, right diagonal and left diagonal is the Kirsch detector. This technique uses the eight point neighborhood of each pixel. This way the feature maps along the respective directions are obtained. Then is compression of the Kirsch patterns. This is done to ensure that the neural network will learn without a large number of training samples.

3.5 Optical Character Recognition

Optical character recognition, usually abbreviated to OCR, is the mechanical or electronic conversion of scanned images of handwritten, typewritten or printed text into machine encoded text. It is widely used as a form of data entry from some sort of original paper data source, whether documents, sales receipts, mail, or any number of printed records. It is crucial to the computerization of printed texts so that they can be electronically searched, stored more compactly, displayed on-line and used in machine processes such as machine translation, text-to-speech and text mining. OCR is a field of research in pattern recognition, artificial intelligence and computer vision.

Optical character recognition (OCR) systems allow desired manipulation of the scanned text as the output is coded with ASCII or some other character code from the paper based input text. For a specific language based on some alphabet, OCR techniques are either aimed at printed text or handwritten text.

History of OCR

1929 – Digit recognition machine
1953 – Alphanumeric recognition machine
1965 – US Mail sorting
1965 – British banking system
1976 – Kurzweil reading machine
1985 – Hardware-assisted PC software
1988 – Software-only PC software
1994-2000 – Industry consolidation

3.6 Tesseract

Developed on HP-UX at HP between 1985 and 1994 to run in a desktop scanner. It came neck and neck with Caere and XIS in the 1995 UNLV test. It never was used in an HP product. It was open sourced in 2005 and is highly portable. Tesseract is a free software optical character recognition engine for various operating systems. Tesseract is considered as one of the most accurate free software OCR engines currently available. It is available for Linux, Windows and Mac OS, however, due to limited resources only Windows and Ubuntu are rigorously tested by developers.

Tesseract Architecture

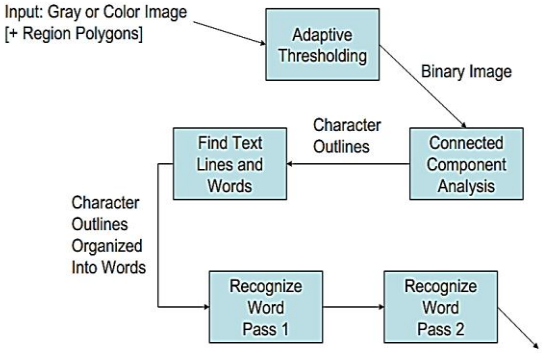

Fig. 3 Tesseract Architecture

An image with the text is given as input to the Tesseract engine that is command based tool. Then it is processed by Tesseract command. Tesseract command takes two arguments: First argument is image file name that contains text and second argument is output text file in which, extracted text is stored. The output file extension is given as .txt by Tesseract, so no need to specify the file extension while specifying the output file name as a second argument in Tesseract command. After processing is completed, the content of the output is present in .txt file. In simple images with or without color (gray scale), Tesseract provides results with 100% accuracy. But in the case of some complex images Tesseract provides better accuracy results if the images are in the gray scale mode as compared to color images. Although Tesseract is command-based tool but as it is open source and it is available in the form of Dynamic Link Library, it can be easily made available in graphics mode.

3.7 Text To Speech

A text to speech (TTS) synthesizer is a computer based system that can read text aloud automatically, regardless of whether the text is introduced by a computer input stream or a scanned input submitted to an Optical character recognition (OCR) engine. A speech synthesizer can be implemented by both hardware and software. Speech is often based on concatenation of natural speech i.e. units that are taken from natural speech put together to form a word or sentence.

Rhythm is an important factor that makes the synthesized speech of a TTS system more natural and the prosodic structure provides important information for the prosody generation model to produce effects in synthesized speech. Many TTS systems are developed based on the principle, corpus-based speech synthesis. It is very popular for its high quality and natural speech output.

Bell Labs Developed VOCODER, a clearly intelligible keyboard-operated electronic speech analyzer and synthesizer. In 1939, Homer Dudely developed VODER which was an improvement over VOCODER. The Pattern Playback was built by Dr. Franklin S. Cooper and his colleagues at Haskins Laboratories. First Electronic based TTS system was designed in 1968. Concatenation Technique was developed by 1970‟s. Many computer operating systems have included speech synthesizers since the early 1980s. From 1990s, there was a progress in Unit Selection and Diphone Synthesis.

Architecture of TTS

The TTS system comprises of these 5 fundamental components:

A. Text Analysis and Detection
B. Text Normalization and Linearization
C. Phonetic Analysis
D. Prosodic Modeling and Intonation
E. Acoustic Processing

The input text is passed through these phases to obtain the speech.

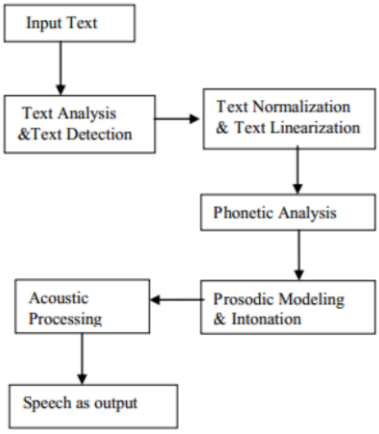

Fig. 4 TTS Architecture

The Text Analysis part is pre-processing part which analyses the input text and organizes it into manageable list of words. It consists of numbers, abbreviations, acronyms and idiomatic and transforms them into full text when needed. Text Normalization is the transformation of text to pronounceable form. The main objective of this process is to identify punctuation marks and pauses between words. Usually the text normalization process is done for converting all letters of lowercase or upper case, to remove punctuations, accent marks , stop words or too common words and other diacritics from letters.

Phonetic Analysis converts the orthographical symbols into phonological ones using a phonetic alphabet, basically known as grapheme-to-phoneme conversion.

The concept of prosody is the combination of stress pattern, rhythm and intonation in a speech. The modelling describes the speaker's emotion. Recent investigations suggest the identification of the vocal features which signal emotional content may help to create a very natural synthesized speech.

The speech will be spoken according to the voice characteristics of a person, there are three type of Acoustic synthesizing available

(i) Concatenative Synthesis
(ii) Formant Synthesis
(iii) Articulatory Synthesis

The concatenation of pre-recorded human voice is called Concatenative synthesis, in this process a database is needed having all the pre-recorded words. The natural sounding speech is the main advantage and the main drawback is the using and developing of large database.

Formant-synthesized speech can be constantly intelligible .It does not have any database of speech samples. So the speech is artificial and robotic.

Speech organs are called Articulators. In this articulatory synthesis techniques for synthesizing speech based on models of the human vocal tract are to be developed. It produces a complete synthetic output, typically based on mathematical models.

4. Block Diagram

- We will be using Raspberry Pi for the project. The first part was booting the Raspberry Pi board by installing the Operating system Raspbian OS and installing the essential libraries and packages.
- Next is the image acquisition system, in which we have interfaced a webcam, to capture the image of the text document.
- This image then goes through Pre-processing, in which we obtain the region of interest (ROI) where-in, we get separate sentences and then words are separated and segmented.
- This data is then given to Template Identification, where the characters are detected, and we obtain the individual alphabets.

- This data is then given to the OCR Algorithm which converts the image data to text data. For OCR we will be writing a program for better outputs.
- The Algorithm scans the image, checks each alphabet or letter and gives a corresponding text output after verifying it with its own database.
- We can use a Dictionary to compare the words detected by the Algorithm for auto-correction. But this is optional.
- Next step is Storage Devices which can save the text data that we get after applying the algorithm in a text file.
- According to the application required the next step's function varies. We have chosen text to speech where the text data is converted to an audio output and is played through the earphones connected to the audio jack.

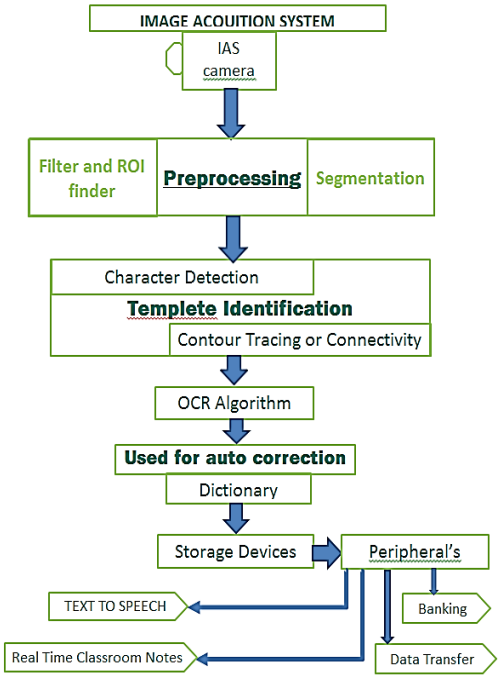

Fig. 5 Block diagram

5. Implementation

5.1 Booting the Pi

The Operating System that is used is Raspbian. It's a derivative of the Linux OS. Specific applications of this software include added multimedia functionalities (essential libraries and packages to our project). It offers a Graphic User Interface which is intuitive to use along with the regular Linux command line interface. To install this:

The image files of Raspbian OS are downloaded from the Raspberry Pi forum.

This image file is extracted and is subsequently burned to the SD card using the software SDFormatter. The SD card is inserted to the Pi and it is given power supply. Default configuration settings are set.

5.2 Updating and Upgrading

Since the Raspberry Pi model in use is B model (the latest being B+), it required updating.

Using apt commands we synchronizes the available software packages and the versions available and downloaded and installed the newer versions.

5.3 Reading Images

We first captured images using RaspiCam and since Standard Image Viewers do not offer image editing services, hence the open source software suite ImageMagik was installed for displaying converting and editing images. It offers support to the Tesseract software that we use for optical Character Recognition (OCR).

5.4 Optical Character Recognition

1. This is a major step of the project and is subdivided into following steps:
2. Installing the open source Tesseract-OCR engine
3. Since Tesseract on its own is a quirky command line tool and a bare back bone structure. It requires layout detection in the image. To achieve this, it is interfaced with ImageMagik.
4. The functionalities of ImageMagik's features are exploited and put to great use. The default output format is changed to churn out pictures which are of the following format: GrayScaleimage, generalised pixel distortion is corrected, text x-height is changed to 20 pixels.
5. Background steps that run when tesseract is called are: Image is opened in ImageMagik and gets converted to a gray scale image first to improve the contrast ratio, is then converted to a black and white image which is read and the identified characters (alphabets and digits in our case) are stored into a word matrix which is later stored as a notepad file containing the text present in the image.
6. The text that is stored in the notepad is in the same order of words and lines as it appears in the image file.

5.5 Text To Speech Converter

To make the Pi read out text, the open source software eSpeak was installed, it is useful for speech synthesizing or as it is commonly referred: text-to-phoneme translation. The eSpeak synthesizer creates voiced speech sounds such as vowels and sonorant constants by adding together sine waves to make the format peaks. It supports Speech Synthesis Markup Language (SSML).

To further improve eSpeak's functionality Festival multi language speech synthesis system was installed. It extends the eSpeak's already existing text to speech system with various APIs (Application programming

Interface) and also boosts the languages supported to include voice packages for Hindi and Marathi. It also improves the words database already existing in eSpeak for the English language.

5.6 Combining Everything

A python script was written which when executed first instructs the Raspberry Pi camera to capture an image, then calls the tesseract OCR to process it, which in turn saves the file with a desired filename, After which the Festival speech package reads the saved text file.

6. Implementation

We have worked on an unprecedented design to create a portable device for assisting the visually impaired. The setup is foldable and hence its portability is enhanced. It can be broken down into two parts and barely takes 5 seconds to be set up again. The two parts of the device are:

1) The stand, onto which the RaspberryPi board is mounted with slot in the wooden board for the camera.
2) A plain slate which has slots for inserting the paper.

The set up looks like this

Conclusion

The 'Reading Assistant for the Visually Impaired' is not just a project that empowers the blind to become independent, but is also a resource saver. It cuts down the cost of printing Braille books along with the time and energy spent into doing so. This is a less costly solution to one of the many challenges that the visually impaired face.

Acknowledgement

We would like to thank our Project guide Ms. Prathibha Sudhakaran, who has been a source of inspiration and her insight and vision has made it possible for us to pursue and understand the developments in these areas. Her patience, encouragement, critique and availability made this dissertation possible. We are also grateful to Xavier Institute of Engineering Management, Principal Dr. Y.D Venkatesh and especially the Head of the Department Dr. Suprava Patnaik and all the faculty and staff who have helped us to be better acquainted with the recent trends in technology and from whom we have learnt so much.

References

V. Ajantha Devi, Dr. S Santhosh Baboo (Jul-Aug 2014), Optical Character Recognition on Tamil Text Image Using Raspberry Pi International Journal of Computer Science Trends and Technology (IJCST) – Vol. 2 Issue 4.

Leija, L.; Santiago, S.; Alvarado, C. (31 Oct-3 Nov 1996), A system of text reading and translation to voice for blind persons Engineering in Medicine and Biology Society, 1996. Bridging Disciplines for Biomedicine Proceedings of the 18th Annual International Conference of the IEEE, Vol. 1, no., pp.405-406 Vol. 1,

Bazzi, I.; Schwartz, R.; Makhoul, J. (Jun 1999), An Omni font open-vocabulary OCR system for English and Arabic, Pattern Analysis and Machine Intelligence, IEEE Transactions on vol.21, no.6, pp.495-504.

J.T. Tou and R.C. Gonzalez (1974), Pattern Recognition Principles, Addison-Wesley Publishing Company, Inc., Reading, Massachusetts

Shinde A. A., D. (2012), Text Pre-processing and Text Segmentation for OCR. International Journal of Computer Science Engineering and Technology, pp. 810- 812.

Smith, R. (2007), An Overview of the Tesseract OCR Engine. In proceedings of Document analysis and Recognition. ICDAR 2007. IEEE Ninth International Conference.

Google. Google Code [Online] 2012. http://code.google.com/p/tesseract-ocr/.