Book Reader Using Raspberry Pi

G. Aparna, (M.Tech, ES) -Research Scholar, Malla Reddy Engineering College (Autonomous)
Mrs. P. Sowjanya, Associate Professor, Malla Reddy Engineering College (Autonomous)

Abstract- Assistive technologies are being developed for visually impaired people in order to live confidently. This project work proposes a camera-based assistive text reading framework to help blind persons read text labels and product packaging from hand-held objects in their daily lives. The project work is framed into three stages. First, Image capturing – Using a mini camera ,the text which the user need to read get captured as an image and have to send to the image processing Platform. Secondly, Text recognition – Using text recognition algorithm, the text will get filtered from the image. Finally, Speech output - A filtered text will be passed into this system to get an audio output. This project work can be able to insist the blind people in their daily life. The entire application will run on Raspberry Pi.

Instead of having an individual chip for the CPU, GPU, USB controller, RAM everything is compressed down into one tidy package. Raspberry Pi needs an Operating system to start up. In the aim of cost reduction, the Raspberry Pi omits any on-board non-volatile memory used to store the boot loaders, Linux Kernels and file systems as seen in more traditional embedded systems. Rather, a SD/MMC card slot is provided for this purpose. After boot load, as per the application program Raspberry Pi will get execute.

Keywords- Raspberry pi, OCR, espeak

I. INTRODUCTION

The National Census of India has estimated around 21.9 Million disabled people in the country. Out of which more than 15 million people in India are blind. This is considered to be the highest among all other disabilities in India. Blind people are an important part of the society. However, their disabilities have made them to have less access to electronic gadgets and computers, Internet, and high quality educational software than the people with clear vision. Many systems have been designed that have some promise for portable use. Portable assistive readers make use of Optical character recognition software. These systems can be used to identify text information within the image. But these systems are intended for and carry out better with document images which contain text snippets with simple background, well organized characters and standard fonts. These systems, thus, fail to distinguish the product labels with complex and attractive backgrounds, different multiple numbers of fonts, different font styles and multiple colours. Portable bar code reader helps a blind user to access information about the different business-related products. The limitation of this reader is, for a blind person it is very hard to exactly locate the barcode and to correctly point the barcode reader at the barcode. Texts printed on the business-related products vary in fonts, colours, font style and the backgrounds. Label s of such products is also designed in multiple decorative patterns. The structure of the proposed framework Some applications (K-Reader Mobile Applications)which are designed specifically for blind users perform good when detecting text information from receipts, fliers, and many other documents. But these systems device fail to give an economic solution of the problem and are available on specific platforms. No smart phones have designed for blind person until now. Thus accessibility of the Mobile application is a different question. Also, these systems cannot read text from business-related products with complex background, non-flat surfaces (text printed on cylindrical surface for e.g. medicinal bottles) and other packaged products. Although a number of assistive reader systems have been designed exclusively for blinds, the proposed system presents a portable and financially viable solution. The proposed structure of work is one of the techniques which work out as a portable assistive product label reader.

The proposed framework is implemented on single board computer–Raspberry pi, in order to provide an economic and potable solution. The framework can be briefly described with the help of –

1. Defining the region of Interest (ROI) -ROI is that part of the captured image which contains the text regions (e.g. product label, sign boards) within the image. It is a challenging problem to automatically localize objects and text
2. Locating the text snippets.
3. Recognition of the located text words.
4. Conversion of the text data into audio format.

II. LITERATURE SURVEY

Presents a strategy of designing image, label or a Book Reader Using Raspberry Pi. The system permits user to here the audio of the book or labels from different sources etc. The camera captures the image and converts to OCR and it gives the output as audio of the source file. The fundamental advantage of the Raspberry Pi is its design. We can here the audio of any file printed on the book or label.

Project targets at the audio based Image detection system, this system consist of USB Camera and Speaker. Which act as an interface between the system and the user Optical Character Recognition or OCR is implemented in this project to recognize characters which are then read out by the system through a speaker.

Presents about the camera is focused in such a position that if a paper is placed before the camera, it captures a full view of the paper into the system. Also, when the camera takes the snapshot of the paper, it is ensured that there should be good clarity of the Image.

Presents all these conditions are met the system takes the Snapshot of the Paper or label, processes it and if it recognizes the content written on the paper it will broadcast on the speaker that the content on the paper has been successfully processed.

Presents all after this it speaks out the content that was converted in to text format in the system from processing the image of the paper. In this way Raspberry Pi Based Reader for Blind helps a blind person to read a paper without the help of any human reader or without the help of physical writing system.

Presents about, complete data of pi is extracted from this site. Information collected from this site is architecture, software downloads, installation steps, documentation regarding pi.

III. SYSTEM ARCHITECTURE

First things to consider before creating the script which takes the picture and stores it automatically are: where the picture is stored, finding the right parameters for the picture so that the image quality and size does not suffer too much.

After a while, some limitations for the pictures are found. The size and quality are reduced to minimize the picture size on the hard drive. The Quality of 75% and the resolution of 1280x720 pixels are sufficient. With these parameters the picture size on the hard drive is around 500KB. That is good starting point, and trade-off between picture quality and available space for picture saving.

All the pictures which are taken by the Python script will be saved to the own folder with current timestamp filename. The folder is located at /var/www/camera/. Apache2 is hosting the folder so that the pictures are available on the website.

Creating the script starts with placing the shebang information and importing the necessary libraries. These libraries are datetime, picamera and time.

```
#!/usr/bin/env/ python
import datetime
import picamera
import time
```

On the second step a function called take Picture should be defined. It does not take any input variables. The function consists of three parts. The first part is the general settings, where the location to the saved pictures and the filename are defined.

```
def takePicture():
location="/var/www/camera/" #Location to the files
date=datetime.datetime.now() #Get current date
file_name=date.strftime("%Y-%m-%d %H%M") #Format the string
```

The second part of the function is defining the settings for the picture size and it starts also the preview mode.

```
#configuration for the pictures
camera = picamera.PiCamera()
camera.resolution = (1280,720)
camera.start_preview()
```

In the last part of the function, the preview mode is kept on for a certain time to warm up the camera. After the warm up time, the function captures the picture and saves it to the predefined location. The picture is named with current timestamp. At the end of the script the preview mode is stopped and the camera is closed.

```
time.sleep(2) #Camera warm up time
# Capture the picture and saved it with the current date
camera.capture("%s%s.jpg" % (location,file_name), quality=75)
camera.stop_preview()
camera.close()
```

Recognition then proceeds as a two-pass process. In the first pass, an attempt is made to recognize each word in turn. Each word that is satisfactory is passed to an adaptive classifier as training data. The adaptive classifier then gets a chance to more accurately recognize text lower down the page. Since the adaptive classifier may have learned something useful too late to make a contribution near the top of the page, a second pass is run over the page, in which words that were not recognized well enough are recognized again. A final phase resolves fuzzy spaces, and checks alternative hypotheses for the x-height to locate small cap text.

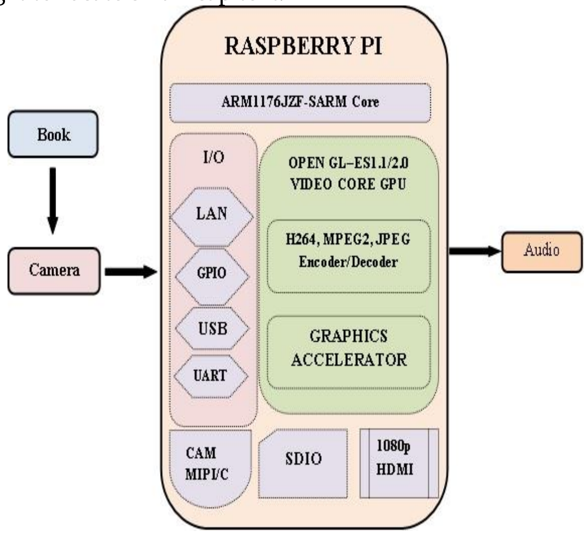

Fig.1: system architecture for product label reading

IV. RASPBERRY PI

A Raspberry Pi is a thirty five dollar, credit card sized computer board which when plugged into an LCD and attachment of a keyboard and a mouse, it is able to complete the functions of any regular PC can. Like a PC, it has RAM, Hard Drive (SD Card), Audio and Video ports, USB port, HDMI port, and Ethernet port. With the Pi, users can create spread sheets, word-processing, browse the internet, play high definition video and much more. It was designed to be a cost friendly computer for users who needed one. There are two models, Model A,B and 3. Model 3 is the faster containing 1GB of RAM as well as the ability to over clock.

Tesseract is an optical character recognition engine for various operating systems. It is free software, released under the Apache License, Version 2.0 and development has been sponsored by Google since 2006. In 2006 Tesseract was considered one of the most accurate open-source OCR engines then available. Matrix Matching converts each character into a pattern within a matrix, and then compares the pattern with an index of known characters. Its recognition is strongest on monotype and uniform single column pages. This method defines each character by the presence or absence of key features, including height, width, density, loops, lines, stems and other character traits. Feature extraction is a perfect approach for OCR of magazines, laser print and high quality images.

V. RESULTS

The result of book reader will consider in many applications like real time reading of a book for blind people even for normal people who prefer to listen audio than reading a book themselves and also converting old printed books into audio books. A low cost, automatic system for reading text books will be implemented that not only converts printed books to digital text, but also reads them as an audio output.

In this project a raspberry pi board connected to a USB camera is used, using which a picture of the page is taken automatically. This image is then processed automatically using a image processing based optical character recognition (OCR) program that can convert the printed text to a digital text, this digital text can then be converted into audio using a voice to text program.

The following Procedure is followed for reading of a book as mentioned below.

After connecting the power cable, USB Camera, HDMI Cable and audio device to the Raspberry Pi

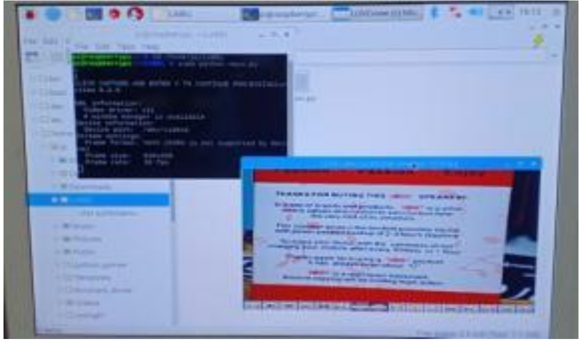

Fig.2: Get Picture in Raspberry Pi Desktop

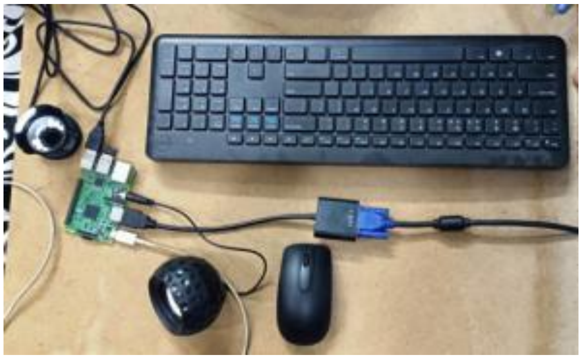

Fig.3: Audio Output from Raspberry Pi

VI. CONCLUSION

In this paper, we have described a prototype system to read printed text on hand-held objects for assisting blind persons. In order to solve the common aiming problem for blind users, we have proposed a motion-based method to detect the object of interest,. Block patterns project the proposed maps of an image patch into a feature vector. Adjacent quality grouping is performed to calculate candidates of text patches prepared for text classification. An Adaboost learning model is employed to localize text in camera-based images.

OCR is used to perform word recognition on the confined to a small area text regions and convert into audio output for blind users. Our future work will extend our localization algorithm to process text strings with characters fewer than three and to design more robust block patterns for text feature extraction. We will also extend our algorithm to contact no horizontal text strings. Furthermore, we will address the important human border issues associated with reading text by blind users.

VII. REFERENCES

[1]. S.V. Rice, F.R. Jenkins, T.A. Nartker, The Fourth AnnualTest of OCR Accuracy, Technical Report 95-03, Information Science Research Institute, University of Nevada, Las Vegas, July 1995.

[2]. R.W. Smith, The Extraction and Recognition of Text from Multimedia Document Images, PhD Thesis, University of Bristol, November 1987.

[3]. R. Smith, “A Simple and Efficient Skew Detection Algorithm via Text Row Accumulation”, Proc. of the 3rd Int. Conf. on Document Analysis and Recognition (Vol. 2), IEEE 1995, pp. 1145-1148.

[4]. World Health Organization. (2009). 10 facts about blindness and visual impairment [Online]. Available: www.who.int/features/factfiles/ blindness/blindness_facts/en/index.html

[5]. Advance Data Reports from the National Health Interview Survey (2008). [Online]. Available: http://www.cdc.gov/nchs/nhis/nhis_ad.htm

[6]. International Workshop on Camera-Based Document Analysis and Recognition (CBDAR 2005, 2007, 2009, 2011). [Online]. Available: http://www.m.cs.osakafu-u.ac.jp/cbdar2011/

[7]. X. Chen, J. Yang, J. Zhang, and A. Waibel, "Automatic detection and recognition of signs from natural scenes," vol. 13, no. 1, pp. 87–99, Jan. 2004.

[8]. D. Dakopoulos and N. G. Bourbakis, "Wearable obstacle avoidance electronic travel aids for blind: A survey," IEEE Trans. Syst., Man, Cybern., vol. 40, no. 1, pp. 25–35, Jan. 2010.

[9]. B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in Proc. Comput. Vision, 2010, pp. 2963–2970.

[10]. Y. Freund and R. Schapire, "Experiments with a new boosting algorithm," in Proc. Int. Conf. Machine Learning, 1996, pp. 148–156.

[11]. N. Giudice and G. Legge, "Blind navigation and the role of technology," in The Engineering Handbook of Smart Technology for Aging, Disability, and Independence, A. A. Helal, M. Mokhtari, and B. Abdulrazak, Eds. Hoboken, NJ, USA: Wiley, 2008.

[12]. A. Shahab, F. Shafait, and A. Dengel, "ICDAR 2011 robust reading competition: ICDAR Robust Reading Competition Challenge 2: Reading text in scene images," in Proc. Int. Conf. Document Anal. Recognit., 2011, pp. 1491–1496.

[13]. K. Kim, K. Jung, and J. Kim, "Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm," IEEE Trans. Pattern Anal. Mach. Intell., vol. 25, no. 12, pp. 1631–1639, Dec. 2003.