IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering

Vol. 8, Issue 03, March 2020

Reader for Visually Impaired using Raspberry Pi

Nagarajan N[1], Sainath@Siddharth G[2], Soorya Prakash S [2], Jayanth Raj[2], Stalin Raj S[2]

Assistant Professor, Department of Instrumentation and Control Engineering,

Sri Manakula Vinayagar Engineering College, Puducherry, India[1]

Student, Department of Instrumentation and Control Engineering,

Sri Manakula Vinayagar Engineering College, Puducherry, India[2]

Abstract: This paper explains the automated document reader for blind people with the help of Raspberry Pi. This process takes place with the help of Optical Character Recognition (OCR) technology for identification of the printed characters using image sensing devices and computerized programming. It converts images of typed, handwritten, or printed text into machine encoded text with the help of OCR. In this research these images are converted into audio output (Speech)with the help of OCR and Text-to-speech synthesis. The conversion of printed document into text files is done with the help of Raspberry Pi which again uses Tesseract library and Python programming. The text files are processed with the help of OpenCV library and python programming language and hence the audio output is obtained.

Keywords: Character recognition, Low power, Document Image Analysis (DIA), Raspberry Pi 3B, Speech Output, OCR based book reader, OpenCV, Python Programming.

I. INTRODUCTION

This project is used for the detection, reading and conversion of documented text to help blind and visually impaired people. The overall theme and algorithm has a success rate of 92 percent on the test set as the unread text is significantly small and distant from the image capturing element. We have proposed a technique to extract text from the written or typed documents, convert them to machine encoded text with Optical Character Recognition(OCR), create the text files and then process them with the help of Digital Image Analysis (DIA) to convert the text into audio output. Our main aim is on enhancing the capabilities of blind people by providing them a method from which the information can be fed to them in the form of a speech signal. This project can also be implemented for the automatic detection of road signs, warning signs, in other terms to improve the blind navigation on larger scale.

II. WORKING PRINCIPLE

When the capture button is clicked, the system captures the image of the document which is placed in front of the capturing element which is connected to the ARM microcontroller through USB. After selecting the process button the captured document image undergoes Optical Character Recognition (OCR) Technology. OCR technology makes the conversion of scanned images of printed text or symbols into text or information that can be understood or edited with the help of a computer program. We are using TESSERACT library function for the OCR technology. With the help of Text-to-speech library the data will be converted to audio file. Camera acts as a detecting element the image of the given document, then the image is processed internally and separates the label from image with the help of open CV library and finally identifies the text which is pronounced in voice format. Now the converted text into audio output is listened with the help of connecting headsets via 3.5mm audio jack or speakers via Bluetooth.

III. PROJECT IMPLEMENTATION

The operating system under which the proposed system is executed with the help of Raspbian which is derived from the Debian operating system. Python programming language is the language used for the Raspbian algorithm which is a script language. Open CV library is the functions used for algorithm. OpenCV is an open source vision library, which is written in C and C++ and runs with the help of Linux, Windows and Mac OS. OpenCV was designed for computational efficiency and real-time applications. OpenCV is written in optimized C and works perfectly in multi-core processors.

The Operating system used is Raspbian (Debian), Language used is Python2.7, Platform used are Tesseract and OpenCV (Linux-library), Library files are OCR engine and TTS engine

DOI 10.17148/IJIREEICE.2020.8311

A. Raspberry PI

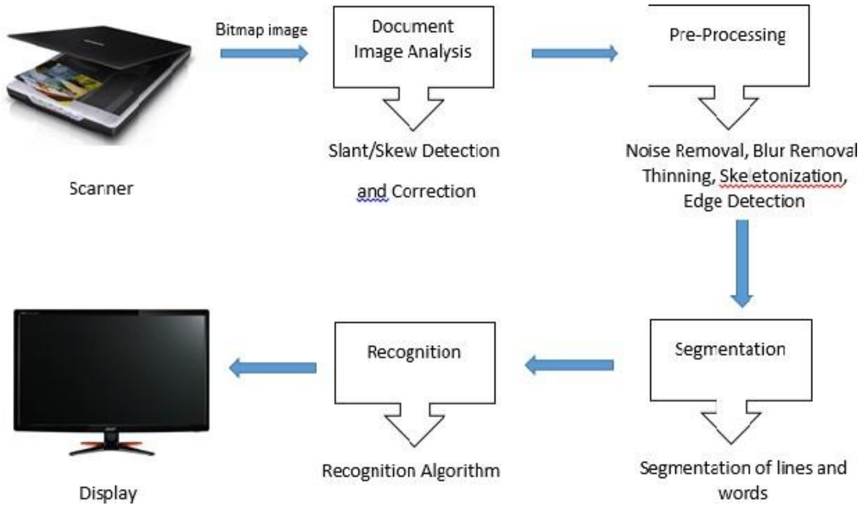
Raspberry Pi is a low cost, small sized computer that plugs into a displaying element or television and uses standard peripheral devices such as keyboard and mouse. There are two models of Raspberry Pi, they are Raspberry Pi 2 and Raspberry Pi 3. These two are similar with few advance features on Raspberry Pi 3. Compared to the Raspberry Pi 2 it has a 1.2GHz 64-bit quad-core ARMv8 CPU, 802.11n Wireless LAN, Bluetooth 4.1, Bluetooth Low Energy (BLE), 4 USB ports, 40 GPIO pins, Full HDMI port, Ethernet port, Combined 3.5mm audio jack and composite video, Camera interface (CSI), Display Interface (DSI), Micro SD card slot, VideoCore IV 3D graphics core

B. OPTICAL CHARACTER RECOGNITION (OCR)

Optical Character Recognition is a text recognition method which converts the written text or printed documents of the text to be rendered into convertible soft copies or text documents. OCR technology is used for the purpose of scanning of text formatted elements from the captured images and converting that image into the editable text documents.

It is a simple and common method of digitizing printed text documents so that they can be easily electronically edited, searched, saved occupying less space, displayed online and used in machine level processes such as cognitive computing, machine learning, translation, text-to-speech etc.

There are two types of OCR. One of them is used for recognizing printed characters and another one is for recognizing hand written text.

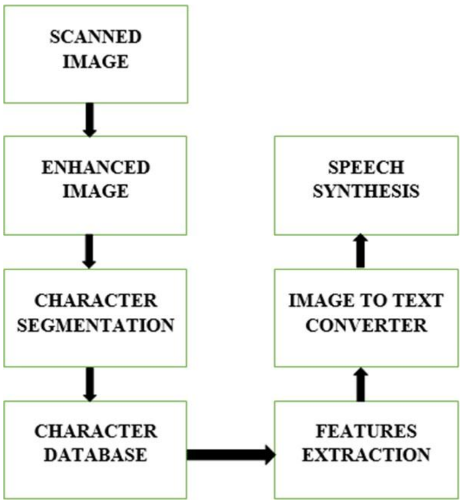

Bitmap image
Document Image Analysis
Pre-Processing
Scanner
Slant/Skew Detection and Correction
Noise Removal, Blur Removal Thinning, Skeletonization, Edge Detection
Segmentation
Recognition
Display
Recognition Algorithm
Segmentation of lines and words

IV. FLOW OF PROCESS

A. IMAGE CAPTURE

The first and foremost step in this process is the one in which the document is placed under the capturing element and that element captures the image of the placed printed or written document. The quality of the image captured by the capturing element must be high so that it has a fast and clear recognition of the printed text due to the high-resolution capturing element. The function for the skew detection checks for a perfect angle of orientation between -15 and +15 degrees and if the orientation is detected, then a simple process of image rotation takes place till the lines of the image matches with the original horizontal axis, which generates skew corrected image. The noise introduced during the work of the capturing element or due to the low quality of the image has to be rectified before further processing.

DOI 10.17148/IJIREEICE.2020.8311

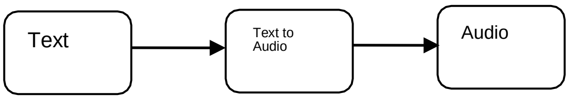


B. Image to Text Conversion

The American Standard Code for Information Interchange (ASCII) values of the captured elements are processed with the help of the Raspberry Pi board. Next, the captured elements are matched with the corresponding and related templates and saved as a normal text transcription. This transcription is further delivered to the audio output where the audio is listened by the visually impaired people.

C. Text to Audio Conversion

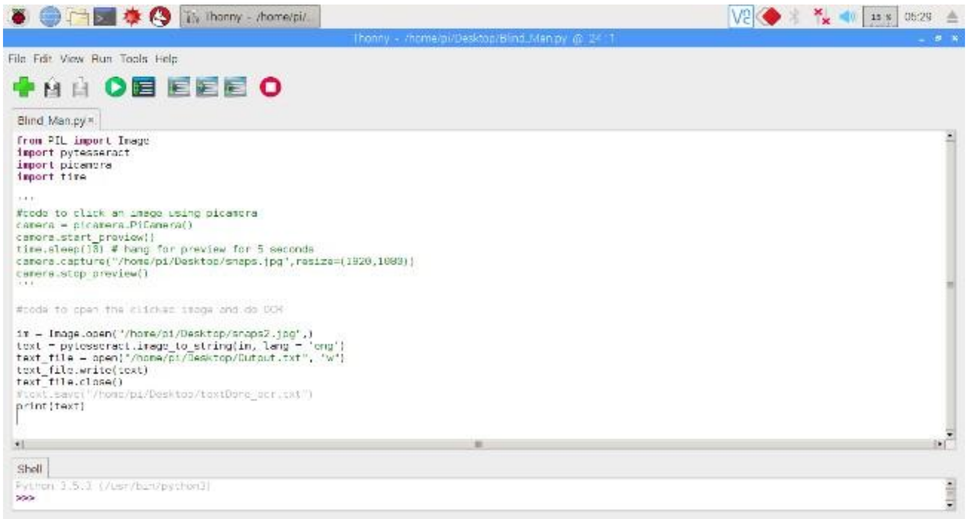
The ultimate aim and scope of this project is obtained with the conclusion of the receding module of Optical Character Recognition. This module performs the operation of conversion of the transformed text to audio output. The Raspberry Pi has an in-built audio jack, the audio is generated and given out from the Raspberry Pi with the help of a Pulse Width Modulation (PWM) output and is filtered at a minimum range. An USB audio card can be used to improve the sound quality of the audio output.



When the character recognition process is completed, the character codes in the text file are processed with the help of Raspberry Pi which recognizes a character with the help of TESSERACT algorithm and python programming.

D. Python Programming

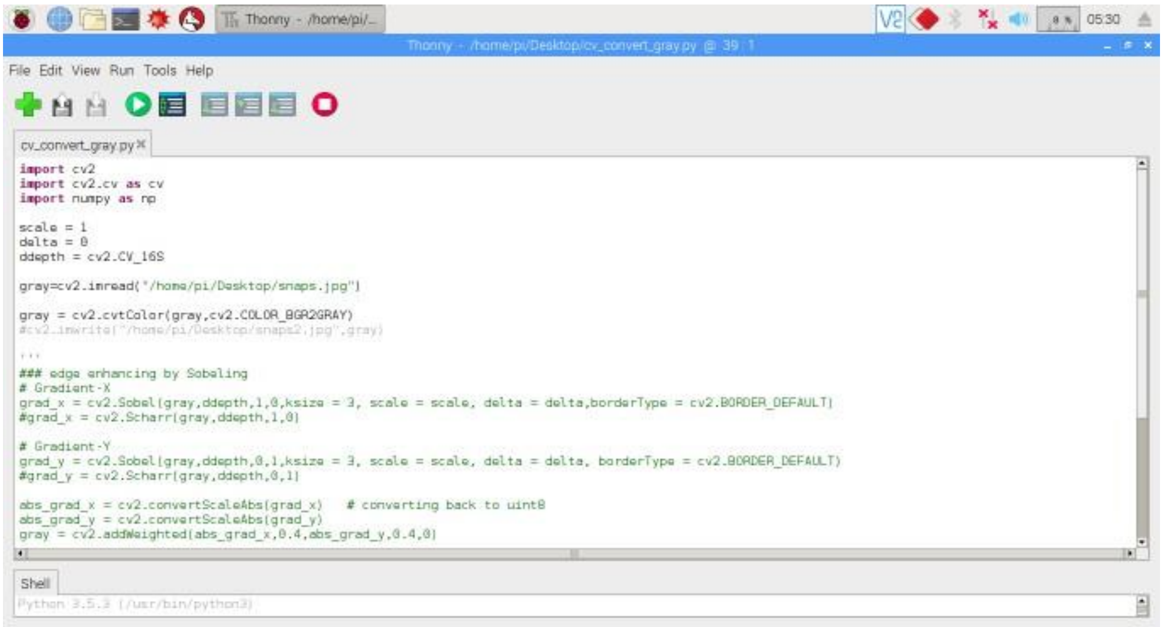
The following screenshots are the windows of the python programming used in this project.


Thonny - /home/pi/...
Thonny - /home/pi/Desktop/Blind_Man.py @ 24 : 1
File Edit View Run Tools Help
Blind_Man.py
Shell

```
from PIL import Image
import pytesseract
import picamera
import time

'''
#code to click an image using picamera
camera = picamera.PiCamera()
camera.start_preview()
time.sleep(18) # hang for preview for 5 seconds
camera.capture("/home/pi/Desktop/snaps.jpg',resize=(1920,1080))
camera.stop_preview()
'''

#code to open the clicked image and do OCR

im = Image.open('/home/pi/Desktop/snaps2.jpg',)
text = pytesseract.image_to_string(im, lang = 'eng')
text_file = open("/home/pi/Desktop/Output.txt", 'w')
text_file.write(text)
text_file.close()
#text.save("/home/pi/Desktop/textDone_ocr.txt")
print(text)
```

```
Python 3.5.3 (/usr/bin/python3)
>>>
```

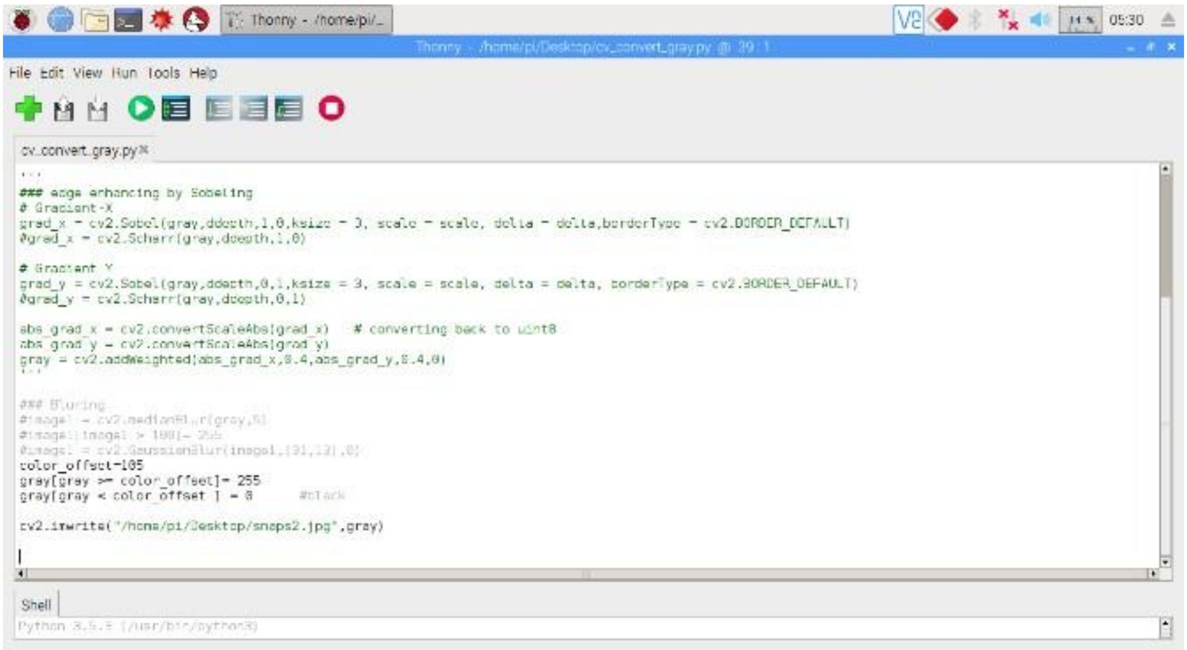
Running camera to click a picture



```
import cv2
import cv2.cv as cv
import numpy as np

scale = 1
delta = 0
ddepth = cv2.CV_16S

gray=cv2.imread("/home/pi/Desktop/snaps.jpg")

gray = cv2.cvtColor(gray,cv2.COLOR_BGR2GRAY)
#cv2.imwrite("/home/pi/Desktop/snaps2.jpg",gray)

'''
### edge enhancing by Sobeling
# Gradient-X
grad_x = cv2.Sobel(gray,ddepth,1,0,ksize = 3, scale = scale, delta = delta,borderType = cv2.BORDER_DEFAULT)
#grad_x = cv2.Scharr(gray,ddepth,1,0)

# Gradient-Y
grad_y = cv2.Sobel(gray,ddepth,0,1,ksize = 3, scale = scale, delta = delta, borderType = cv2.BORDER_DEFAULT)
#grad_y = cv2.Scharr(gray,ddepth,0,1)

abs_grad_x = cv2.convertScaleAbs(grad_x)   # converting back to uint8
abs_grad_y = cv2.convertScaleAbs(grad_y)
gray = cv2.addWeighted(abs_grad_x,0.4,abs_grad_y,0.4,0)
```

```
Python 3.5.3 (/usr/bin/python3)
```

Using open CV library to enhance image

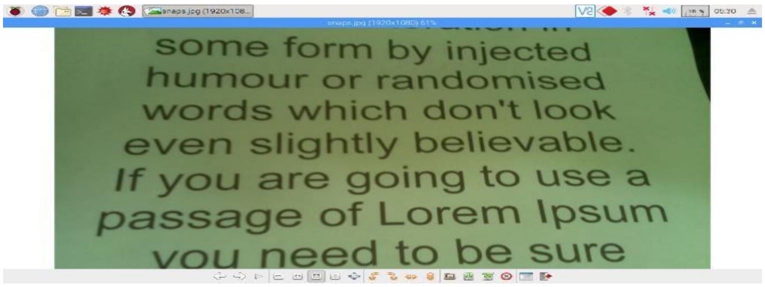


```
'''
### edge enhancing by Sobeling
# Gradient-X
grad_x = cv2.Sobel(gray,ddepth,1,0,ksize = 3, scale = scale, delta = delta,borderType = cv2.BORDER_DEFAULT)
#grad_x = cv2.Scharr(gray,ddepth,1,0)

# Gradient-Y
grad_y = cv2.Sobel(gray,ddepth,0,1,ksize = 3, scale = scale, delta = delta, borderType = cv2.BORDER_DEFAULT)
#grad_y = cv2.Scharr(gray,ddepth,0,1)

abs_grad_x = cv2.convertScaleAbs(grad_x)   # converting back to uint8
abs_grad_y = cv2.convertScaleAbs(grad_y)
gray = cv2.addWeighted(abs_grad_x,0.4,abs_grad_y,0.4,0)
'''

### Bluring
#image1 = cv2.medianBlur(gray,5)
#image1[image1 > 100]= 255
#image1 = cv2.GaussianBlur(image1,(31,13),0)
color_offset=105
gray[gray >= color_offset]= 255
gray[gray < color_offset ] = 0      #black

cv2.imwrite("/home/pi/Desktop/snaps2.jpg",gray)
```

```
Python 3.5.3 (/usr/bin/python3)
```

Using open CV library to enhance image

E. **Result**



some form by injected
humour or randomised
words which don't look
even slightly believable.
If you are going to use a
passage of Lorem Ipsum
you need to be sure

Clicked Image by Raspberry Pi camera

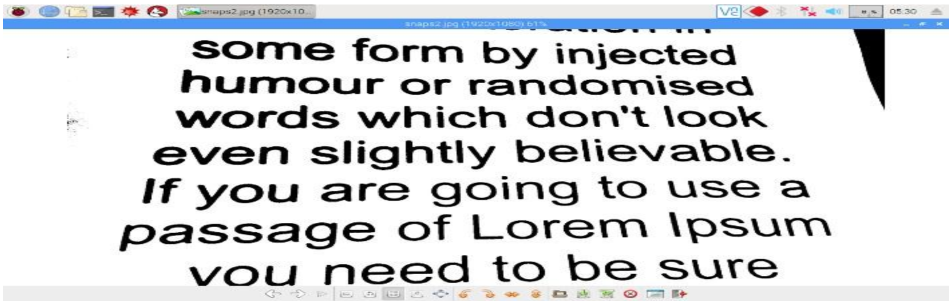
DOI 10.17148/IJIREEICE.2020.8311

IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering

Vol. 8, Issue 03, March 2020



some form by injected
humour or randomised
words which don't look
even slightly believable.
If you are going to use a
passage of Lorem Ipsum
you need to be sure

Enhanced Image with OpenCV library

V. CONCLUSION

With the help of this process, we have got represented an enhanced system to scan a written text to help the visually impaired people, to extract text from extremely advanced backgrounded documents, we've got projected a completely different text localization with models of stroke orientation and edge distributions. Block patterns project the projected feature maps of a captured image element into a featured vector. Adjacent character grouping is performed to calculate the number of elements of text ready for text classification. Optical Character Recognition (OCR) is used to perform character recognition on the captured text regions and convert into audio output for visually impaired people. During this process, the capturing element acts as input for the project. Because the Raspberry Pi has a high-powered camera which starts to stream. The streaming elements are displayed on the peripheral device victimization with Graphical User Interface (GUI) application. Once the required text document for reading is placed in front of the capturing element then the capture button is clicked to capture the image and transfer to the Raspberry Pi board. Then with the help of TESSERACT library function, the image are converted to knowledge and it is detected from the captured element are going to be displayed on the standing bar. The obtained knowledge are converted to be listened with the help of headphones using Text- to-speech synthesis.

REFERENCES

[1]. Ms.AthiraPanicker Smart Shopping assistant label reading system with voice output for blind using raspberry pi, Ms.Anupama Pandey, Ms.Vrunal Patil YTIET, University of Mumbai

[2]. Dimitrios Dakopoulos and Nikolaos G.Bourbakis Wearable Obstacle Avoidance Electronic Travel Aids for Blind IEEE Transactions on systems, man and cybernetics, Part C (Applications and Reviews). Vol. 40, issue 1, Jan 2010.

[3]. William A. Ainsworth A system for converting English text into speech IEEE Transactions on Audio & Electroacoustics, Vol.21, Iss:3, Jun 1973

[4]. Michael McEnancy Finger Reader Is audio reading gadget for Index Finger IJECCE Vol. 5, Issue 4 July-2014.

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

DOI 10.17148/IJIREEICE.2020.8311