

VISION FOR VISUALLY IMPAIRED

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science Engineering

By

AARUSH SINGH (Reg No. 38110001)

SUJEET KUMAR (Reg No. 38110569)



DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNINIVERSITY)

**Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600 119**

APRIL 2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with "A" Grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600 119.

Phone: 044 - 2450 3150 / 51 / 52 / 54 / 55 Fax: 044 - 2450 2344

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the Bonafede work of **AARUSH SINGH (Reg No: 38110001)** and **SUJEET KUMAR (Reg No: 38110569)** who carried out the project entitled **"VISION FOR VISUALLY IMPAIRED"** under our supervision from **NOVEMBER 2021** to **APRIL 2022**.

INTERNAL GUIDE

Dr. T. SASIKALA, M.E., Ph.D., Prof. and Dean

HEAD OF THE DEPARTMENT

Dr. T. SASIKALA, M.E., Ph.D., Prof. and Dean

Submitted for Viva voice Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, AARUSH SINGH (Reg No: 38110001) hereby declare that the Project Report entitled as **“VISION FOR VISUALLY IMPAIRED”** done by me under the guidance of **Dr. T. SASIKALA, M.E., Ph.D., Prof. and Dean**, Department of Computing at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of **BACHELOR of ENGINEERING (B.E.)** degree in **COMPUTER SCIENCE ENGINEERING (CSE)**.

DATE: 10/05/2022

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. SASIKALA, M.E., Ph.D., Prof. and Dean**, School of Computing for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. T. SASIKALA, M.E., Ph.D., Prof. and Dean**, Department of Computing for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of **COMPUTER SCIENCE AND Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT:

Although artificial vision systems could potentially provide very useful input to assistive devices for blind people, such devices are rarely used outside of laboratory experiments. Many current systems attempt to reproduce the visual image via an alternative sensory modality (often auditory or somatosensory), but this dominant "scoreboard" approach, is often difficult to interpret for the user. Here, I propose to offload the recognition problem onto a separate image processing system that then provides the user with just the essential information about the People in their surroundings using Facial Recognition . Specifically, I am going to work on image processing algorithm not only robustly, precisely, but something which rapidly recognize the people around them. In addition, of Facial Recognition, I am also using Google- Text-To Speech/PYTTS which basically helps my Algorithm to provide a user friendly experience.

This is a helpful aid for visually impaired people and people with disabilities. Facial Recognition will generate an output string and converts them into audio speech using pre and post processing with gTTS (Google Text to Speech). Google is used as its platform for virtual assistant which can be used in day to day life activities like checking mails, weather-forecast, news etc., further using Google Assistant, and python language we implement a voice based home automation. The major objective of this project is to help visually impaired by using various fields of technology.

This is a useful guide for outwardly debilitated endlessly individuals with incapacities. Facial Recognition will produce a result string and converts them into sound discourse utilizing pre and post handling with gTTS (GoogleText to Speech). Google is utilized as its foundation for remote helper which can be utilized in everyday life exercises like checking sends, climate estimate, news and soon, further utilizing Google Assistant, and python language we carry out a voice based home mechanization. The significant target of this task is to help outwardly hindered byutilizing different areas of innovation.

CHAPTERS	CONTENTS
1	Introduction / Block Diagram
2	Literature Survey
3	Aim and Scope
4	System Architecture /Methodology
5	Simulation output, Links and Accuracy
6	Conclusion

Chapter 1:

INTRODUCTION:

As per the UNESCO and the World Health Organization (WHO), around 300 million individuals all over the planet are assessed to be outwardly debilitated, out of which 87% live in non-industrial nations. Due to outwardly impairedness individuals can't perceive individuals which makes the significant issue for outwardly disabled populace which are dealing with numerous issues, A significant issue for an outwardly debilitated or outwardly hindered individual (BVI) to associate with the world to share information . India has the largest share of the world's blind? The estimated size of the severely vision impaired Indians is 1 crore (10 million) persons. Added to this are another 1 crore persons who are Low Vision (moderately vision impaired who too are disabled to work and live like normal sighted people).Visually Impaired people face a lot of difficulties.

There have been developments on mobile phones and computers that assist a blind person by combining computer vision tools with other existing expedient products such as Optical Character Recognition (OCR) system. This can be diminished by executing a mix of flexible innovation for example Facial Recognition and Text to Speech Engine which is a great device for individuals carrying on with their everyday life.

This Prototypes works in two phase i.e pre- handling and post handling where extraction, division and acknowledgment happens where the last data is gotten which is then changed over into discourse or sound result by utilizing different discourse motor, for example, gTTS, E-Speak however gTTs is more exact than TTs. The preffered system assists blind people by capturing the text and then by reading it to them. Extracting the text present is enacted with OCR. It is a tactic for transformation of images of writings on a label, printed books etc.OCR replaces binary images with texts and also detects white spaces. It also parses the integrity of the recognized text.

BLOCK DIAGRAM :

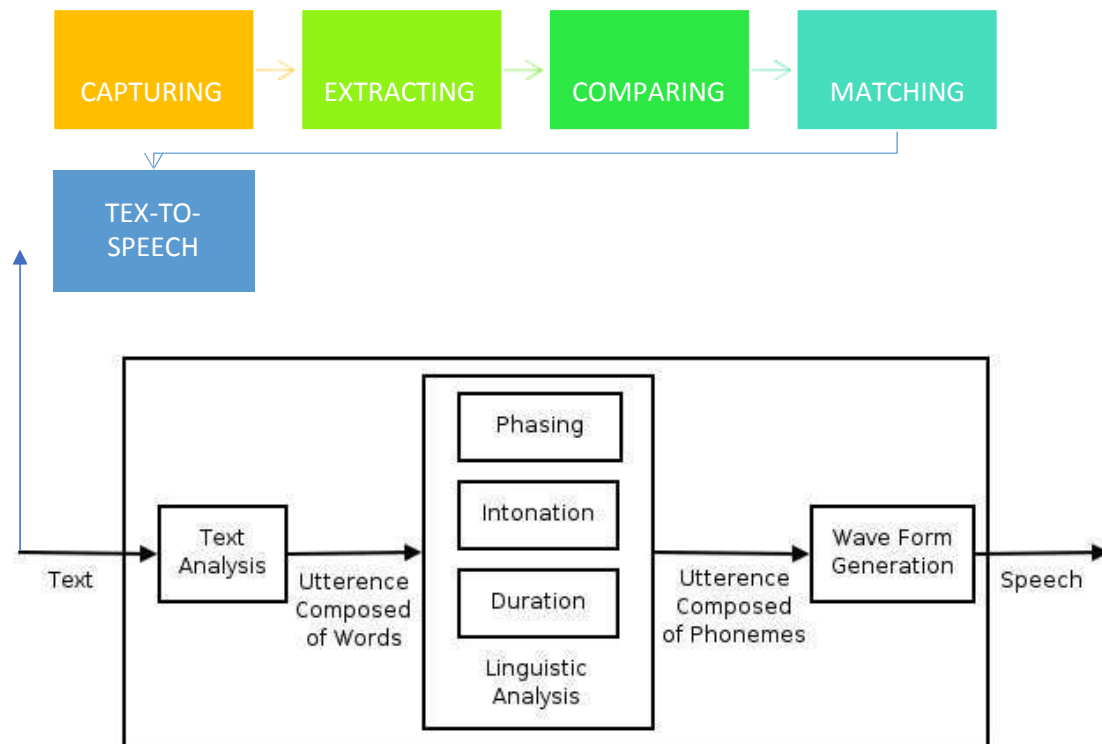


Figure 1 : Block Diagram

Chapter 2:

Literature Survey

TITLE	AUTHOR	DESCRIPTION	YEAR /JOURNAL	LIMITATIONS
Sign language recognition using deep learning	Pengkun Liu; Ruoxin Xiong	Human beings interact with each other either using a natural language channel such as words, writing, or by body language (gestures) e.g. hand gestures	2012/IEEE	Can't able to assist in Diseases detection at its earlier stage
CONVERSION OF SIGN LANGUAGE TO TEXT AND SPEECH USING MACHINE LEARNING TECHNIQUES	Hahm, O. Petersen, H.	The sign language is the basic communication method within hearing disable people. People with hearing disabilities face problems in communicating with other hearing people without a translator	2014/IEEE	Increases the time of the radiologist in evaluation. Hard to implement in Neural network systems.
Sign Language to Text and Speech Translation in Real Time Using Convolutional Neural Network	Mainetti, L. Vilei, A.	the implementation of a system that recognize the sign language would have a significant benefit impact on deaf people social live.	2011/IEEE	Timing response is not good. Take more the analysis

Chapter 3:

Aim:

- The main aim of this project is to overcome the image recognition problem onto a separate image processing system that then provides the user with just the essential information about the People in their surroundings using Facial Recognition.
- The Prototype deals with the aim of solving disability to see for visually impaired People as it recognizes faces of the people around them by using Image processing and Computer Vision algorithm which is nothing but a Success in Advancement of Facial Recognition

Scope:

- More distant in time, it very well may be utilized for other Indian dialects.
- It has a wide degree in coming posterity as far as man-made brainpower and controlling everything by a word.
- By utilizing Google right hand, we can make way following robots, driverless vehicle which will chip away at Google orders.
- It can likewise be utilized for making candy Giving framework.
- Very recurrent well-known applications is home automation are lighting control, outdoor yard water system controlling framework, controlling kitchen machines, and security reconnaissance systems.

Chapter 4:

SYSTEM ARCHITECTURE/METHODOLOGY :

The Prototype deals with the aim of solving disability to see for Blind People as it recognizes faces of the people around them by using Image processing and Computer Vision algorithm which is nothing but an Success in Advancement of Facial Recognition **without using pre-defined Facial Recognition library of Python.**

Step 1 : Creating A Database/ Getting Samples as Reference for Matching

- The initial step, utilizing Common Haarcascade XML File which assists me with Extracting the Features from the info picture by live picture catching camera.
- This utilizations AI procedures to get a serious level of precision based on the thing is designated "preparing information".
- This purposes "necessary picture" ideas to figure the "highlights" recognized.
- Now when creating the Database of the people for the visually impaired person by taking 100 pictures by converting RGB to GreyScale (To make the environment free from noise of various colors for extracting the Features with higher Accuracy) of a single person(within 5 to 6 seconds) to work my algorithm for >90% Accuracy. After Collecting the Database.
- Code for Creating Database

```

import cv2
import numpy as np
face_classifier = cv2.CascadeClassifier('C:/Users/Aarush
Singh/Desktop/VISION FOR BLIND/haarcascade_frontalface_default.xml')
def face_extractor(img):
    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    faces = face_classifier.detectMultiScale(gray,1.3,5)
    if faces is ():
        return None
    for(x,y,w,h) in faces:
        cropped_face = img[y:y+h, x:x+w]
    return cropped_face
cap = cv2.VideoCapture(0)
count = 0
while True:
    ret, frame = cap.read()
    if face_extractor(frame) is not None:
        count += 1
        face = cv2.resize(face_extractor(frame), (200, 200))
        face = cv2.cvtColor(face, cv2.COLOR_BGR2GRAY)
        file_name_path = 'C:/Users/Aarush Singh/Desktop/VISION FOR
BLIND/DATA/AARUSH' + str(count) + '.jpg'
        cv2.imwrite(file_name_path, face)
        cv2.putText(face, str(count), (50, 50),
cv2.FONT_HERSHEY_COMPLEX, 1, (0, 255, 0), 2)
        cv2.imshow('Face Cropper', face)
    else:
        print("Face not Found")
        pass
    if cv2.waitKey(1) == 13 or count == 100:
        break
cap.release()
cv2.destroyAllWindows()
print('Colleting Samples Complete!!!')

```

Step 2 : Training of Data(Making The Machine To Know the Purpose of Database)

More than one of the pictures are of individual A and the excess picture is of individual B. The calculation makes a facial implanting of each picture and afterward looks at them.

After the examination, the organization will be changed somewhat with the goal that embeddings of individual A are more like each other than they are to the inserting of individual B. Therefore, this helps the calculation to utilize facial estimations that permit it to precisely characterize pictures of a similar individual as being like one another. This cycle is then rehashed many thousands or even huge number of times. At last, the organization should then have the option to deliver exact facial embeddings for faces it has never identified.



```
import cv2
import
numpy as
np from
os
import
listdir
from os.path import isfile, join
data_path = 'C:/Users/Aarush Singh/Desktop/VISION FOR
BLIND/DATA/' onlyfiles = [f for f in listdir(data_path) if
isfile(join(data_path,f))] Training_Data, Labels = [], []
for i, files in
    enumerate(onlyfiles):
        image_path = data_path
        + onlyfiles[i]
        images = cv2.imread(image_path,
```

Step 3 : Setting Up Text-To-Speech Engine :

A text-to-speech system (or "engine") is composed of two parts:

- Front-end
- Back-end.

The front-end has two major tasks.

First, it converts raw text containing symbols like numbers and abbreviations into the equivalent of written-out words. This process is often called text normalization, pre-processing, or tokenization.

The front-end then assigns phonetic transcriptions to each word, and divides and marks the text into prosodic units, like phrases, clauses, and sentences. The process of assigning phonetic transcriptions to words is called text-to-phoneme or grapheme-to-phoneme conversion. Phonetic transcriptions and prosody information together make up the symbolic linguistic representation that is output by the front-end. The back-end—often referred to as the synthesizer—then converts the symbolic linguistic representation into sound.

T-T-S Engine Code :

```
import pyttsx3
def my_speak_cloud(my_message):
    engine = pyttsx3.init()
    rate = engine.getProperty('rate')
    engine.setProperty('rate', rate)
    engine.say('{}'.format(my_message))
    engine.runAndWait()
    #rate = engine.getProperty('rate')
```

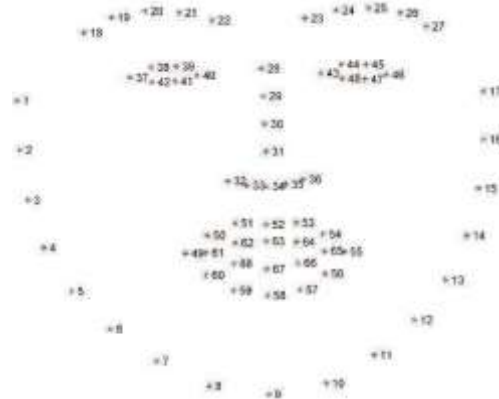
```
message= ''
```

```
A Known Person has been Recognized, This is MANMEET SINGH in front of
you. ''
```

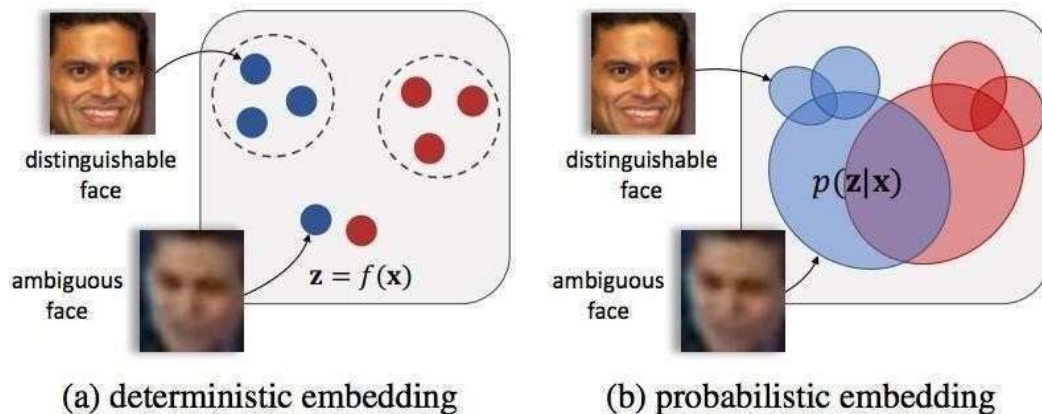
```
my_speak_cloud(message)
```

Step 4 : Implementing Live Image capturing and Matching with the Database and Conversion of String/Text into Audio Signals using Text-To-Speech Engine.

Before the algorithm can compare faces, we must convert the face images into data that the algorithm can understand. To do this, the system calculates measurements based on facial features and landmarks, also known as facial keypoints.using HAARCASCADE XML file.



Depending on the neural network, these landmarks may or not be used in creating the embedding. Sometimes the landmarks are used to crop the face image and remove noise in the background of the image.



The image above provides a good visualization of the difference between deterministic and probabilistic embeddings. On the left, the facial embedding in the latent space is represented as a point estimation with no indicators to the uncertainty of the embeddings. Whereas, with PFEs, the mean of the distribution represents the best estimation of the facial features in the image. The variance in the distribution represents the uncertainty.

Essentially, by creating face embeddings you are converting a face image into numerical data. That data is then represented as a vector in a latent semantic space. The closer the embeddings are to each other in the latent space, the more likely they are of the same person. However, the accuracy of models which use deterministic face embeddings depends on the clarity of the input images.

Furthermore, such models are often tested under constrained settings. In practice, input images (from surveillance video for example) are often taken in unconstrained or uncontrolled settings. The image quality may be low or portions of the face may be covered in the image. In such cases, approaches that use deterministic face embeddings suffer in performance.

Compiled Code :

```
import cv2
import numpy as np
from os import listdir
from os.path import isfile, join
data_path = 'C:/Users/Aarush Singh/Desktop/VISION FOR VISUALLY
IMPAIRED/DATA/'
onlyfiles = [f for f in listdir(data_path) if
isfile(join(data_path,f))]
Training_Data, Labels = [], []
import pyttsx3
def my_speak_cloud(my_message):
    engine = pyttsx3.init()
    rate = engine.getProperty('rate')
    engine.setProperty('rate', rate)
    engine.say('{}'.format(my_message))
    engine.runAndWait()
    #rate = engine.getProperty('rate')
message='''
A Known Person has been Recognized, This is AARUSH SINGH in front of you.
'''
for i, files in enumerate(onlyfiles):
    image_path = data_path + onlyfiles[i]
    images = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
    Training_Data.append(np.asarray(images, dtype=np.uint8))
    Labels.append(i)
Labels = np.asarray(Labels, dtype=np.int32)
model = cv2.face.LBPHFaceRecognizer_create()
##Local Binary Pattern (LBP) is a simple yet very efficient texture
operator which labels the pixels of an image by thresholding the
neighborhood of each pixel and considers the result as a binary number.
model.train(np.asarray(Training_Data), np.asarray(Labels))
print("Model Training Complete!!!!")
face_classifier = cv2.CascadeClassifier('C:/Users/Aarush
Singh/Desktop/VISION FOR BLIND/haarcascade_frontalface_default.xml')
def face_detector(img, size = 2):
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = face_classifier.detectMultiScale(gray,1.3,10)
    if faces is():
        return img,[]
    for(x,y,w,h) in faces:
        cv2.rectangle(img, (x,y), (x+w,y+h), (0,255,255),2)
        roi = img[y:y+h, x:x+w]
        roi = cv2.resize(roi, (200,200))
    return img,roi
```

Chapter 5:

SIMULATION OUTPUT:



Figure 3. Recognition of Known face.



Figure 4. Recognition of Unknown face.

Reading Text from the Image using Tesseract and Extracting captured text to notepad :

Pytesseract or **Python-tesseract** is an Optical Character Recognition (OCR) tool for python. It will read and recognize the text in images, license plates, etc. Here, we will use the tesseract package to read the text from the given image.

Mainly, 3 simple steps are involved here as shown below:-

- Loading an Image saved from the computer or download it using a browser and then loading the same. (Any Image with Text).
- Binarizing the Image (Converting Image to Binary).
- We will then Pass the Image through the OCR system.

```
import cv2

cam = cv2.VideoCapture(0)

cv2.namedWindow("test")

img_counter = 0

while True:
    ret, frame = cam.read()
    if not ret:
        print("failed to grab frame")
        break
```

```

cv2.imshow("test", frame)

k = cv2.waitKey(1)
if k%256 == 27:
    # ESC pressed
    print("Escape hit, closing...")
    break
elif k%256 == 32:
    # SPACE pressed
    img_name = "img.png".format(img_counter)
    cv2.imwrite(img_name, frame)
    print("{} written!".format(img_name))
    img_counter += 1

cam.release()
cv2.destroyAllWindows()

import cv2
import pytesseract
import numpy as np
from PIL import ImageGrab
import time

pytesseract.pytesseract.tesseract_cmd = 'C:\Program Files\Tesseract-OCR\tesseract.exe'
img = cv2.imread('img.png')
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
#print(pytesseract.image_to_string(img))

#Detecting Words
hImg, wImg, _ = img.shape
boxes = pytesseract.image_to_data(img)
print(boxes)
boxes = pytesseract.image_to_data(img)
for a,b in enumerate(boxes.splitlines()):
    #print(b)
    if a!=0:
        b = b.split()
        if len(b)==12:
            x,y,w,h = int(b[6]),int(b[7]),int(b[8]),int(b[9])
            cv2.putText(img,b[11],(x,y-
5),cv2.FONT_HERSHEY_SIMPLEX,1,(0,0,255),2)
            cv2.rectangle(img, (x,y), (x+w, y+h), (50, 50, 255), 1)

cv2.imshow("Result",img)
cv2.waitKey(0)

import pytesseract
from PIL import Image
pytesseract.pytesseract.tesseract_cmd = 'C:\Program Files\Tesseract-OCR\tesseract.exe'
img=Image.open('img.png')
text=pytesseract.image_to_string(img)
print(text)

f= open("iot.txt","w+")
f.write(text)
f.close()

```

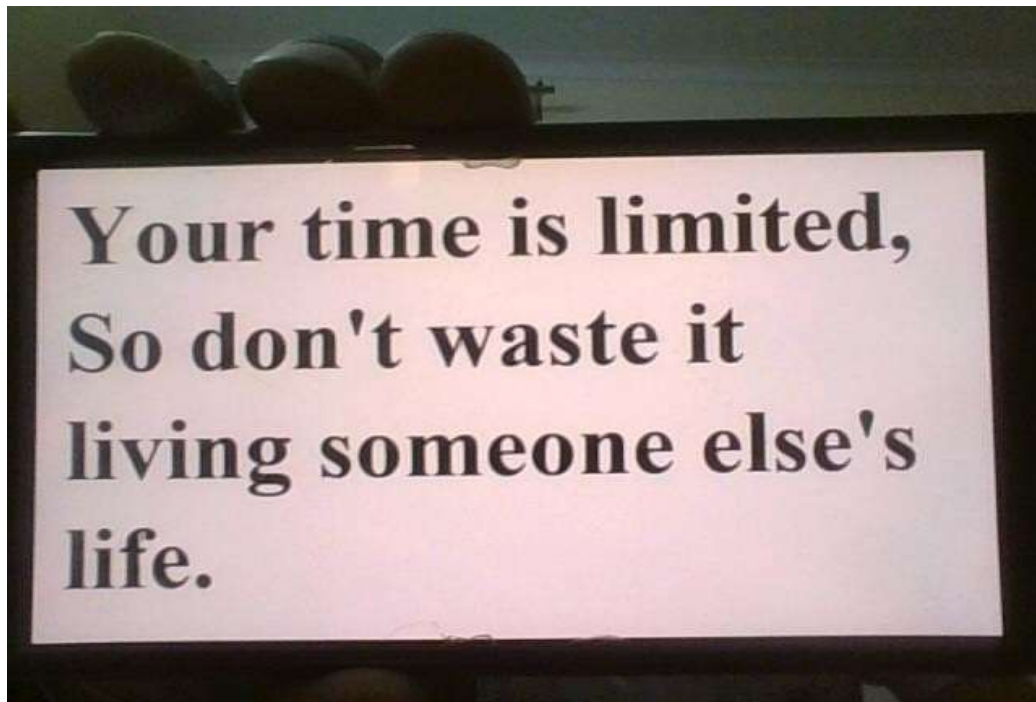
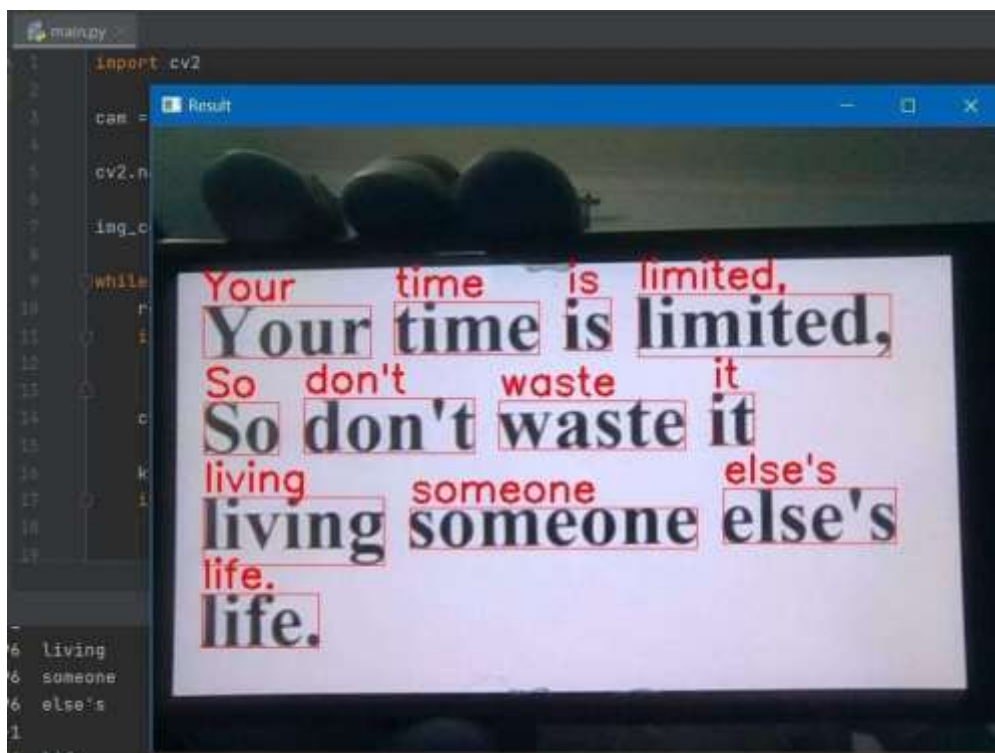


Figure 5. Capturing an image



level	page_num	block_num	par_num	line_num	word_num	left	top	width	height	conf	text
1	1	0	0	0	0	0	0	640	480	-1	
2	1	1	0	0	0	38	128	518	123	-1	
3	1	1	1	0	0	38	128	518	123	-1	
4	1	1	1	1	0	38	128	518	49	-1	
5	1	1	1	1	1	38	137	126	40	96	Your
5	1	1	1	1	2	182	134	109	40	95	time
5	1	1	1	1	3	312	131	33	40	96	is
5	1	1	1	1	4	366	128	199	48	96	limited,
4	1	1	1	2	0	39	204	414	47	-1	
5	1	1	1	2	1	39	211	56	40	96	So
5	1	1	1	2	2	114	208	128	42	95	don't
5	1	1	1	2	3	261	210	140	37	96	waste
5	1	1	1	2	4	422	204	31	40	96	it
2	1	2	0	0	0	37	277	523	122	-1	
3	1	2	1	0	0	37	277	523	122	-1	
4	1	2	1	1	0	38	277	522	59	-1	
5	1	2	1	1	1	38	284	136	52	96	living
5	1	2	1	1	2	194	292	216	31	96	someone
5	1	2	1	1	3	430	277	130	42	96	else's
4	1	2	1	2	0	37	358	88	41	-1	
5	1	2	1	2	1	37	358	88	41	96	life.
2	1	3	0	0	0	0	0	640	480	-1	
3	1	3	1	0	0	0	0	640	480	-1	
4	1	3	1	1	0	0	0	640	480	-1	
5	1	3	1	1	1	0	0	640	480	95	

Your time is limited,
So don't waste it

living someone else's
life.]

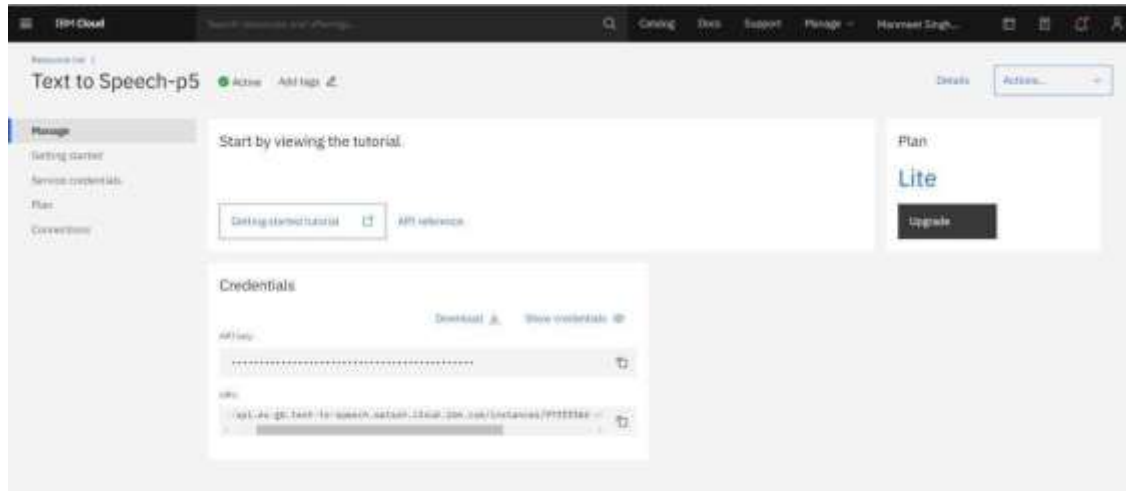
Figure 6. Image To Text conversion



Figure 7. Extracting captured text to notepad using pycharm

Step 5 : Conversion of Text into Audio Signal using IBM and Jupyter Notebook :

We used IBM text to speech and Jupyter Notebook to convert text file to mp3 file.



```
In [1]: # Documentation: https://cloud.ibm.com/docs/text-to-speech?topic=text-to-speech-voices
```

0. Install Dependencies

```
In [1]: !pip install ibm_watson
```

```
Requirement already satisfied: ibm_watson in c:\anaconda\lib\site-packages (5.1.0)
Requirement already satisfied: python-dateutil<=2.5.3 in c:\anaconda\lib\site-packages (from ibm_watson) (2.8.1)
Requirement already satisfied: websocket-client<=0.48.0 in c:\anaconda\lib\site-packages (from ibm_watson) (0.48.0)
Requirement already satisfied: requests<3.0,>=2.0 in c:\anaconda\lib\site-packages (from ibm_watson) (2.24.0)
Requirement already satisfied: ibm-cloud-sdk-core<=3.3.6 in c:\anaconda\lib\site-packages (from ibm_watson) (3.9.0)
Requirement already satisfied: six>=1.5 in c:\anaconda\lib\site-packages (from python-dateutil<=2.5.3->ibm_watson) (1.15.0)
Requirement already satisfied: urllib3<=1.25.0,!=1.25.1,!=1.26,>=1.21.1 in c:\anaconda\lib\site-packages (from requests<3.0,>=2.0->ibm_watson) (1.25.11)
Requirement already satisfied: certifi<=2017.4.17 in c:\anaconda\lib\site-packages (from requests<3.0,>=2.0->ibm_watson) (2020.6.20)
Requirement already satisfied: idna<3,>=2.5 in c:\anaconda\lib\site-packages (from requests<3.0,>=2.0->ibm_watson) (2.10)
Requirement already satisfied: chardet<4,>=3.0.2 in c:\anaconda\lib\site-packages (from requests<3.0,>=2.0->ibm_watson) (3.0.4)
Requirement already satisfied: PyJWT<3.0.0,>=2.0.1 in c:\anaconda\lib\site-packages (from ibm-cloud-sdk-core<=3.3.6->ibm_watson) (2.0.1)
```

```
In [2]: from ibm_watson import TextToSpeechV1
        from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
```

```
In [3]: apikey = 'EbeidnlyVigIHRV8shURA2nPORJswB-8nUCIqI129mIf'
        url = 'https://api.eu-gb.text-to-speech.watson.cloud.ibm.com/instances/97fff36b-d5de-dbed-931c-16d453aedfcd'
```

```
In [4]: # Setup Service
        authenticator = IAMAuthenticator(apikey)
        tts = TextToSpeechV1(authenticator=authenticator)
        tts.set_service_url(url)
```

2. Convert with A Basic Language Model

```
In [5]: with open('./speech.mp3', 'wb') as audio_file:
        res = tts.synthesize('Hello world!', accept='audio/mp3', voice='en-US_CharlotteV3Voice').get_result()
        audio_file.write(res.content)
```

3. Reading from a File

```
In [6]: with open('lot.txt', 'r') as f:  
        text = f.readlines()  
  
In [7]: text = [line.replace("\n", '') for line in text]  
  
In [8]: text = ''.join(str(line) for line in text)  
  
In [9]: with open('./lot.mp3', 'wb') as audio_file:  
        res = tts.synthesize(text, accept='audio/mp3', voice='en-GB-charlottev3voice').get_result()  
        audio_file.write(res.content)
```

Result Links:

English (United Kingdom) :

<https://1drv.ms/u/s!Ak9mCOCedA0UgOdvIJO7Hk7jZ62K?e=f4dJI3>

English (United States) :

<https://1drv.ms/u/s!Ak9mCOCedA0UgOdvIJO7Hk7jZ62K?e=f4dJI3>

French :

<https://1drv.ms/u/s!Ak9mCOCedA0UgOdvIJO7Hk7jZ62K?e=f4dJI3>

ACCURACY :

As Above Simulation Results shows the Working of the Prototype, Accuracy is:

- Known Faces : >=90%
- Unkown Faces: >=80% Depending on the Surrounding Noise.

Chapter 6:

CONCLUSION:

- We have put forward a multifunctional framework which go about as a guide for improving the nature of one's life. This task is made on different stages like python with its different bundles which is programed and run on Python IDLE as Virtual Prototype. By executing this framework, we are performing picture to text to discourse change, which is useful for Visually weakened People to make their life more straightforward.
- Whenever that individual comes before the outwardly weakened person (Wearing camera) It perceives the Face of that individual and Gives a string as Output, Following the acknowledgment happens that result string is then changed over into discourse or sound as result by utilizing different discourse motor, for example, gTTS, E-Speak however gTTs is more precise than TTs.(we are utilizing pytts) which straightforwardly goes to the ears of the Visually disabled Person Wearing Headphones + Camera.

REFERENCES:

- [1] Durette, B., Louveton, N. , Alleysson, D. , and H'erault, I., 2008. Visuo• auditory sensory substitution for mobility assistance: testing The VIBE. In Workshop on Computer Vision Applications for the Visually Impaired, Marseille, France.
- [2] Hernandez, A F. R. et al, 2009. Computer Solutions on Sensory Substitution for Sensory Disabled People. In Proceedings of the 8th WSEAS International Conference on Computational Intelligence, Man• machine Systems and Cybernetics, pp. 134-138.
- [3] Tang, H. , and Beebe, D. .I. , 2006. An oral tactile interface for blind navigation. IEEE Trans Neural Syst Rehabil Eng, pp. 116-123.
- [4] Auvray, M., Hanneton, S. , and O'Regan, J. K., 2007. Learning to perceive with a visuo - auditory substitution system: Localisation and object recognition with 'The vOICe'. Perception, pp. 416-430..
- [5] Kammoun, S. et al, 2012. Navigation and space perception assistance for

the visually impaired: The NAVIG project. In IRBM, Numro special ANR TECSAN, 33(2), pp. 182-189.

[6] Brian Katz, F. G. et al, 2012. NAVIG: Guidance system for the visually impaired using virtual augmented reality. In Technology and Disability, pp. 163-178.

[7] Abboud, S., Hanassy, S., Levy-Tzedek, S., Maidenbaum, S., and Amedi, A, 2014. EyeMusic: Introducing a "visual" colorful experience for the blind using auditory sensory substitution. In Restorative neurology and neuroscience.

[8] Martinez, B. D. C., Vergara-Villegas, O. O., Sanchez, v. G. c., De Jess Ochoa Domnguez, H., and Maynez, L. O., 2011. Visual Perception Substitution by the Auditory Sense. In Beniamino Murgante, Osvaldo Gervasi, Andrs Iglesias, David Taniar, and Bernady O. Apduhan, editors, ICCSA (2), volume 6783 of Lecture Notes in Computer Science, pp. 522-533. Springer.

[9] Jabnoun, H., Benzarti, F., and Amiri, H., 2014. Visual substitution system for blind people based on SIFT description. In International Conference of Soft Computing and Pattern Recognition. Tunisia, pp. 30.0.-30.5.

[10] Renier, L. et al, 2005. The Ponzo illusion with auditory substitution of vision in sighted and early-blind subjects. In Perception.

[11] Zhang, I., Ong, S. K. and Nee, A Y. c., 2009. Design and Development of a Navigation Assistance System for Visually Impaired Individuals. In

Proceedings of the 3rd International Convention on Rehabilitation Engineering & Assistive Technology, i-CREATE '09, New York, NY, USA.

[12] Jafri, R., Ali, SA, Arabnia, HR., and Fatima, S., 2013. Computer vision-based object recognition for the visually impaired in an indoors environment: a survey. In The Visual Computer, Springer Berlin Heidelberg, pp. 1-26.

[13] Lowe, GD., 2004. Distinctive Image Features from Scale-Invariant Keypoints. In Int. J. Comput. Vision, 60(2), pp. 91-110.

[14] Bay, H., Ess, A, Tuytelaars, T., and Gool, L.v. 2008. SURF: Speeded Up Robust Features. Computer Vision and Image Understanding (CVIU), pp. 346--359.

[15] S. Hare., I., Sina, S., and P. Dupplaw, D. 2011. OpenIMAJ and ImageTerrier: Java libraries and tools for scalable multimedia analysis and indexing of images. In Proceedings of the 19th ACM international conference on Multimedia (MM '11). ACM, New York, NY, USA, 691- 694. DOI=10.1145/2072298.2072421.

[16] Durette, B., Louveton, N., Alleysson, D., and H'erault, I, 2008. Visual-auditory sensory substitution for mobility assistance: testing The VIBE. In Workshop on Computer Vision Applications for the Visually Impaired, Marseille, France.

[17] Hernandez, A F. R. et al, 2009. Computer Solutions on Sensory Substitution for Sensory Disabled People. In Proceedings of the 8th WSEAS

International Conference on Computational Intelligence, Man• machine Systems and Cybernetics, pp. 134-138.

Tang, H. , and Beebe, D. .I. , 2006. An oral tactile interface for blind navigation. IEEE Trans Neural Syst Rehabil Eng,pp. 116-123.

[17] Ondrej, C.. and Matas, 1. 2005. Matching with PROSAC-progressive sample consensus. Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Vol. I. IEEE.