

# INTELLIGENT CHARACTER RECOGNITION SYSTEM FOR VISUALLY IMPAIRED PEOPLE



## **Submitted To**

Dr. Md. Hasanuzzaman Sagor  
Assistant Professor  
EEE Department, BRAC University

## **Submitted By**

Md. Sabbir Alam  
ID: 11121067  
Bhuiyan Washek Faisal  
ID: 11121078  
Mohammad Al Imran Hasan  
ID: 11121070



# **Intelligent Character Recognition system for Visually Impaired People**

A Project submitted to the Department of Electrical and Electronic Engineering of  
BRAC University in partial fulfillment of the requirements for the Bachelor of  
Science degree in Electrical and Electronics Engineering

## **Supervisors:**

1. Dr. Md. Hasanuzzaman Sagor  
Assistant Professor  
Dept. of EEE, BRAC University
2. Mr. Avijit Das  
Former Lecturer  
Dept. of EEE, BRAC University

## **Project Members:**

1. Md. Sabbir Alam  
ID: 11121067
2. Bhuiyan Washak Faisal  
ID: 11121078
3. Mohammad Al Imran Hasan  
ID: 11121070

## DECLARATION

We hereby declare that the project title “Intelligent Character Recognition (ICR)System for Visually Impaired People” submitted to the Department of Electrical and Electronics Engineering has been written on the works and result found by us. The work has not been presented elsewhere for project.

Signature of Supervisor

---

Dr. Md. Hasanuzzaman Sagor

Signature of Authors

---

Md. Sabbir Alam

---

Bhuiyan Washek Faisal

---

Mohammad Al Imran Hasan

## ACKNOWLEDGMENT

We are very happy to introduce our intelligence character system with the grace of almighty Allah. There were many people who helped us in many ways and there were some whom extra acknowledgment needlike Mr. Avijit Das (Former Lecture of EEE Department, BRAC University) Sir who took us in when we were struggling to find out our suitable topic. When there was no direction, he showed us the way. He inspired us to make a system to help other people who needed it most. His guidance made us successful in the end.

There were few problem we faced on the way to develop this project like making a software. Zuhair Mehtab Ali is among those who shared his knowledge with us to enhance our understandings in this project.

Our team mates were very patient, supportive, effective and cooperative. We were very lucky to be together and make this project work. Our team mates were always eager to advance that system so that it actually can help visually impaired people.

We can't thank enough to Dr. Md. Hasanuzzaman Sagor (Assistant Professor of EEE Department, BRAC University) Sir who is very punctual, supportive, enlightened and patient. He took us in his wing when we badly needed his help & support. Whose continuous positive support, guidance and help has effectively lead to the completion of our thesis. His faith in us and motivation has provided us with the confidence to achieve our goals.

In the end we would like to thank our parents and well-wishers for supporting and encouraging us throughout our work and university life.

## ABSTRACT

The project is to develop a system, which use optical character recognition (OCR) software to detect words, line and paragraph, pronounce that accordingly. Furthermore to help a visually impaired people. We also added few extra features like voice control switching which execute with respect to given command. This system will help blind people with their daily necessary things. Seeing rising number electronic device made in different aspect of life, ICR system help listening text which are written on paper and help household switching control. Current systems, however, have problems with complexity, high costs, non-open sources and multiple incompatible standards. This project intends to design an open source, reasonably priced and way easy to use ICR, which will be done by interfacing the open source Arduino, Raspberry Pi microcontroller; creating a simple, easy-to-use system to help blind people. The project was carried out in quite a few stages. Firstly, Arduino is the first set for rolling paper on the tray for taking image of the document. Next, webcam will take image to the Raspberry Pi and matching image with its own library for filter and processing its word, text, line and paragraph. Intelligent character recognition is specially used for recognizing typed letter. Last of all, it will convert image as an audio file and gives output to the headphone. Blind people can listen the audio. We used power bank for its power. There are more extra feature added for better communication like home switching like controlling light on/off and fan on/off with the help of voice command.

## TABLE OF CONTENT

Declaration-----	i
Acknowledgment -----	ii
Abstract -----	iii
List of Figures -----	viii
Abbreviation -----	ix
<b>CHAPTER 1</b>	
1.1 Introduction-----	1
1.2 Motivation -----	1
1.3 Time to Rise -----	2
1.4 Project framework -----	2
<b>PART I</b>	
<b>CHAPTER 2</b>	
<b>Hardware Component</b>	
2.1 Introduction-----	3
2.2 Component Used-----	3
<b>CHAPTER 3</b>	
<b>Hardware Working Procedure</b>	
3.1 Image to audio processing-----	11
3.2 Rolling In -----	11
3.3 Rolling Out -----	12
3.4 Voice Command -----	13

**PART II****CHAPTER 4****Line finding Algorithm**

4.1 Line Finding Algorithm-----	14
4.2 Basis: Regression analysis-----	17
4.3 Text Row Accumulation Algorithm (Line finding Algorithm) -----	23
4.4 Basic Procedure (Page skew detection)-----	23
4.5 Basic procedure [Text line Detection independent of Skew]-----	24

**CHAPTER 5****Baseline Fitting**

5.1 Baseline Fitting-----	28
---------------------------	----

**CHAPTER 6****Fixed Pitch Detection and Algorithm**

6.1 Fixed Pitching Detection and Algorithm -----	30
--	----

**CHAPTER 7****Proportional Work Finding**

7.1 Proportional work Finding -----	31
7.2 Practical limitations-----	32
7.3 Tesseract Solution -----	33

**CHAPTER 8****Word Recognition**

8.1 Word Recognition -----	34
8.2 The Extraction & Recognition of texts from multiple image-----	35
8.3 Complexity reduction -----	36
8.4 Word recognition Algorithm -----	40
8.5 Voice command-----	40

**CHAPTER 9****Conclusion**

9.1 Project budget and timeline -----	42
9.2 Project Risk Analysis-----	43
9.3 Sustainability of project outcomes -----	43
9.4 Related work-----	44
9.5 Conclusion -----	44
<b>APPENDIX</b> -----	<b>45</b>
<b>APPENDIX 2 CODES</b> -----	<b>47</b>
<b>BIBLIOGRAPHY</b> -----	<b>57</b>



## List of Figure

Fig: 2.1 Arduino -----	4
Fig: 2.2 Raspberry Pi -----	5
Fig: 2.3 Relay -----	6
Fig: 2.4 Web Cam -----	6
Fig: 2.5 Not Gate -----	7
Fig: 2.6 Potentiometer -----	7
Fig: 2.7 Dry Cell -----	8
Fig: 2.8 Motor -----	8
Fig: 2.9 MOSFET -----	9
Fig: 2.10 Breadboard -----	9
Fig: 2.11 LED -----	10
Fig: 2.12 Motor Gear -----	10
Fig: 3.1 Image to audio processing -----	11
Fig: 3.2 Rolling In -----	12
Fig: 3.3 Rolling Out -----	12
Fig: 3.4 Voice Command -----	13
Fig: 4.1 Regression Analysis -----	17
Fig: 4.2 Hough Transformation -----	18
Fig: 4.3 2-D Hough space -----	19
Fig: 4.4 Distance and different point and angle of a latter. -----	19
Fig: 4.5 Different points P. -----	20
Fig: 4.6 Six possible Line for each point. -----	20
Fig: 4.7 Identify text rows direction. -----	22
Fig: 4.8 -----	24
Fig: 4.9 Connected Components/Blobs -----	24
Fig: 4.10 Upright Bounding Box of each Blob -----	25
Fig: 4.11 Filtering Blobs by size -----	26
Fig: 4.12 Text row & overlap detection -----	26
Fig: 4.13 -----	27
Fig: 5.1 Base Line Filtering -----	28
Fig: 5.2 Cluster Identification -----	29
Fig: 5.3 -----	29
Fig: 6.1 Fixed Pitch word -----	30
Fig: 6.2 Chopping Fixed Pitch Word -----	30
Fig: 7.1 Ideal Space between word and Character -----	32
Fig: 7.2 horizon gap between boundary boxes -----	32
Fig: 7.3 Different Type Baseline -----	33
Fig: 8.1 Pitch Character -----	34
Fig: 8.2. 10/12 Square Pixels -----	36
Fig: 8.3 Statement Identification -----	38
Fig: 8.4 Outer Most point of "Q" -----	40

## Abbreviation

ICR - Intelligence character recognition.

AI - Artificial intelligence.

OCR – Optical character recognition.

---

## CHAPTER 1

Blind people facing many obstacle to coup up with society, ICR system try to help and eradicate with their reading and switching problem. Due to technological advantage, ICR system help to build themselves so that they can contribute to the society.

- Introduction
- Motivation
- Time to Rise

### 1.1 INTRODUCTION

Intelligent Character Recognition (ICR) System used to design to help visually impaired people to understand by hearing them reading material, voice control switching like fan, light etc., identification quick response code. With advancement of technology things are becoming simpler and easier for us. With the help of Tesseract technology we can easily develop ICR system to eradicate visually impaired people's blindness. Tesseract is an optical character recognition (OCR) for numerous type of system. It helps visually impaired people through many ways to translate written paper. That also helpsthem to educate, learn new thing. It as wellhelps their daily household switching or knowing the weather report and so on. Blind people always need help reading text, switching, and communication on internet through other but out small system reduce their work load and help standing with dignity among other society members. So that they can contribute more effectively to the society and help themselves.

Our project consists of two different parts to create ICR system. First part is the hardware and second part is software. Software and Hardware part descried what type of new software we make and what kind of device we use on that project.

### 1.2 Motivation

The rapid growing of technology has significantly transformed living standard of our life. On the other hand, visually impaired people almost left behind from benefit of modern technology. Recent studies shows that out of 7 billion people of the world, around 314 million is visually impaired. Among them 45 million people are completely blind, with 19 millions of children. This means someone in our world goes blind in every five seconds and a child in every minute.

---

Almost 90 percent visually disabled people live in the developing countries. Bangladesh being a developing country has nearly 750,000 blind people currently with an addition of 12,000 people each year (Anon, 2017)[1](Nabi, 2017) [2]. Blind people usually use Braille system to read and write. Braille technique is hard to learn . ICR system makes reading easier and interesting. It helps not only reading but also inspire people to do something good.

### **1.3 Time to Rise**

As time is proceeding ahead, technology is developing and evolving every single moment. No one can claim something to be “latest” because the moment someone does so we can see the presence of something newer and better in front of our very own eyes. We endeavor to cope up with the dynamic changes but it is pretty difficult due to lack of availability of adequate and sufficient resources and technology. Two of the basic fundamental intentions of technology are to make things that are not complicated to be understood by the user and makes working of the user more convenient. Things are simple when the interface between human and technology is least complex. It is an outmost effort by scientists to make the interfacing system more and more convenient for the users. Revolution made by the mobile technology made our modern life easier as it endowing with new services and related commerce with more and more availability. It is not the time to fall behind visually impaired people. Our system will give them a new light to their life and makes world more Colorful.

### **1.4 Project framework**

Our Intelligent Character Recognition (ICR) is specially used for recognizing handwriting & typed letter. Our main goal for this project is to use this system for visually impaired people. Our system will take image of the readable document as an input. Then recognize the pattern of the image. It will convert image as an audio file. Visually impaired people can listen the audio. We will use camera to take input. We will use Raspberry Pi, battery as power supply & audio device which will give output.

Our project divided into two part. I) Hardware part and II) Software Part

## Part I

### CHAPTER 2

#### Hardware Component

##### 2.1 Introduction:

In our thesis we made a very sensitive device which must be very pinpoint & efficient. For that reason we were very cautious while choosing our hardware components. We tried our level best to make our device more efficient. While it was a difficult task for us to make more efficient we also tried to make it less costly. We used Rasp Berry Pi & Arduino for our processing system. We also used Web Cam for taking inputs & used speaker or headphones for outputs. While assembling those microprocessor we also used few small components like relay, resistance. Now I would like to describe briefly about those components below.

##### 2.2 Component Used:

Those components we used are given below:

1. Arduino UNO
2. Raspberry PI 3B
3. Relay
4. Web Cam
5. NOT Gate
6. Potentiometer
7. Resistor
8. Dry Cell
9. Motor
10. MOSFET
11. Motor Gear
12. Switch
13. Wire
14. Bread Board

15. LED

16. USB Cable

### 2.2.1 Arduino UNO:

Basically Arduino is a software company, project, and user community that designs and manufactures computer open-source hardware, open-source software, and microcontroller-based kits for building digital devices and interactive objects that can sense and control physical devices. The Arduino Uno is one kind of microcontroller board which based on the ATmega328. It has 14 digital input/output pins, 6 analog inputs, a 16MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. Arduino/Genuino Uno is a microcontroller board based on the ATmega328P. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. You can tinker with your UNO without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.

“UNO” is an Italian word which means one. This “UNO” was chosen to mark its release of Arduino Software 1.0 which is reference of Arduino series. This Uno board was the first board of Arduino which called as Version 1.0. It has many evolved versions now (Arduino.cc, 2017)[9].

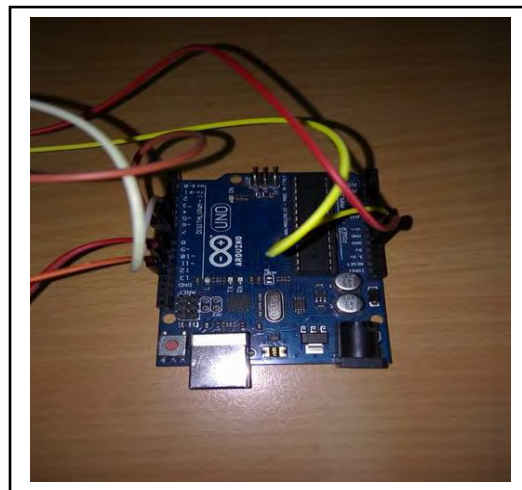
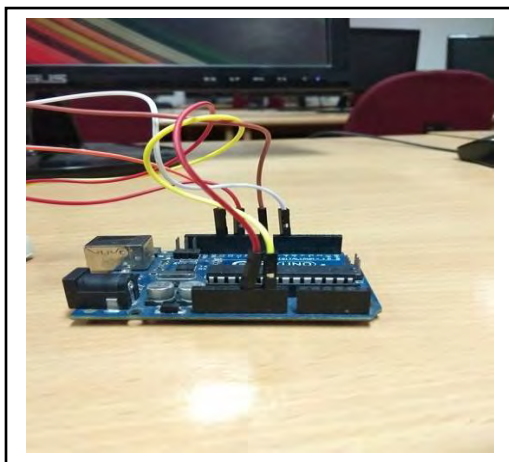


Fig: 2.1 Arduino

### 2.2.2 Raspberry Pi 3 Model B:

The Raspberry Pi is a series of small single-board computers which was developed in UK by the Raspberry Pi Foundation to provide teaching of basic computer science in institutions and in developing countries.

Raspberry Pi's 2<sup>nd</sup> generation is called Raspberry Pi 3. Which consists 900MHz quad-core ARM Cortex-A7 CPU and 1GB RAM. It has several features. 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 and video Core IV 3D graphics core are its main feature. It's also an ARMv7 processor rather than an ARMv6 processor, which limited the software available on the original hardware (Raspberry Pi, 2017) [8].

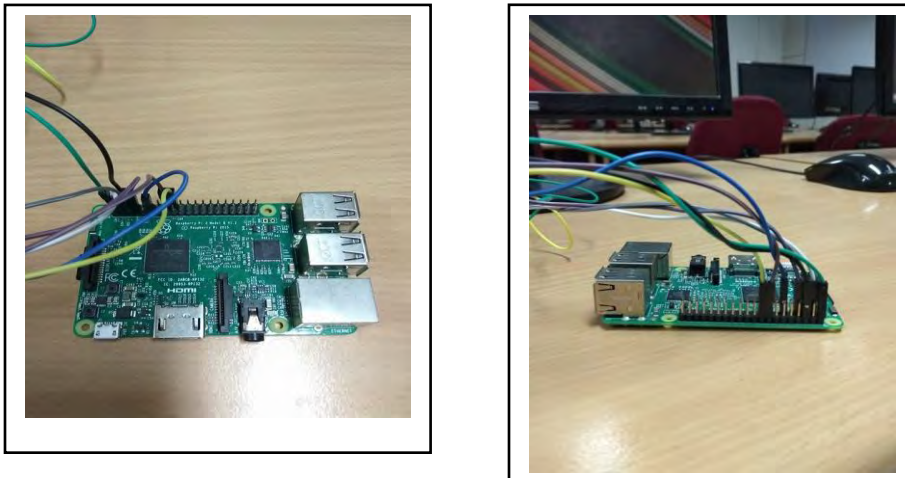


Fig: 2.2 Raspberry Pi

### 2.2.3 Relay:

Relay is a kind of switch which open and close circuits electromechanically or electronically. It mainly controls circuit by opening & closing contact in other circuit. We used 2 Channel 5V Opto Isolated Relay Module which is a very simple relay board. It works in 0V to 5V. This relay uses transistor to run itself. On the other hand optocoupler basically separates output. We can easily use it in high voltage like in DC we can use 30V & for AC we can use it in 250V devices. Most important part is we can use it in Arduino or microprocessor. It has 3 pins output. Which are NO, COM, NC. NC & COM pin normally connected with each other. NO pin normally open relay circuit.

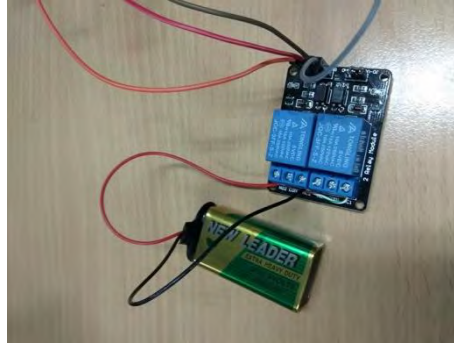


Fig: 2.3 Relay

### 2.2.4 Web Cam:

Our Webcam has resolution of 8 megapixel. It also consists a microphone which can take audio signal as input. It can give fluent video transmission without any lag & also can work in low light. It has a USB port which can be connected in any USB supportable device or microprocessor. We connected our webcam with Raspberry Pi.



Fig: 2.4 Web Cam

### 2.2.5 NOT Gate:

Our NOT Gate is 74HC04N which is a Hex Inverter. Its input has clamp diodes for that reason it can unable current limiting resistor which control excess voltage. It has CMOS input & CMOS output level. It has total 14 pins & one pin is input pin. It can supply 50mA.



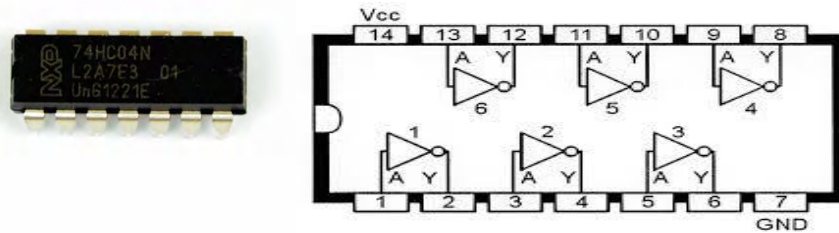


Fig: 2.5 Not Gate

Supply voltage minimum 2V & maximum 6V. It can give 5.2mA as output. It is a very sensitive device which need proper precaution while handling.

### 2.2.6 Potentiometer:

It is a kind of resistor which is adjustable from 50-10000 Ohm. It has 3 pins. Middle pin decides the output voltage & the rest 2 pins connect with resistive element. Potentiometer works linearly. It is very good to use in Bread board & microprocessors. We used potentiometer to control our motor speed. It controlled supply voltage in motor.



Fig: 2.6 Potentiometer

### 2.2.7 Resistor:

Resistor is an electrical component which regulate flow of current in electrical circuit. It's a very important component in electronics which can be found in every electronic devices. There are various types of resistor. We can find out its resistivity by its color code. It has 2 terminal. We used 3 types of resistor which are 100 Ohm, 1000 Ohm & 10000 Ohm.

### 2.2.8 Dry Cell:

Dry cell is an electrochemical cell that converts chemical energy to electric energy. It's kind of battery. It can store energy in an immobilized electrolyte paste. It has a metal electrode or element carbon which is surrounded by electrolyte paste & enclosed by metal for keeping it untouched from other materials. We used 9V dry cell in our thesis which gave power to motor.



Fig: 2.7 Dry Cell

### 2.2.9 Motor:

We used DC motor in our project. Dc motor is a kind of motor which convert electrical energy to mechanical energy by using direct current. We used brushed dc motor here. It is cost efficient & highly reliable. It can run in various amount of voltage. We used a 5V Dc motor which rotate gear to turn on & turn off pages.



Fig: 2.8 Motor

### 2.2.10 MOSFET:

We used a transistor which is called BD135. It's a NPN Bipolar Power Transistor. There are 3 pins in this transistor which are emitter, collector & base. It can run in 1.5 A & 45V. Here base is gate, collector is drain & emitter is source. When gate pass its saturation region it makes a path

from source to drain. Then it makes the path short source to drain. So it has 0V there. On the other hand when the gate is in saturation level the path from source to drain is off. For that voltage difference it has 5V. We used BD135 as an automatic switch here.

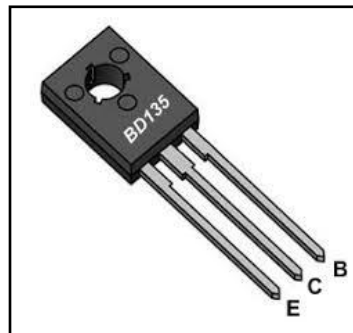


Fig: 2.9 MOSFET

### 2.2.11 Breadboard:

Breadboard is one of the most important components in circuit setup. Breadboard is a kind of solderless device where we can connect other components. We can interconnect all the components in breadboard according to our wish. We just need to change row & column to put components in series & parallel connection. It has very simple procedure to add components, just need to put components terminals in breadboard's hole.

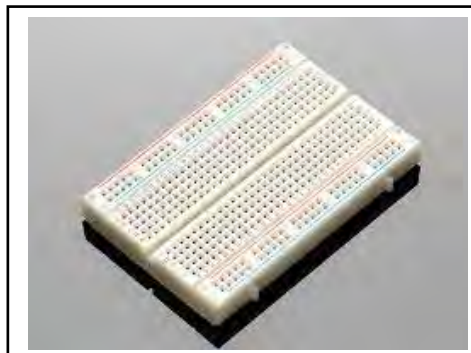


Fig: 2.10 Breadboard

### 2.2.12 Switch:

Switch is a small controlling device. That device can decide whether this circuit connection is on or off. That device can control flow of electron & potential difference between two terminals. We also used switch for the same reason. We used switch to control DC motor.

### 2.2.13 LED:

LED is a small semiconductor which shows light as an output. Its abbreviation is Light emitting diode. Though it is not that much bright, but it shows us whether there is any current flow or not. It made with PN junction. It's not expensive, can be used for decade. Its efficiency is high.



Fig: 2.11 LED

### 2.2.14 Motor Gear:

Gear is a kind of component which can help increase & decrease speed. We also used gear for that same reason. We used 2 gears, one of them is small & another one is big. Small one is connected with motor, another one is connected with roller. We only used 5V motor which has less power. We used gear to increase power. Small gear rotate fast & big gear rotates slow but can gain more power than small one.



Fig: 2.12 Motor Gear

## CHAPTER 3

### Hardware Working Procedure

#### 3.1 Image to audio processing:

Raspberry Pi takes power from adapter. This turns on Raspberry Pi. We also used a switch board which takes input from power. After that we take a 0V input from switch board to Raspberry Pi. Which makes Raspberry Pi's pin number 23 get high. When it gets high it sends a signal to webcam to take an image. Then our webcam takes an image. After taking image it sends that image to Raspberry Pi. Then Raspberry Pi processes that image & converts that to audio file. There is an audio device for giving output. Audio device takes audio file from Raspberry Pi then we can hear that audio from speaker as an output.

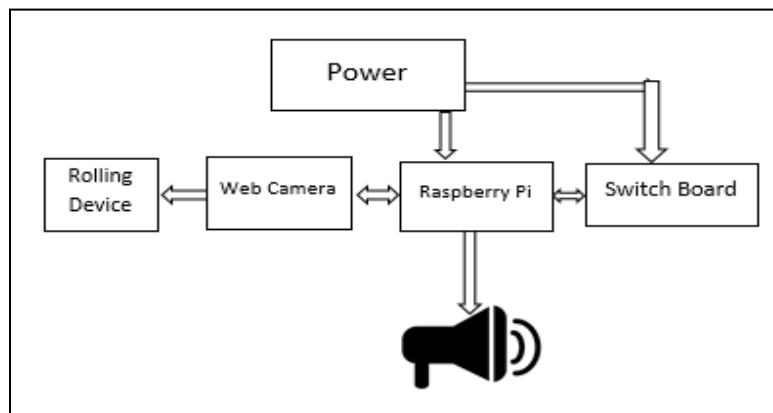


Fig: 3.1 Image to audio processing

#### 3.2 Rolling In:

In that step we tried to roll in the paper which will be given by user. We used Raspberry Pi, Arduino, Motor, switch board & relay to roll in. That entire component took power from source except relay. Relay took power from Arduino. After that we take a 0V input from switch board to Raspberry Pi. Which makes Raspberry Pi's pin number 23 get high. When it gets high it sends a signal of 5V to Arduino. That Arduino will turn on the relay switch. That relay has 2 channels. We used 2 common pins for output & made those 2 channels positive negative short. This relay giving motor input. When motor rotate clockwise then the paper rolls in. Then this paper is ready to capture by webcam.

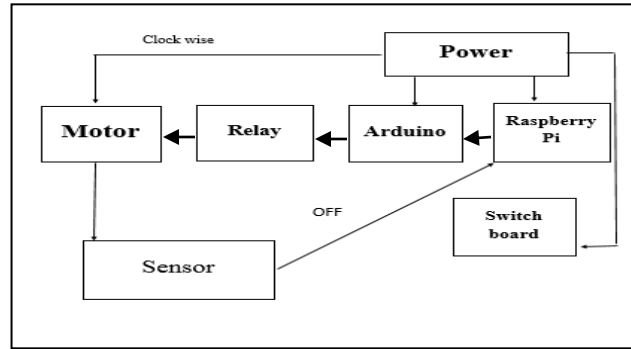


Fig: 3.2 Rolling In

### 3.3 Rolling Out:

In that step we tried to roll in the paper which will be given by user. We used Raspberry Pi, Arduino, Motor, webcam & relay to roll out. That entire component took power from source except relay. Webcam gives a signal to Raspberry Pi. Then Raspberry Pi sends a signal of 5V to Arduino. That Arduino will turn on the relay switch. Relay takes a 5V signal from Arduino. That relay has 2 channels. We used 2 common pins for output & made those 2 channels positive negative short. This relay giving motor input. When motor rotate anticlockwise then the paper rolls out.

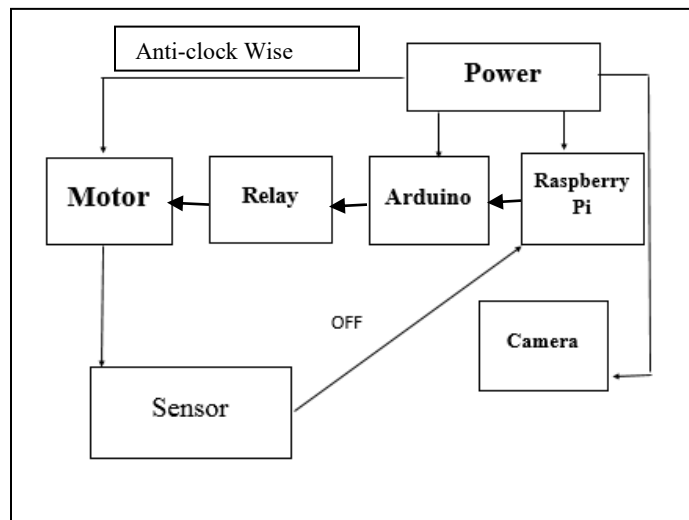


Fig: 3.3 Rolling Out

### 3.4 Voice Command:

We used Raspberry Pi, Arduino, Relay, LED and Microphone for this step. All components took power from source except LED & Microphone. Here Microphone takes our voice command as input. It passes that input to Raspberry Pi. Raspberry Pi processes our voice command into a text message. There is another text message which was set previously in Raspberry Pi. Then Raspberry Pi cross match this input with the other saved message. If that message matches then it passes a signal to relay. Then the relay turns on the LED. Otherwise it doesn't work if that text doesn't match.

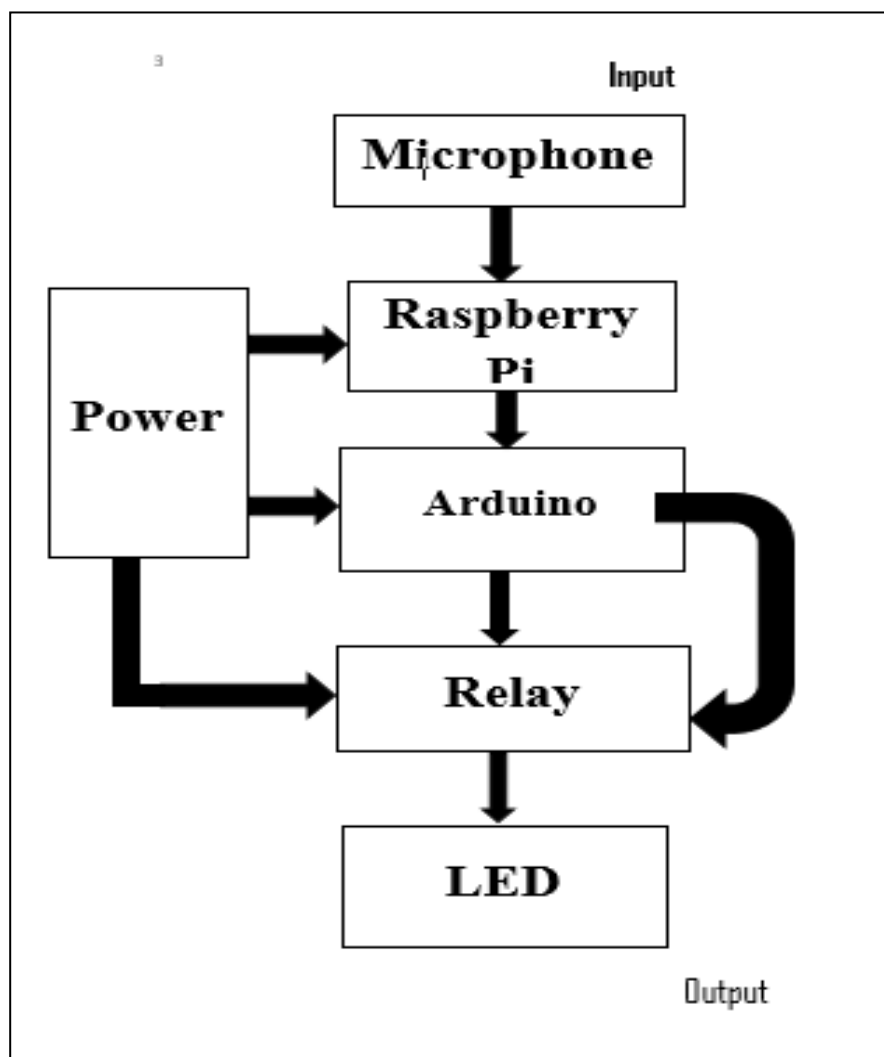


Fig: 3.4 Voice Command

## Part II

Tesseract is one of the accurate Free OCR developed by Google. It's detect pages, skewed pages and identify letters, words, and sentences, pronounce it accordingly. Voice command system helps to control fans, lights of the room or house.

- Chapter 7: Line finding Algorithm
- Chapter 8 Baseline Filtering.
- Chapter 9 Fixed Pitch detection and Algorithm.
- Chapter 10 Proportional Work Finding
- Chapter 11 Word Recognition
- Chapter 12 Voice command

## CHAPTER 4

### Line finding Algorithm

#### 4.1 Line Finding Algorithm:

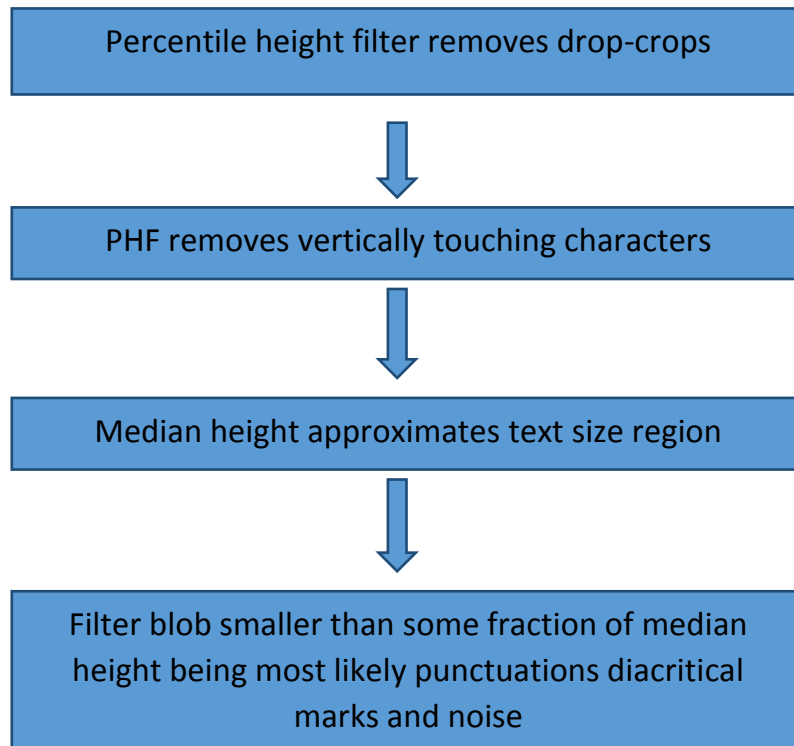
1. Takes an image for input
  2. Identify each object (character) as blob
  3. Detect edges/boundary for the object and test line.
  4. Produce objects with only boundary in same line
- **Detection:**

Skewed page and recognize object without de-skewing

Includes:

- i. Blob filtering:
  - Assume page layout analysis has already provided text regions of a roughly uniform text size. Blob filter processes are

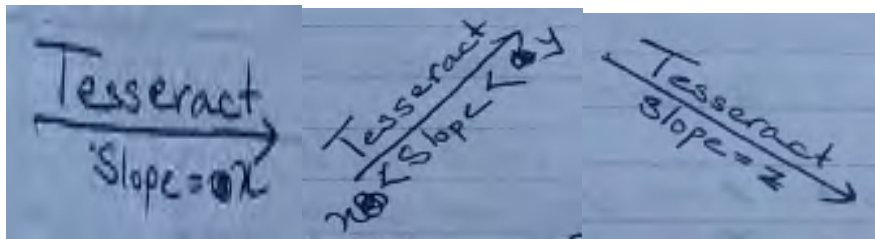




ii. Line Construction(overview):

Identifies blobs filter existing in the same slop (line).

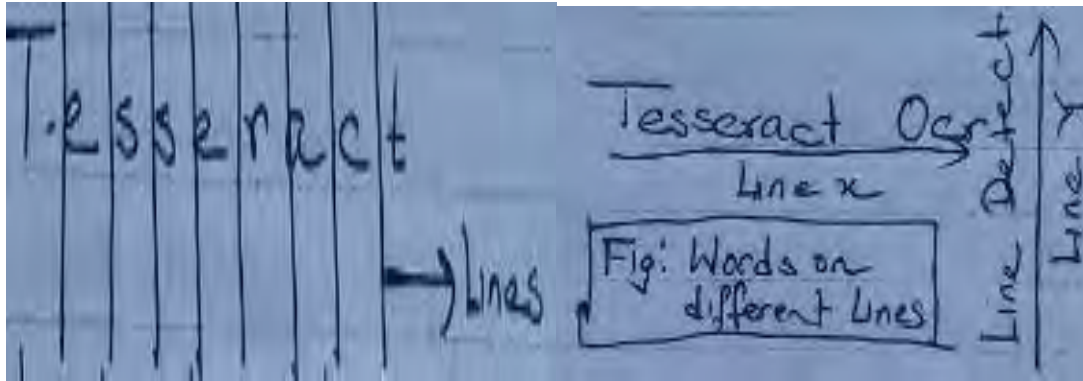
Example:



Input: filtered blobs from previous section.

Process: filtered blobed are most likely to fit a model of

1. Non-overlapping
2. Parallel
3. Sloping Lines.
4. Example:



This indicates that the blobs are a same line i.e. line of characters. It reduces danger of assigning to an in correct text line in presence of a skew. A least median of squares fit is used to estimate the baselines and filtered out blobs are fitted back in appropriate line.

- **Page Layout Analysis (summary):**

Based on “A simple and efficient skew detection Algorithm via text row Algorithm”  
(Ray smith)

- i. Input – page/image contain text
- ii. Process -
  - Finds rows of text in same page
  - Breaks down at skew angles in excess of 15 degrees
  - More accurate, faster than “bairds Algorithm”

- **Detect skew angle from raw bitmaps by extracting feature:**

- i. Fourier transform based method
- ii. Projection profile based method
- iii. Hough transform based method
- iv. Left margin search method

- v. Local region complexity method
- vi. Interline cross correlation method
- vii. Text row accumulation method
- viii. Fractal and least squares method
- ix. Morphology method

**4.2 Basis: Regression analysis** (Estimating relationship between one dependent variable and one or more independent variable)

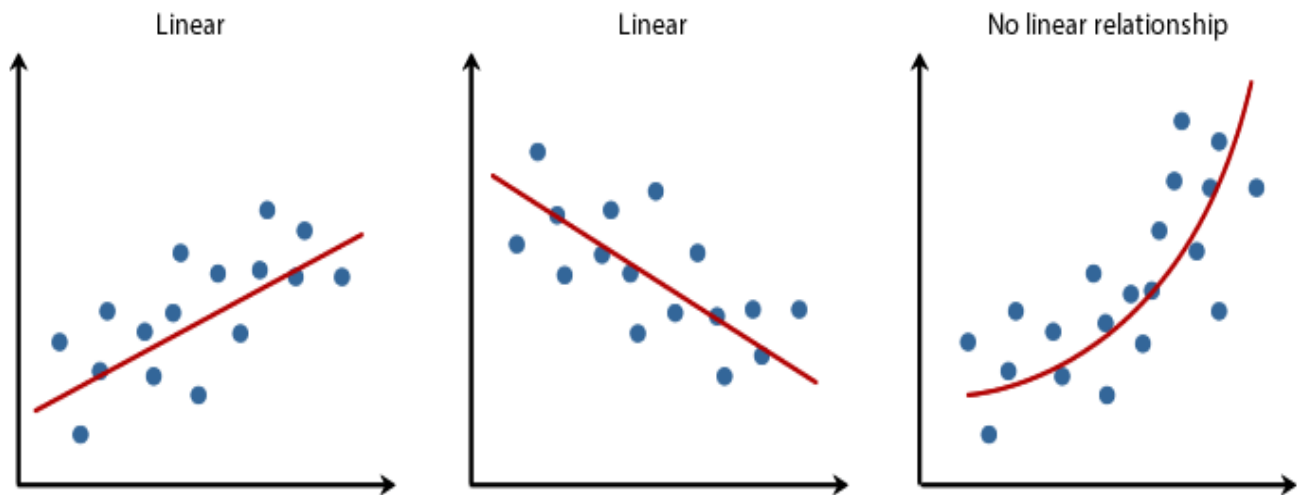


Fig: 4.1 Regression Analysis

- **Reasons for detecting skew of a page:**

Horizontal and vertical projection profiles are spoiled due to skew. Similar image processing algorithm requires text or objects on the page to be aligned horizontal to X-axis.

- This method (text row algorithm) can detect text in a skewed page. Because of this, a page does not need to be de-skewed in order to detect text line in a page.
- This method can detect “skew angle” of each text line.
- This method can estimate skew angle of the line.

▪ **Method of skew detection:**

- I. Simple Hough transformation by applying it to a subset of pixels in an image.

**Hough transform algorithm (for detecting skewed lines):**

- Straight line(  $y = mx + b$  ) can be represented as a point in the parameter space by  $(b, m)$ , that is

$$Y = mx + b$$

Where  $m = \text{slop (angle of indication)}$

$B = \text{intercept from y-axis}$

Polar coordinate system: Parameters:  $(r, \theta)$

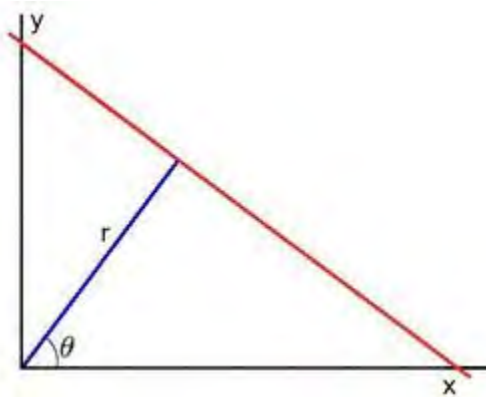


Fig: 4.2 Hough Transformation

- II. For vertical lines.  $m = \alpha$ . So, in order to calculate all possible line Hesse normal form is used where

$$r = x \cos \theta + y \sin \theta$$

Where  $r = \text{closest distance between origin and closest point}(x, y) \text{ on strength line}$

For Horizontal Lines,  $\theta = 0^\circ$

$$\therefore r = x$$

For vertical lines,  $\theta = 90^\circ$

$$\therefore r = y$$

So, using Hesse Normal form, a straight line can be represented as a point  $(r, \theta)$  on two dimensional Hough space.

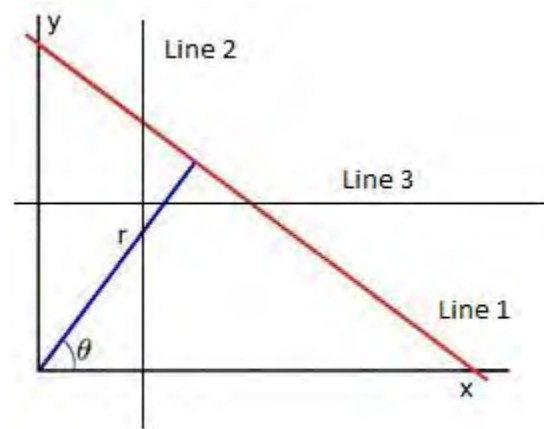


Fig: 4.3 2-D Hough space

Here,

$r_i =$  Distance between origin and point closest to it, on line  $i$ .

$\theta_i =$  angle between  $X$  – axis and  $r_i$ .

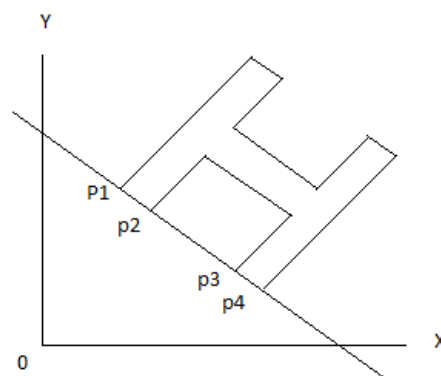


Fig: 4.4 Distance and different point and angle of a latter.

Since,

$$r = x \cos \theta + y \sin \theta$$

$$y = -\frac{\cos \theta}{\sin \theta} x + \frac{r}{\sin \theta}, \begin{cases} \theta \in [0, 180] \text{ and } e \in \mathbb{R} \\ \theta \in [0, 360] \text{ and } e \in \mathbb{R} \end{cases}$$

- For each data point (Edge pixel) a number of line (Li-Lr) are plotted going through it at different angles ( $\theta = 0, 30, 60, 90, 120, 150$ )

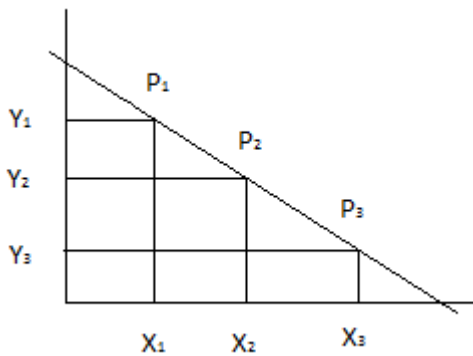


Fig: 4.5 Different points P.

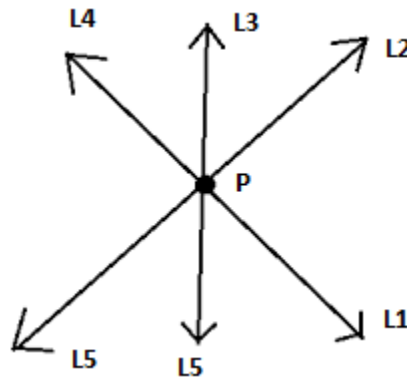


Fig: 4.6 Six possible Line for each point.

For,

$$L1, \theta = 90^\circ \text{ [from origin]}$$

$$L2, \theta = 120^\circ$$

$$L3, \theta = 150^\circ$$

$$L4, \theta = 0^\circ$$

$$L5, \theta = 30^\circ$$

$$L6, \theta = 60^\circ$$

For, P1, when  $\theta = 90^\circ$

$$r = x \cos \theta + y \sin \theta$$

$$r = 40\cos 90 + 40\sin 90 = 40$$

$$\therefore (r, \theta) = (40, 90)$$

$$\text{So, } \theta = 120^\circ, r = 40.6$$

$$\theta = 150^\circ, r = 0.4$$

$$\theta = 0^\circ, r = 40$$

$$\theta = 30^\circ, r = 69.6$$

$$\theta = 60^\circ, r = 81.2$$

Similarly,

For P1,

$\theta$	r
0	40
30	69.6
60	81.2
90	70
120	40.6
150	0.4

For P2,

$\theta$	r
0	57.1
30	79.5
60	81.2
90	70
120	40.6
150	0.4

For P3,

$\theta$	r
0	74.6
30	84.6
60	80.6
90	50
120	6
150	-34.6

Step:

- The length (r) and angle ( $\theta$ ) of each perpendicular line (T) is measured. It is shown in above table.
- The above three steps are repeated for each data point.
- The value ( $r, \theta$ ) at which all the data points (P1-P3) meet, is the line which passes through all the data points (the line to be detected).

Here, for  $\theta = 60^\circ, r \approx 81$ [for p1, p2, p3].

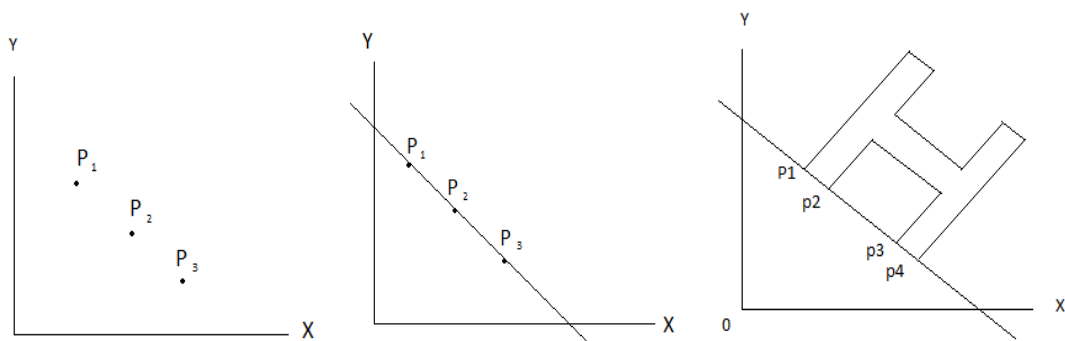


Fig: 4.7 Identify text rows direction.

$\therefore$  The row of the text of the edges of p1, p2, p3 is skewed at an angle  $30^\circ$  from the origin.



### 4.3 Text Row Accumulation Algorithm (Line finding Algorithm)

Features:

1. Find row of the text (group of the blobs). It is possible to obtain an accurate estimate of the skew angle (angle of indication with respect to base line of window) of each line of text.
2. The skew angles of all rows of text helps to estimate the skew angle of the whole page to a high degree of accuracy.
3. Rows of the text can be found independently of the skew angle.

### 4.4 Basic Procedure (Page skew detection):

1. Hough transform algorithm finds the skew ( $r, \theta$ ) and detect straight lines 'Edge Pixels' of an image (p1-p3).

The same technique can be applied to same subset of pixels of an image in order to find rows of text and their skew. The pixels chosen must be:

- Fewer in number [indicates blobs of object in an image]
- Most likely to form straight lines parallel to the baseline of text row [ identifies character in the row]

2. Select some subset of pixels processing the above characteristic.  $P \in \{P_1, P_2, \dots, P_n\}$ .
3. To achieve the desired accuracy, project the points ( $P_i$ ) in parallel with as many directions as possible points.

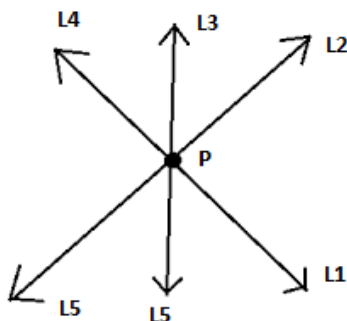


Figure: Possible directions a line can exist with respect to each pixels.

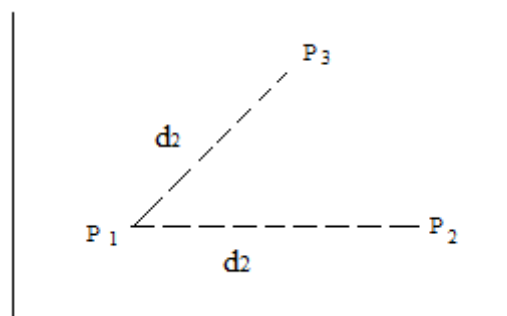


Figure: 2 Possible direction for line to exist for  $P_1$  Parallel to the direction 1 and 2.

4. The direction for which highest spikes are achieved, is the direction in which maximum lines exist. So, that direction is considered as direction of skew of the page.

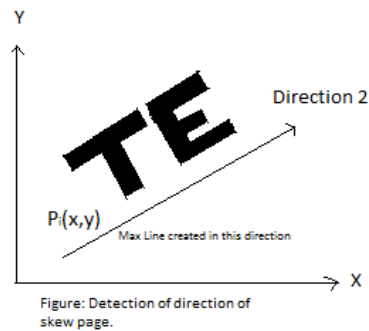


Fig: 4.8

#### 4.5 Basic procedure [Text line Detection independent of Skew]

Find rows of text in a page in the presence of:

- Broker and joined characters
- Speckle noise
- Page Skew, ever in excess of 20 degrees.

Perform connected analysis of the page to find connected components needed for recognition process.

**Connected Component:** A set of pixels that might represent an object (like characters of shapes) and possess a common attribute. (Like pixel of same color clustered together).

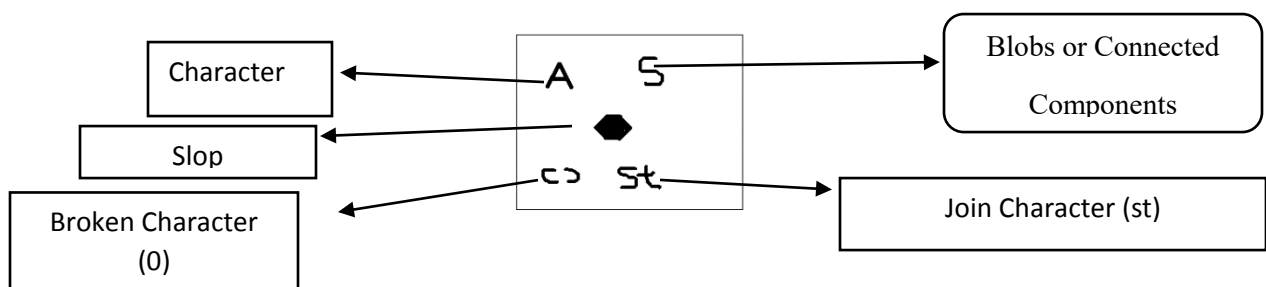


Fig: 4.9 Connected Components/Blobs

The connected components are referred to as blobs. All direction and location of a blob relate to the co-ordinates of its upright boundary box.

Step-1: Find connected components

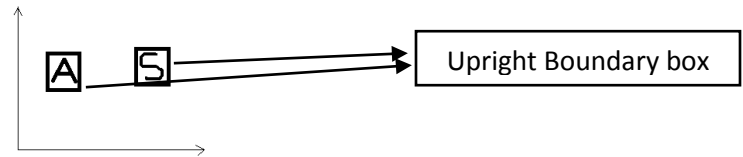


Fig: 4.10 Upright Bounding Box of each Blob

### Character Classify:

Filter the blobs by selecting a subset of Blob's which have a good chance of ne presenting the body text: The key is to filter.

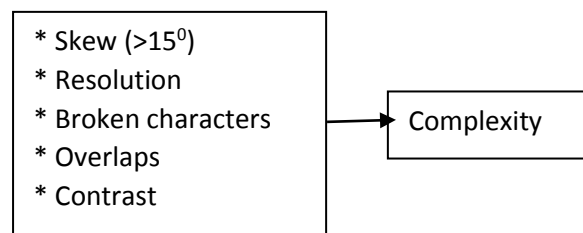
- \*\* Drop-caps
- \*\* Under lives          Access ones
- \*\* Noise

The blobs (accessories) left out in this stage will be put back in their pleaces of text line later.

[During 2<sup>nd</sup> stage analysis]

Step-2: From set of a specific text now, chose the ones which might represent

A text.



- Blobs smaller than a fixed number of pixels in height are removed [as accessories]. The remaining blobs are then fitted by size.
- Blobs with height between (20%-95%) are returned provided their width is not excessive.

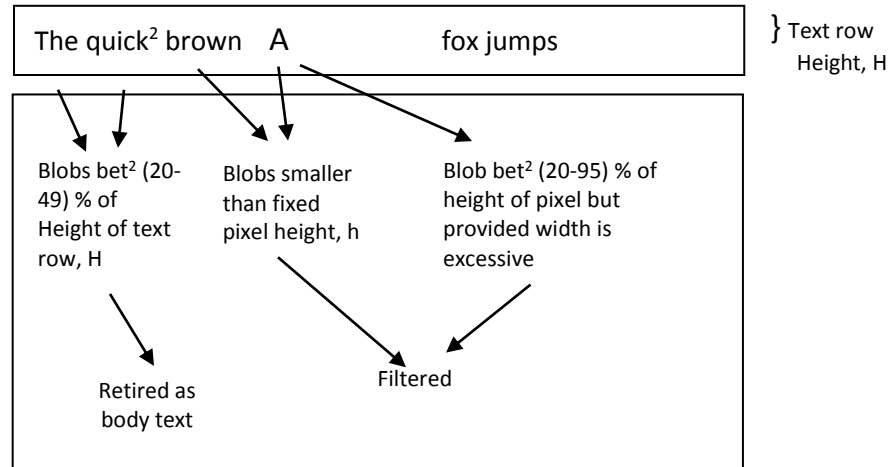


Fig: 4.11 Filtering Blobs by size

- Sort the retained blobs by x-coordinate (of the left edge) as the sort key. This sort arranges the blobs of text in order and makes it possible to track skew across the page and text you.
- Step-3 and 4: Filter and keep only the blobs responding to character in text row. Sort blobs in x-Coordinate to detect skew.
- Make initial rows in order to find the overlapping between multiple blobs of character and gradual vertical shift across the page.

Algorithm:

- Set running average Y Shift (Shift) to Zero
- For each blob in sorted order.
- Find the existing row which has most vertical overlap with the blob

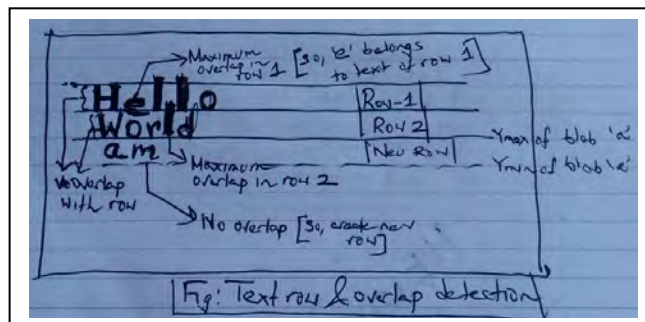


Fig: 4.12 Text row & overlap detection

If there is no overlapping row, then

1. Make a new row and put blob in it
2. Record the top and bottom coordinates of the blob as the top and bottom coordinates of the row.

Else,

1. Add the blob to the row
2. Expand the top and bottom limits of the row with the top and bottom limits of the blob clipping the row height to a limit.

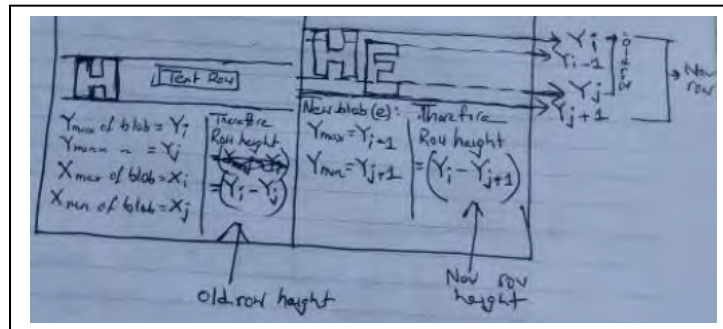


Fig: 4.13

- Update running average y-shift with bottom of the blob. Here, New shift,  $Shift_{n+1} = Shift_N + (1-x) \text{ New Shift}$   
Where,  $0.5 < x < 0.7$  and in easing to large and with inducing number of rows or page.
- End If
- End for

## CHAPTER 5

### Baseline Fitting

#### 5.1 Baseline Fitting

Baseline: Fitting algorithm:

Input: Takes a set of lines of text (text rows)

Process: The lines of text are fitted with the baseline using quadratic spline

Output: Produce previous set of line text fitted with the baseline representing a body of text lines (like paragraph)

##### i) Baseline Fitting:

- This system enabled Tesseract to handle pages with curved baselines which are common artifact in scanning
- The baselines of text row are fitted using a quadratic spline.

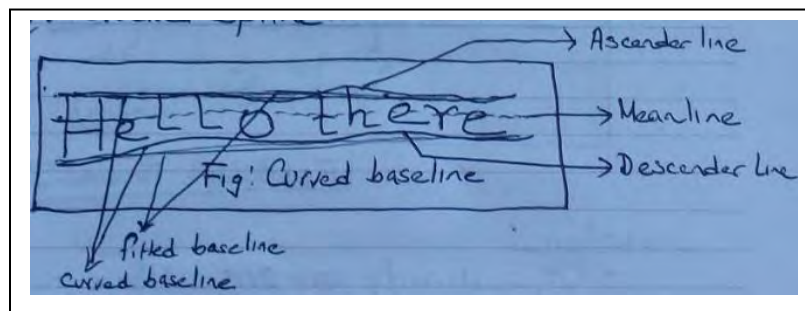


Fig: 5.1 Base Line Filtering

**Algorithm:** [For each Text row]

- Partition the blobs in to groups with a reasonably continuous displacement getting the original straight baseline



Fig: 5.2 Cluster Identification

- A quadratic spline is fitted to the most populous partition (assumed to be the baseline) by least square fit.

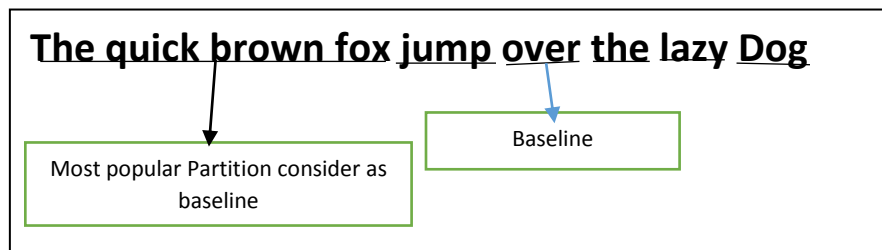


Fig: 5.3

### Limitation:

Discontinuity may arise when multiple spline segments are required [for multiple segments being mostly populated]

**Suggestion:** Traditional cubic spline might work better

Output: Lines of text (Words) with fitted baseline.

## CHAPTER 6

### Fixed Pitch Detection and Algorithm

#### 6.1 Fixed Pitching Detection and Algorithm

**Input:** Takes a Line of text.

**Process:**

- Determine whether blobs are fixed pitch.
- Chop fixed pitched blobs to separate characters using the pitch.
- Disable chopper and associate on those words for the word recognition step

**Output:** Provide words constructing blobs without fixed pitch

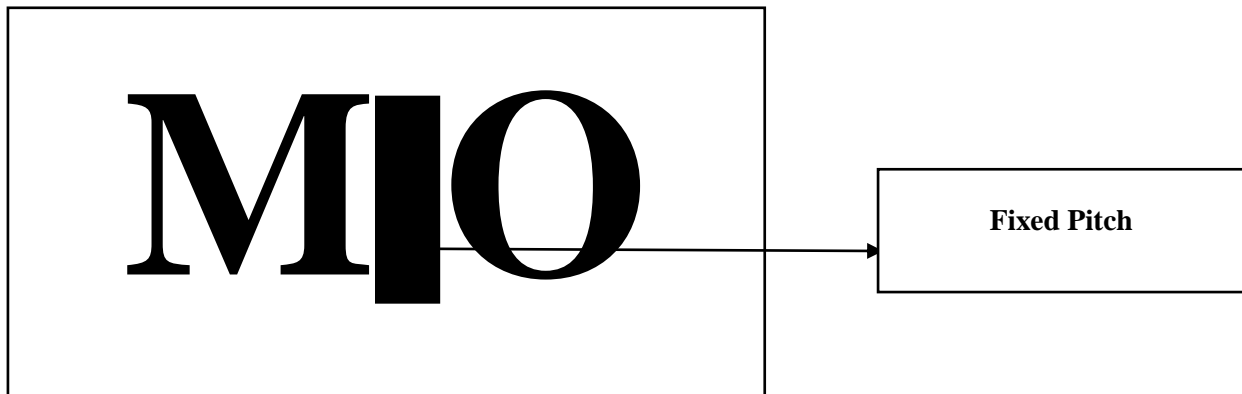


Fig: 6.1 Fixed Pitch word

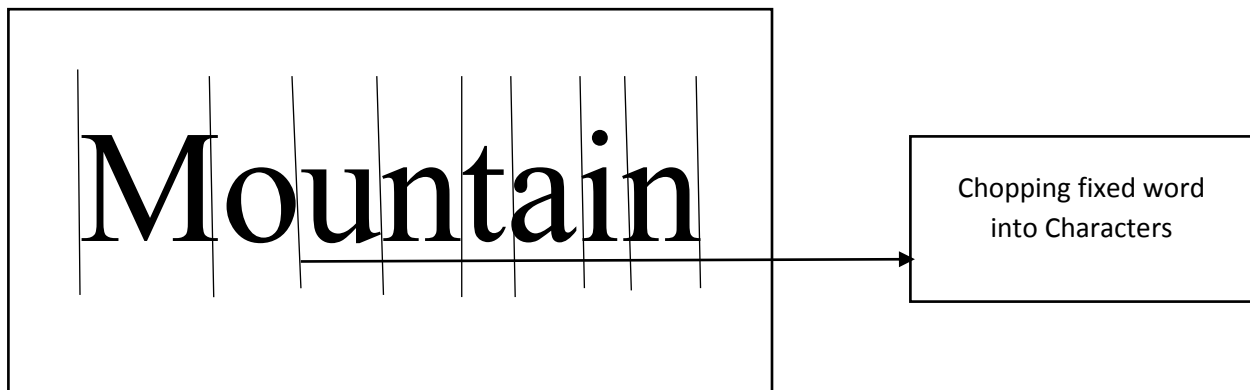


Fig: 6.2 Chopping Fixed Pitch Word

**Output:** Line of text containing words either Confining Fixed Pitch or not.



## CHAPTER 7

### Proportional Work Finding

#### 7.1 Proportional work Finding

**Word finding algorithm:**

**Input:** Takes lines of text passed from fixed pitch Detection and chopping.

**Press:** Identify Words (group of characters) in the line.

**Output:** Produce lines of text each containing a set of words. Each word consist of one of more characters (blobs).

**Main Concept:**

1. Each line of text contains one or more words with gaps ( $G_w$ ) between them.
2. Each word contains are or more characters with gaps ( $G_c$ ) between them.

The gap between two characters can be:

- Either so close that the edges off upright bounding box of the two character overlap.
- Or far enough that the bonding boxes have gap ( $G_B$ ) between them.
- Normally,  

$$0 < G_B < G_c < G_n$$

That is,

- The gap between two character (last for former and first for latter) of two words will be greatest. Then comes gap between two characters of some word and finally gap between bonding box of two character of same words.

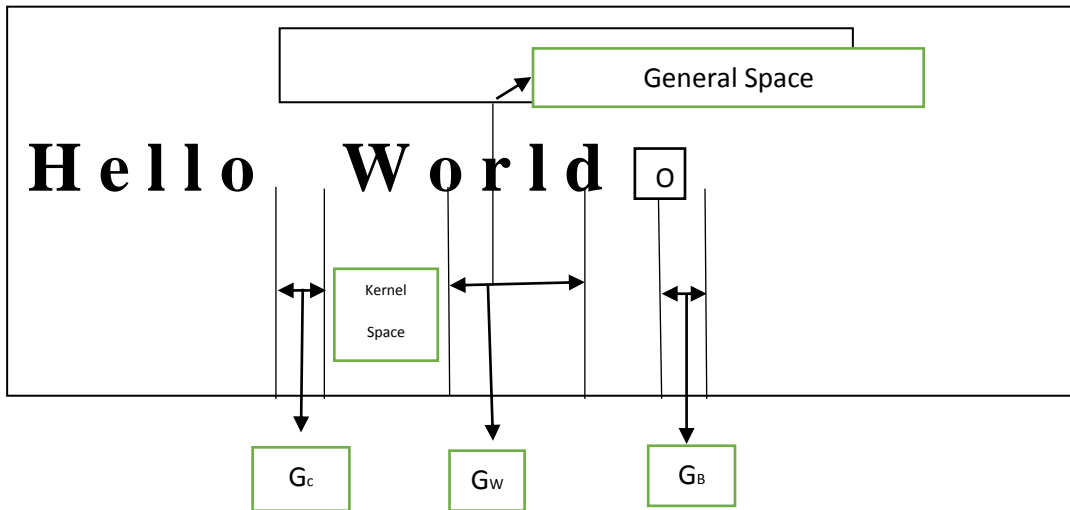


Fig: 7.1 Ideal Space between word and Character

Therefore, gap greater than a specified limit between two characters (blobs) should indicate change of word. The gap, if less than the specified threshold should indicate characters of same word.

## 7.2 Practical limitations:

The gap between two characters of the same word might be equal to the gap between two words. The gap (kernel space) between two characters of the same word characters of the same word can be greater than the general venal space (average gap between charters of same word). There might not be horizon gap between bounding boxes of two characters.

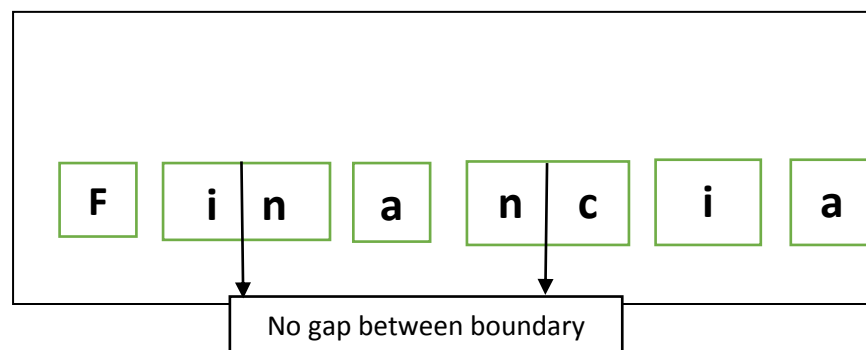


Fig: 7.2 horizon gap between boundary boxes

### 7.3 Tesseract Solution:

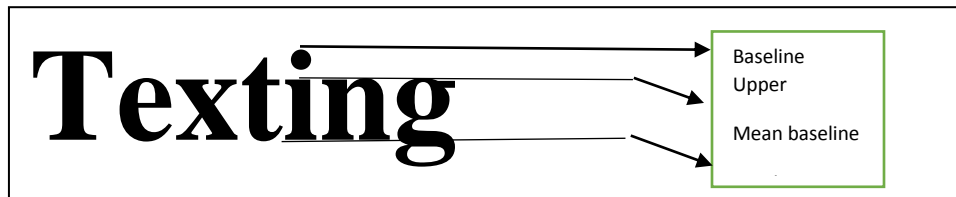


Fig: 7.3 Different Type Baseline

- Baseline upper: The highest vertical limit for characters in a word (The largest character provides it)
- Mean baseline: The near highest vertical height of the characters of word. If,  $H_i$  is the upper Y-coordinate of a character in a word with  $n$  characters. Then  $H_i$  (max) = Baseline upper  

$$\frac{1}{2} \sum_{i=0}^n H_i = \text{Mean baseline}$$
- Tesseract measures the gaps in a limited vertical range between Baseline & mean baseline.

Spaces close to this threshold are made fuzzy, So that a final decision can be made at the end of word recognition (GitHub, 2017)[5][6].

## CHAPTER 8

### Word Recognition

#### 8.1 Word Recognition

##### Word Recognition Algorithm:

**Preprocess:** 2<sup>nd</sup> stage analysis

**Input:** Takes nor fixed-pitch text

**Process:**

- Chops Joined Character.
- Associates broken character.

**Output:** Produce characters to be recognized in static character classifier

##### Fixed pitch character:

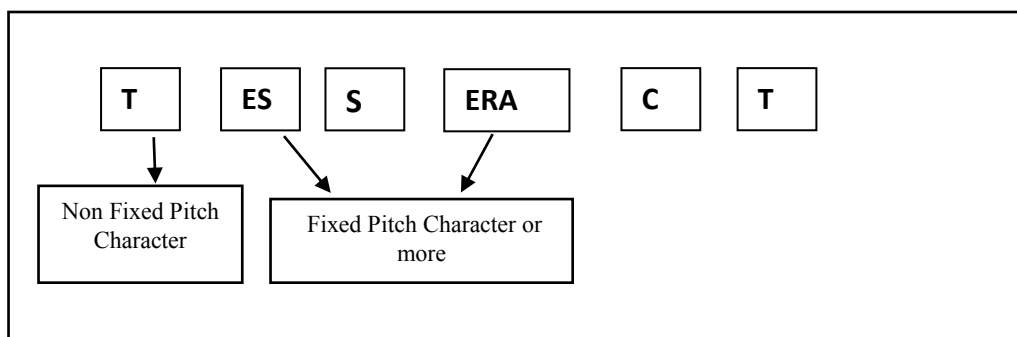


Fig: 8.1 Pitch Character

Detection of two or more characters as one due to lack of space between the characters (Joined characters) is as known as Fixed Pitch character

The first step of this stage is to take the blobs left but at the first stage analysis of identification of blobs that might represent a body text.

**Input:** Blobs that did not become identified as body text in first stage of recognition. The types included in this category are:

- Drop-Caps
- Undefined
- Noise (Induces noise & fixed pitch character)

## 8.2 The Extraction & Recognition of texts from multiple image:

**Process:** The blobs with worst classifier left out in the first stage are chopped from concave vertices, of a polygonal approximation of the outlines, and may have either a concave vertex opposite, or a line segment.

### Polygonal approximation of the outline:

- This method is used to chop blobs and try to identify it as characters.
- Inspired from the two original methods used in OCR(Opensource.com, 2017)[7]:
  - a) **Matrix Matching:** Bit mapped image of a Character is matched directly with a set of templates.
  - b) **Feature Extraction:** Some higher level structure of the character is found and compared with a standard.

**A) Matrix Matching:** Fix a character cell to contain an array of 10 by 12 pixels (square)

- A sample of each known character is presented to the machine to be used as test data.
- Each pixel is digitized by the machine either to black, grey or white using simple the threshold.
- Black region is the interior portion of the character, white is the exterior & grey indicates boundary between black & white region.

- Grey pixels are used to provide flexibility by regarding them as matching both black & white pixels.

These black & white samples of known characters are used to detect any unknown character presented to the machine.

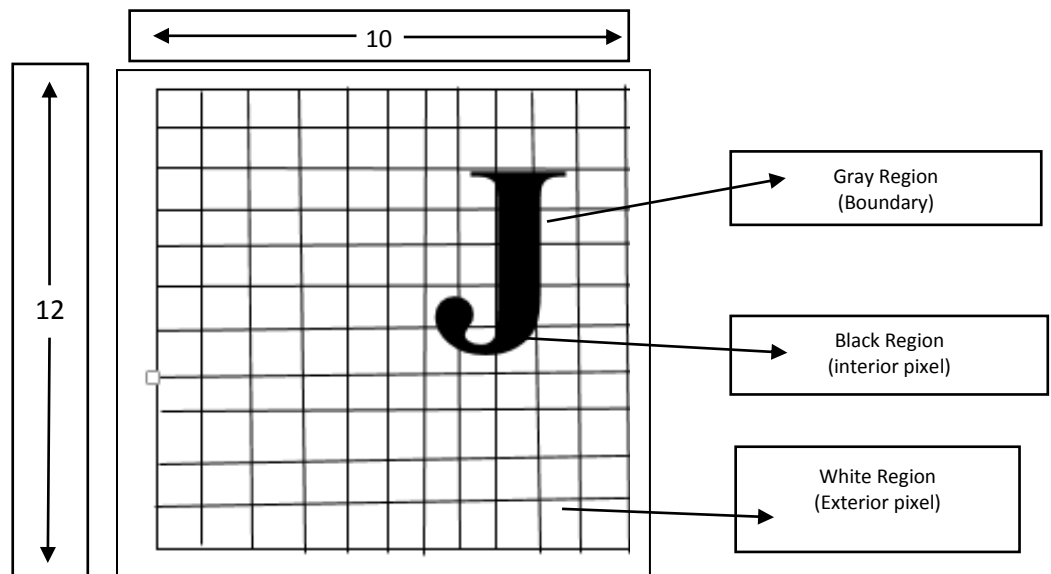


Fig: 8.2. 10/12 Square Pixels

**Character recognition:** When the machine is presented with an unknown character, the pixels are threshold to black & white.

- Corresponding pixels are compared in every template (test data provided earlier)
- The number of pixels matching black for black & white for white are counted.
- Grey pixels are ignored and the unknown character is recognized as the one being with the highest value match.

### 8.3 Complexity reduction:

In order to reduce complexity certain pixel position are identified in test data that helps to distinguish between characters better than the other E.g. some pixels may be white in all text data.

- A large portion of pixels in the test data are changed to grey. Once all templates (test data) have been created, each is compared against all others
- The pixels of one test data that has the greatest discriminating power i.e. helps the most to identify a character are called important pixels.
- The pixels with least discriminating power are set to grey in all template subjects to constraint that all templates remain different.
- In the way, only those points that are capable of distinguishing between characters are retained.
- The reduction results in an average of only 11.3 pixels being tested for each character.

### **Limitations:**

- The accuracy of recognizer decreases with horizontal or vertical deviation of the position of a character.
- Different fonts have different aspect ratio, thickness of line, size, etc. Change in any of that reduces accuracy of recognizing a character.
- Majority of printed text as opposed to type written is proportionally spaced which makes it difficult to locate a character in a page.

### **B) Feature Extraction:**

- Gnamstead et al describes an OCR system that analyses the strokes comprising each character.
- A line finding algorithm segments the characters into straight and curved line sections.
- The length, slope & curvature of each segment are combined with information on the connectivity with other segments into a 40 bit representation.
- The set of these 40 bit representations which make each character is called statement for the character.
- The statement of unknown character is compared with the statements for the standard characters for which the system was trained.
- The character is recognized as being the one corresponding to the statement which best matches the statement of the unknown.

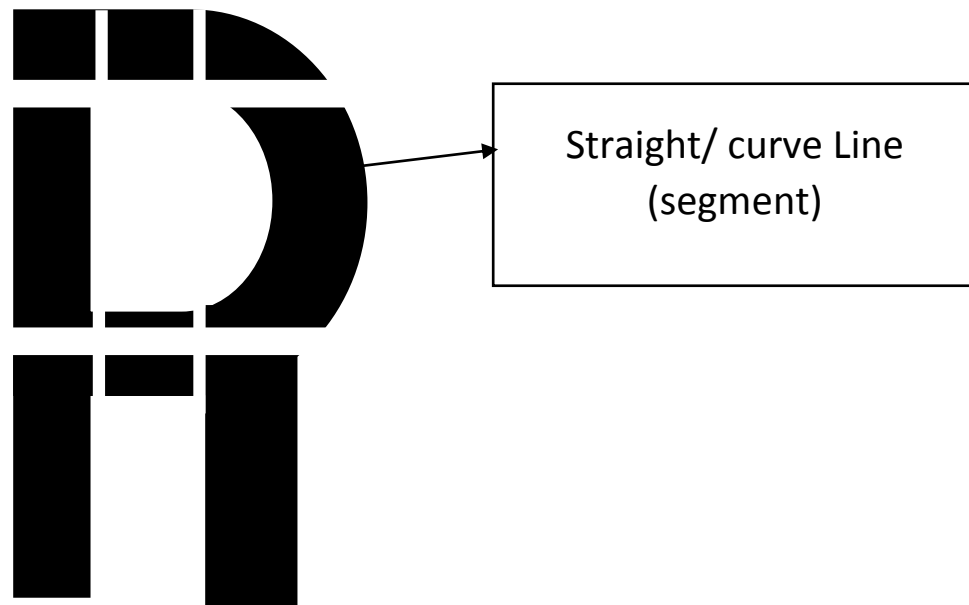


Fig: 8.3 Statement Identification

### Advantage:

- Characters can be recognizing any orientation.
- As long as the slope for all sections deviate from the standard by same amount, the Character can still be recognized.
- Using a Manchester Mark I Computer, with 4000 machine instructions, this system recognized characters at an average rate of 60 seconds per character.
- If features are chosen well enough, recognition of the characters are independent of position size, orientation & font.

### Limitations:

- Recognition time is much stowed than Matrix Matching.
- Steps are complex and hence the process has to be software based rather than hardware.

Example:

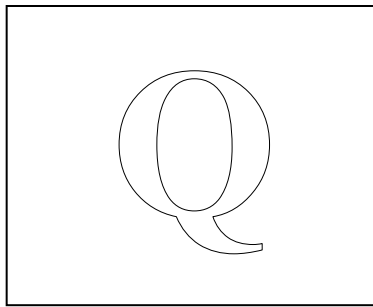


Approcach-1

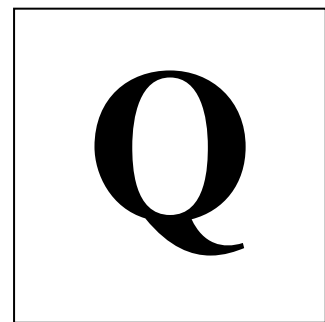
Approcach-2



Binary Image



Outline of Binary image



Tined Versionof binary Image

- Feature extraction applied to a binary Image (each pixel is either black or white) of a character has two possible approaches:
  - a) Find the outline of a character that may be approximated by a polygon to make it easier to manipulate. The result is also a bitmapped Image
  - b) Thin the lines to produce a Skeleton. The result is also a bit mapped Image but the features may still be found easily. The skeleton may be congested to vectors & stored in a graph representation

### Outline of a character:

A set of outermost points of a character that can be used to find and extract convex and concave sections of a character, is outline.

### Outer most point:

Let,

Outline consist of 'n' points with coordinate vectors  $E_i$ , where  $E_i = (X_i, Y_i)$  and 'i' ranges from 1 to n.

- The vectors may have an arbitrary origin, a definite short Point like bottom left corner of the character cell.
- Therefore, outermost point in the direction of vector 'A' is,  $E_i$ , such that  $A \cdot E_i$  is maximum over 'i' from 1 to n

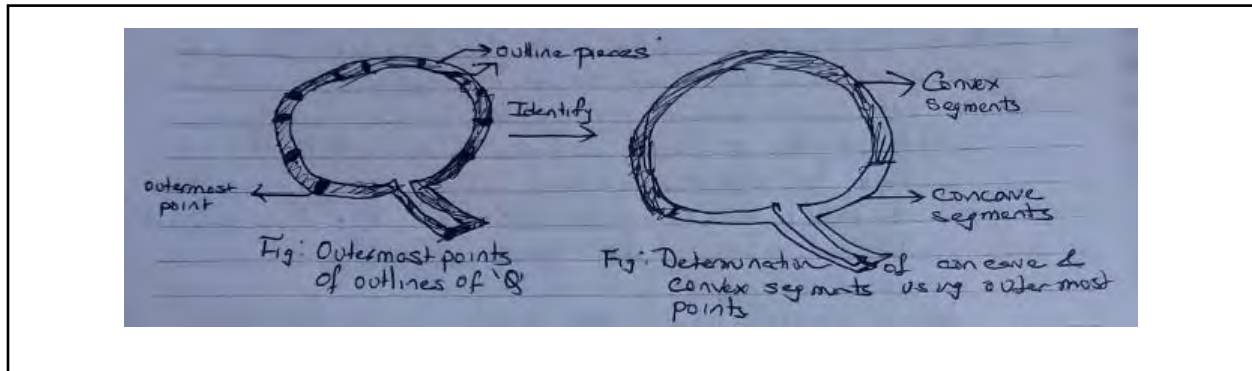


Fig: 8.4 Outer Most point of “Q”

### 8.4 Word recognition Algorithm:

1. Performs 2<sup>nd</sup> stage analysis on fixed pitch character left out during the 1<sup>st</sup> stage. The analysis involves chopping fixed pitch characters & then try to identify the characters.
  2. Chopping fixed pitch characters: Blobs with worst confidence from character classifier are chopped.
    - Candidate chop points are found from concave vertices of a polygonal approximation of the outline.
    - It may have either another concave vertex or a line segment opposite to it.
    - It may take up to their pairs of chop points to successfully separate joined characters form ASCLL selection.
    - Chops are executed in priority order. Any chop that fails to improve the confidence of the result is undone but not completely discarded for future use.

#### 3. Associating Broken Characters:

Takes Chopped & non-fixed pitch characters of the word which did not produce satisfactory result yet(GitHub, 2017)[].

### 8.5 Voice command

Using Python Language we make switching (ON/OFF light, fan etc.) system software. The voice command system is configure with Raspberry Pi 3 Microcontroller. Raspberry Pi 3 PIN GP10 is

being setup for microphone. The software enable python speech recognizer. According to the given speech to the software, it try to matching with the recorded command. When it detect fan or light on or off then it execute affording to the command.

## CHAPTER 9

### Conclusion

#### 9.1 Project budget and timeline

This is the budget for our project now. It can be increased for the demand of necessary parts and software such as arduino etc. Primarily its looks like big thing to carry, but when the project done we will redesign the whole circuit so that it would look like an optical device and comfortable to wear. Most of the hardware are bought from techshopbd (shop link given in the reference) and software are free(En.wikipedia.org, 2017)[3](Techshopbd.com, 2017) [4].

Item Number	Item Name	Price (Taka)	Quantity
1	Raspberry Pi 3	4031.07	1
2	Raspberry Pi 3 Acylic Enclosure	500	1
3	Web Camera	3000.00	1
4	PINENG 10000mAh Power Bank Model: MIS-00396	4400	2
5	Micro SDHC Card Transcend 32GB Class 10 Model : MIS-00325	900	1
6	Wifi Module	700	1
7	Audio device (earphone Bluetooth)	3000	1
8	Different type cables	200	1
9	LDR (sensor)	10	1
10	Slide switch(power)	30	2
11	Breadboard	240	2
12	Body	2000	1

13	Arduino Uno	3000	1
14	DC Motor 9V	80	1
15	Relay	70	2
16	Camera Lens	2000	1
	<b>Total</b>	24161.07	

## 9.2 Project Risk Analysis

At first ICR machine will take more space than a mobile phone. When we finish completing project we need some time to make ICR device small so that it can easily fit on the eye like eyeglasses or optical. We are using raspberry Pi 3 to implement this project but then again there are some difficulties we might face such as the RAM of Raspberry Pi is 1GB, it can slow down the starting time line of output.

## 9.3 Sustainability of project outcomes

As our project now finished, it is based on only English language and in future our goal is to work on other languages particularly like Bengali. The hardware and software of the project will help in the long run. In addition it gives advantage to make our device more sophisticated, accurate, and enjoyable. This is one kind of device which can eradicate curse of blindness & decrease dependency on Braille system. Intelligent Character Recognition system directly helps mass people to understand reading material without taking help from others. As we know that most of the visually impaired individuals in our society (Bangladesh) have to live on begging. They live their life without hope, honor and future. Some visually impaired people spending their life being governed by their family. Intelligent Character Recognition can change the reality not only for visually impaired folks but also bring a change in the point of view of our society. Using this device that they could easily learn, listen, acquire knowledge, get empowered, shape their future and contribute to the society. This ICR device cannot be measured by its advantages and contribution because it can open a new horizontal possibilities, which were inconceivable beforehand to visually impaired people.

## 9.4 Related work

As we know English language Optical character recognition (OCR) is to read only when papers are perfectly scanned, voice command we get from mobile or internet, facial recognition and command switching from other kind of software developed by many other people and company around us. But there are nothing created like we developed here. We tried to create and accumulate all different types of idea into small, beautiful and perfect in a platform so that the visual impaired people would get benefited from it. This way we hope they will overcome their mental, visual struggle and contribute to society.

## 9.5 Conclusion

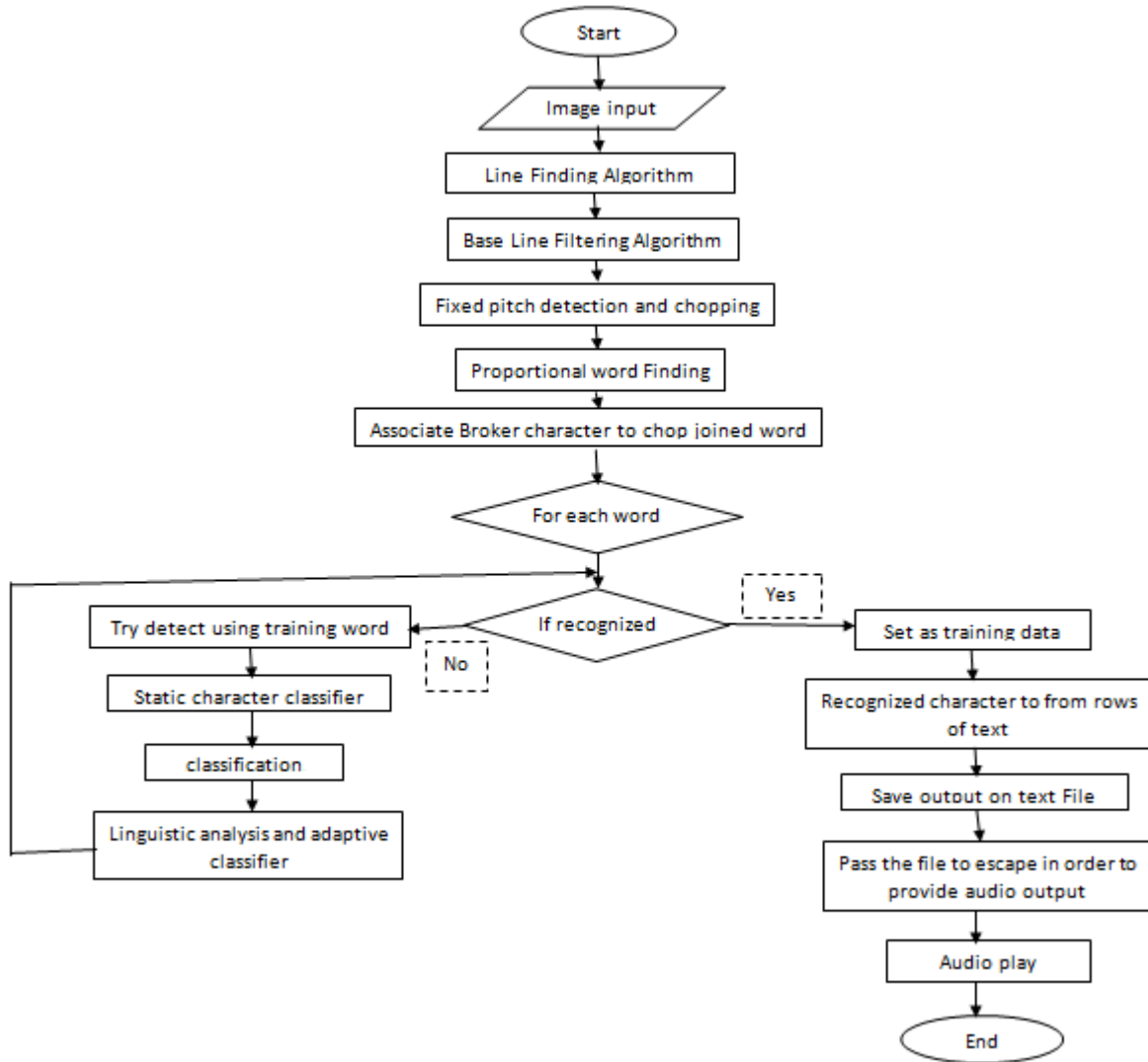
Image processing, image analysis and machine vision are most closely field to computer vision i.e. the basic techniques that used and developed in these fields are more or less identical. Combining techniques in these fields can be applied in various applications like Pattern/Object Recognition, Imaging, and Machine Learning etc. Among these applications, pattern recognition and image processing can be used to aid blind people in daily tasks by extracting information from outside environment.

In this paper design and implementation of prototype of text detection and recognition system has been discussed. After combining different techniques for Text detection and extraction. Text detection followed by recognition using supervised pattern recognition algorithm not only improves accuracy but also decreases delay of the system. After successful recognition, text is converted into audio output. Using this system, Voice Command aid can be developed to assist visually impaired people during navigation, weather checking, to know the known or unknown, and indoor or outdoor environments.

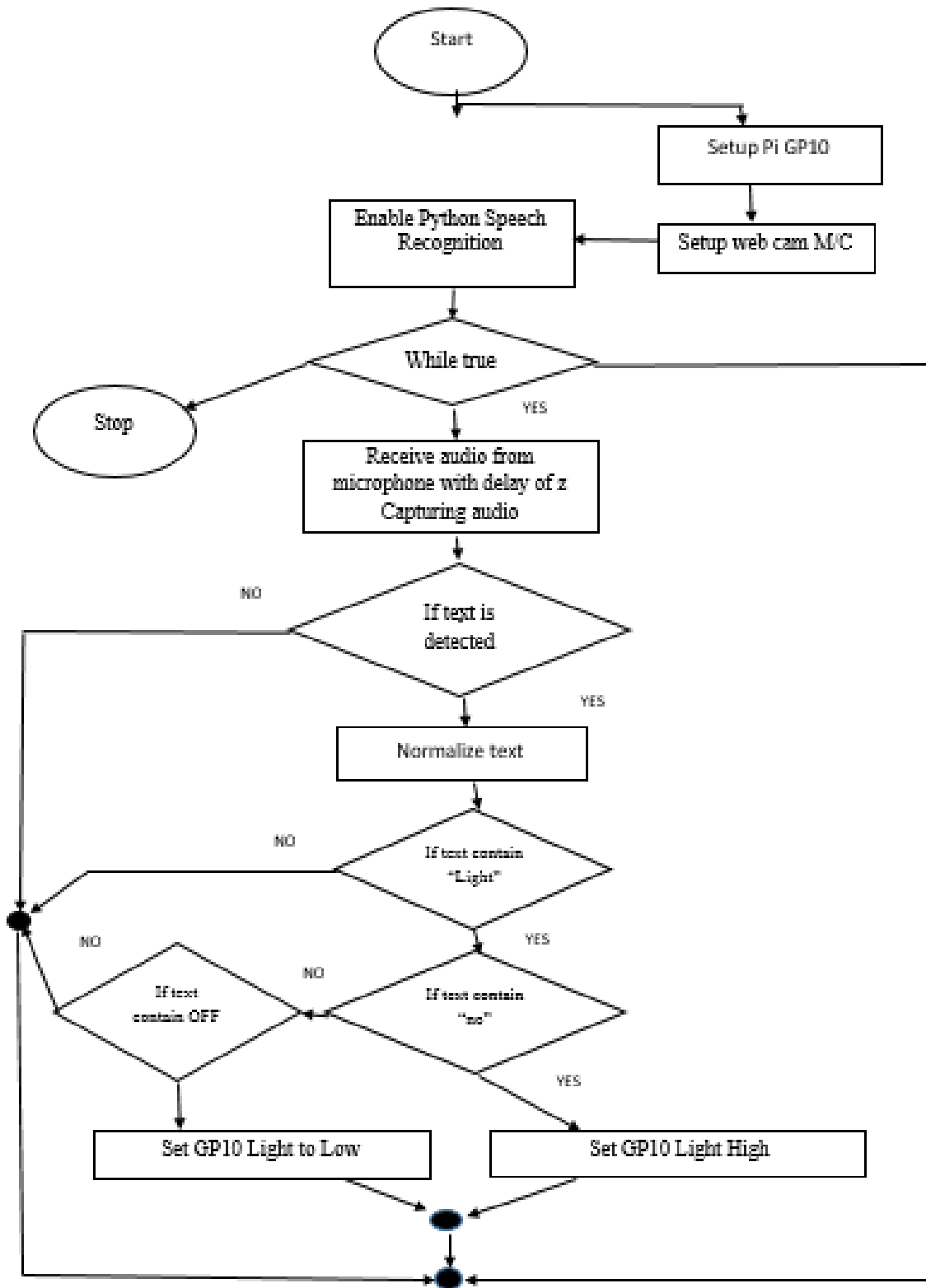
## APPENDIX

### Appendix for Flowchart

#### Tesseract OCR Flowchart



### Voice Command Flowchart:





## Appendix 2 Codes

### Capturing Image Code:

```
import cv2
import numpy as np

#Step 1: captures Image
cap = cv2.VideoCapture(0) #ignore the errors
cap.set(3, 960)    #Set the width important because the default will timeout
                  #ignore the error or false response
cap.set(4, 544)    #Set the height ignore the errors
r, frame = cap.read()
cv2.imwrite("webcam_image.jpg", frame)
```

#Step 2: Adjusts Brightness and Contrast of the Image using CLAHE

```
img = cv2.imread("webcam_image.jpg")
#-----Converting image to LAB Color model-----
lab= cv2.cvtColor(img, cv2.COLOR_BGR2LAB)
#cv2.imshow("lab",lab)

#-----Splitting the LAB image to different channels-----
l, a, b = cv2.split(lab)
#cv2.imshow('l_channel', l)
#cv2.imshow('a_channel', a)
#cv2.imshow('b_channel', b)

#-----Applying CLAHE to L-channel-----
clahe = cv2.createCLAHE(clipLimit=5.0, tileGridSize=(8,8))
cl = clahe.apply(l)
```

```
#cv2.imshow('CLAHE output', cl)

#-----Merge the CLAHE enhanced L-channel with the a and b channel-----
limg = cv2.merge((cl,a,b))
#cv2.imshow('limg', limg)

#-----Converting image from LAB Color model to RGB model-----
final = cv2.cvtColor(limg, cv2.COLOR_LAB2BGR)
cv2.imwrite("webcam_image_adjusted.jpg", final)

#Step 4: Denoise the grayscale image
img = cv2.imread("webcam_image_adjusted.jpg")

dst = cv2.fastNlMeansDenoisingColored(img,None,10,10,7,21)
cv2.imwrite('webcam_image_adjusted_denoised.jpg', dst)

#Step 3: Converts image to grayscale
image = cv2.imread("webcam_image_adjusted_denoised.jpg")

gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
cv2.imwrite("webcam_image_gray.jpg", gray_image)

#Step 4: Denoise the grayscale image
img = cv2.imread("webcam_image_gray.jpg")

dst = cv2.fastNlMeansDenoisingColored(img,None,10,10,7,21)
cv2.imwrite('webcam_image_gray_denoised.jpg', dst)
```

#Step 5: Converts grayscale denoised adjusted image to normalized Histogram Image

```
img = cv2.imread('webcam_image_gray_denoised.jpg',0)
hist_eq = cv2.equalizeHist(img)
#res = np.hstack((img,eq)) #stacking images side-by-side
cv2.imwrite('webcam_image_gray_normalized.jpg',hist_eq)
```

#Step 6: Converts normalized Histogram image to Binary Image

```
image= cv2.imread("webcam_image_gray_normalized.jpg",
cv2.CV_LOAD_IMAGE_GRAYSCALE)
thresh = 100
im_bw = cv2.threshold(image, thresh, 255, cv2.THRESH_BINARY)[1]
cv2.imwrite('webcam_image_bin.jpg', im_bw)
```

### **Voice Command Code:**

```
#!/usr/bin/env python3
import time
import speech_recognition as sr
import RPi.GPIO as GPIO #GPIO controls GPIO pins of the device
import os #Executes Command Line code
import sys
GPIO.setmode(GPIO.BOARD) ## Use board pin numbering
led_pin = 13
GPIO.setup(led_pin, GPIO.OUT) ## Setup GPIO Pin 7 to OUT
stop = False #Condition for Program to end
led_glow_delay_time = 0.1 #Delay time for each LED pulse
core_directory = ""
image_to_speech_url = core_directory + "image_to_sound.sh"
video_to_speech_url = core_directory + "video_to_image.sh"
image_url = core_directory + "letter.jpg"
```

#Runs code for Reading text from an image file which is saved in the device

```
defread_from_picture():
    command = "sudosh " + image_to_speech_url + " " + image_url
    os.system(command)
```

#Runs code for Capturing an image from webcam and reading it out

```
defread_from_video():
    os.system("sudosh " + video_to_speech_url)
def speak(text):
    command = "sudo -u pi espeak \"'\" + text + \"'\""
```

```
os.system(command)
```

#Blinks light n time

```
defblink_light(times):
```

```
if times > 0:
```

```
    i = 0
```

```
    while i < times:
```

```
        GPIO.output(led_pin, True)
```

```
        time.sleep(led_glow_delay_time)
```

```
        GPIO.output(led_pin, False)
```

```
        time.sleep(led_glow_delay_time)
```

```
        i = i + 1
```

#Process Command and perform task

```
defprocess_task(command):
```

```
    command=command.lower()
```

```
    if "light" in command:
```

```
        if "on" in command:
```

```
            speak("Turning on the light")
```

```
            GPIO.output(led_pin, True)
```

```
        elif "off" in command:
```

```
            speak("Turning off the light")
```

```
            GPIO.output(led_pin, False)
```

```
elif "google" in command:
    #blink_light(10)
print("You said:"+command)
elif "exit" in command:
speak("Exiting voice recognition")
    #GPIO.cleanup()
sys.exit(0)
    #stop = True
    #blink_light(5)
    #time.sleep(0.5)
    #blink_light(5)
print("You said:"+command)
elif "picture" in command:
    #blink_light(3)
    #time.sleep(0.3)
    #read_from_picture()
    #blink_light(3)
print("You said:"+command)
elif "video" in command:
    #blink_light(3)
    #read_from_video()
print("You said:"+command)
#Main Function that will run
if __name__ == "__main__":
    r = sr.Recognizer()
speak("Voice recognition activated")
while stop == False:
r.dynamic_energy_threshold = True
r.dynamic_energy_adjustment_ratio = 1.5
with sr.Microphone() as source:
print("Speak:")
```

```
r.adjust_for_ambient_noise(source, duration = 1)
audio = r.listen(source, timeout=None, phrase_time_limit=7)
```

```
try:
text = r.recognize_google(audio, language="en-GB")
print("You said " + text)
process_task(text)
except sr.UnknownValueError:
print("Could not understand audio")
except sr.RequestError as e:
print("Could not request results; {0}".format(e))
```

### **Raspberry Pi GPIO:**

```
import RPi.GPIO as GPIO
import time
import sys
import os
import subprocess
from subprocess import check_output
import signal
GPIO.setmode(GPIO.BCM)
specific_image_pin = 17
video_image_pin = 23
cancel_read_pin = 24
roll_paper_pin = 2
roll_paper_in = True
light_glow_pin = 3
glow_light = True
roll_out_paper_pin = 4
roll_paper_out = True
```

```

paper_in = False
proc_id=-1
reading_from = "empty"
GPIO.setup(specific_image_pin, GPIO.IN, pull_up_down=GPIO.PUD_UP) # GPIO 17 set up as
inputs, pulled up to avoid false detection.
GPIO.setup(video_image_pin, GPIO.IN, pull_up_down=GPIO.PUD_UP)
GPIO.setup(cancel_read_pin, GPIO.IN, pull_up_down=GPIO.PUD_UP)
GPIO.setup(roll_paper_pin, GPIO.OUT)
GPIO.setup(light_glow_pin, GPIO.OUT)
GPIO.setup(roll_out_paper_pin, GPIO.OUT)
defroll_in_paper(val):
print "Rolling paper in"
globalpaper_in
paper_in = True
cmd = os.system('sudo -u pi espeak "Rolling paper in"')
GPIO.output(roll_paper_pin, roll_paper_in)
GPIO.output(light_glow_pin, glow_light)
try:
time.sleep(10)
exceptKeyboardInterrupt:
cmd = os.system('sudo -u pi espeak Cancelling')
GPIO.output(roll_paper_pin, (not roll_paper_in))
defroll_out_paper(val):
print "Rolling paper out"
globalpaper_in
cmd = os.system('sudo -u pi espeak "Rolling paper out"')
GPIO.output(roll_paper_pin, (not roll_paper_in))
GPIO.output(light_glow_pin, (not glow_light))
GPIO.output(roll_out_paper_pin, roll_paper_out)
try:
time.sleep(10)

```

```

exceptKeyboardInterrupt:
cmd = os.system("sudo -u pi espeak Cancelling")
GPIO.output(roll_out_paper_pin, (not roll_paper_out))
paper_in = False
defremove_procs(name):
procs = map(int,check_output(["pidof",name]).split())
print "Total processes: " + str(len(procs))
    #proc_id = check_output(["pidof","espeak"])
iflen(procs) > 0 :
    #print "Killing PID:: " + str(proc_id)
    #os.system("ps -o cmd= {}".format(proc_id))
i=0
while(i<len(procs)):
print "Killing PID:: " + str(procs[i])
os.system("ps -o cmd= {}".format(procs[i]))
cmd = os.kill(int(procs[i]), signal.SIGTERM)
os.system("ps -o cmd= {}".format(procs[i]))
i+=1
    #os.system("ps -o cmd= {}".format(proc_id))
    #sys.exit(0)
else:
print "Processes=" + str(len(procs))
defcancel_read(channel):
print "Cancelling..."
globalpaper_in
ifpaper_in:
roll_out_paper(True)
remove_procs("espeak")
    #remove_procs("capture_image.py")
globalreading_from
ifreading_from == "video":

```



```

remove_procs("video_to_image.sh")
elifreading_from == "image":
remove_procs("image_to_sound.sh")
reading_from = "empty"
defread_specific_image(channel):
globalreading_from
reading_from = "image"
    #GPIO.output(roll_paper_pin, roll_paper_in)
file = "/home/pi/Desktop/character_recognition/core/letter.jpg"
print "Reading from the file " + file
    #cmd = os.system("sudo -u pi ./image_to_sound.sh "+file)
proc = subprocess.Popen(["sudo", "-u", "pi",
"/home/pi/Desktop/character_recognition/core/image_to_sound.sh", file])
    #proc.communicate()
    #print "PID: " + str(proc.pid)
    #global proc_id
    #proc_id=proc.pid
    #proc.wait()
reading_from = "empty"
defread_from_video(channel):
globalreading_from
reading_from = "video"
print "Reading from video"
    #cmd = os.system("sudo -u pi ./video_to_image.sh ")
globalpaper_in
if not paper_in:
roll_in_paper(True)
proc = subprocess.Popen(["sudo", "-u", "pi", "python",
"/home/pi/Desktop/character_recognition/core/capture_image.py"])
proc.wait()
    #cmd = os.system("sudo -u pi espeak Rolled")

```

```
roll_out_paper(True)
file = "/home/pi/Desktop/character_recognition/core/webcam_image.jpg"
print "Reading from the file " + file
    #cmd = os.system("sudo -u pi ./image_to_sound.sh "+file)
proc = subprocess.Popen(["sudo", "-u", "pi",
"/home/pi/Desktop/character_recognition/core/image_to_sound.sh", file])
    #proc.communicate()
    #print "PID: " + str(proc.pid)
    #global proc_id
    #proc_id=proc.pid
reading_from = "empty"
GPIO.add_event_detect(specific_image_pin, GPIO.FALLING, callback=read_specific_image,
bouncetime=300)
GPIO.add_event_detect(video_image_pin, GPIO.FALLING, callback=read_from_video,
bouncetime=300)
GPIO.add_event_detect(cancel_read_pin, GPIO.FALLING, callback=cancel_read,
bouncetime=300)
cmd = os.system("sudo -u pi espeak Hello")
while(True):
    try:
        time.sleep(3)
    except KeyboardInterrupt:
        print "Exiting"
        GPIO.cleanup()
        break
    sys.exit()
```

## BIBLIOGRAPHY

- [1] Anon, (2017). [online] Available at: [http://print.thefinancialexpress-bd.com/old/more.php?date=2012-01-06&news\\_id=92797](http://print.thefinancialexpress-bd.com/old/more.php?date=2012-01-06&news_id=92797) [Accessed 24 Dec. 2017].
- [2] Nabi, M. (2017). *Eye care in Bangladesh*. [online] The Daily Star. Available at: <http://www.thedailystar.net/news-detail-206301> [Accessed 24 Dec. 2017].
- [3] En.wikipedia.org. (2017). *Miniature camera*. [online] Available at: [https://en.wikipedia.org/wiki/Miniature\\_camera](https://en.wikipedia.org/wiki/Miniature_camera) [Accessed 24 Dec. 2017].
- [4] Techshopbd.com. (2017). *Home / Techshopbd*. [online] Available at: <https://www.techshopbd.com/> [Accessed 24 Dec. 2017].
- [5] GitHub. (2017). *tesseract-ocr*. [online] Available at: <https://github.com/tesseract-ocr> [Accessed 24 Dec. 2017].
- [6] GitHub. (2017). *haywood/Handwriting-Recognition-System*. [online] Available at: <https://github.com/haywood/Handwriting-Recognition-System> [Accessed 24 Dec. 2017].
- [7] Opensource.com. (2017). *Google's Optical Character Recognition (OCR) software works for 248+ languages*. [online] Available at: <https://opensource.com/life/15/9/open-source-extract-text-images> [Accessed 24 Dec. 2017].
- [8] Raspberry Pi. (2017). *Raspberry Pi 3 Model B - Raspberry Pi*. [online] Available at: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/> [Accessed 24 Dec. 2017].
- [9] Arduino.cc. (2017). *Arduino - Home*. [online] Available at: <https://www.arduino.cc/> [Accessed 24 Dec. 2017].