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 Recoginition 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 :

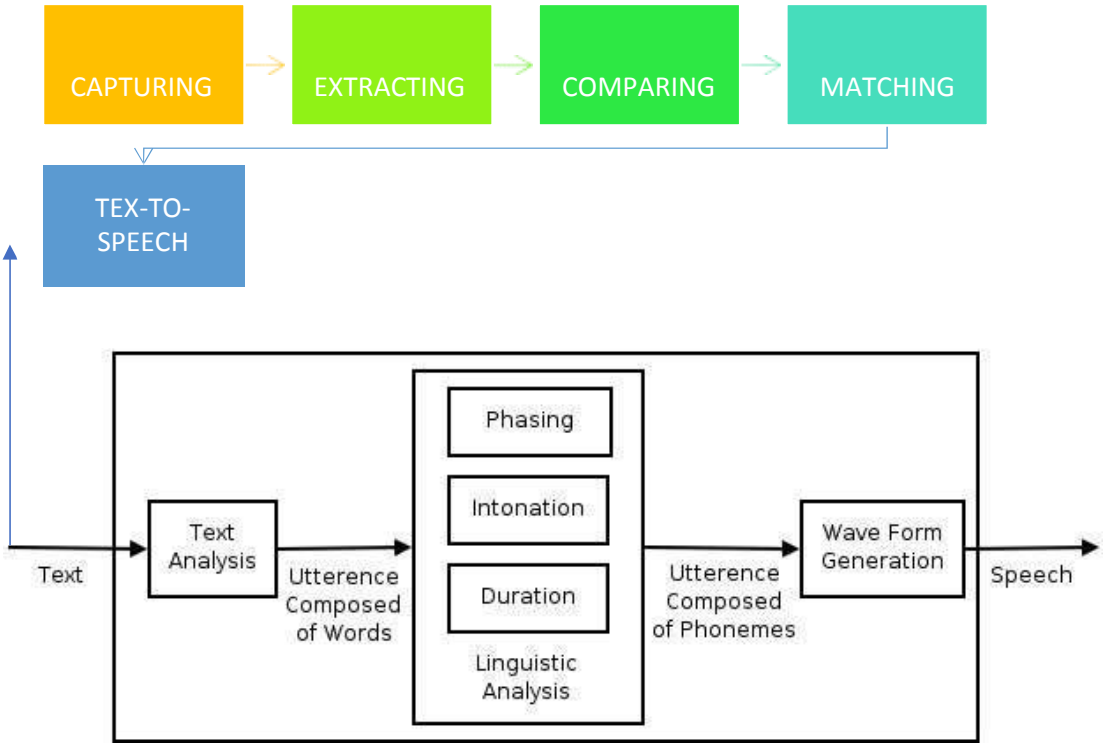


Figure 1 : Block Diagram

Chapter 2:

Literature Survey

TITLE	AUTHOR	DESCRIPTION	YEAR /JOURNAL	LIMTATIONS
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 recurrant 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.

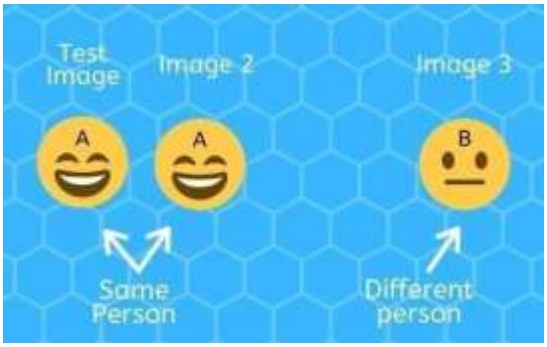
- Code for Creating Database

```
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 An 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.

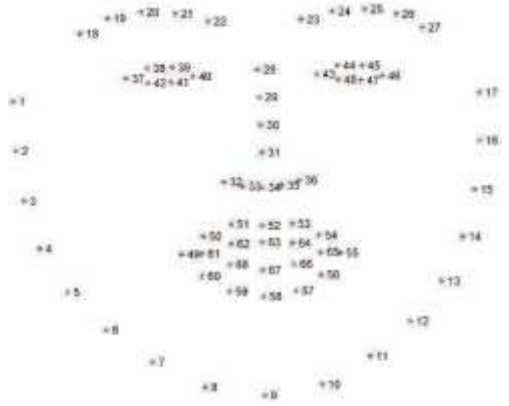
The front-end then assigns phonetic transcriptions to each word, and divides and marks the text into prosodic units, like phrases, clauses, and sentences. The process of assigning phonetic transcriptions to words is called text-to-phoneme or grapheme-to-phoneme conversion. Phonetic transcriptions and 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 :

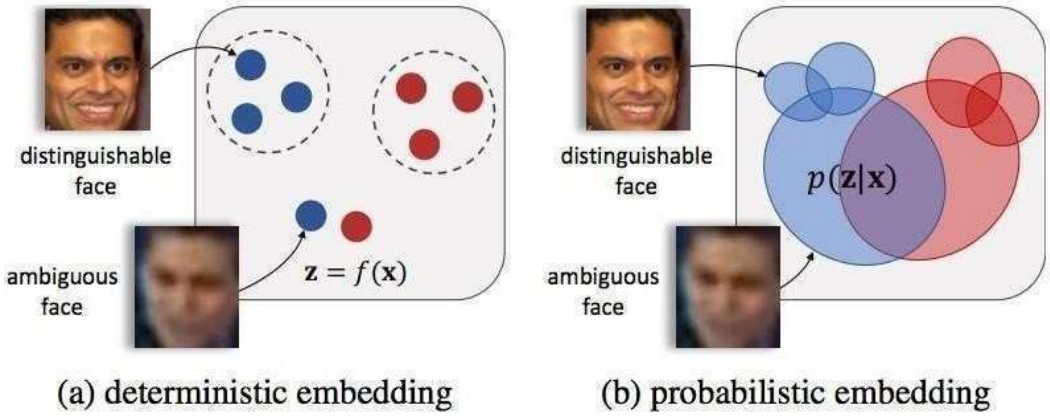
```
import pyttsx3
def my_speak_cloud(my_message):
    engine = pyttsx3.init()
    rate = engine.getProperty('rate')
    engine.setProperty('rate', rate)
    engine.say('{}'.format(my_message))
    engine.runAndWait()
    #rate = engine.getProperty('rate')
message=' ' '
A Known Person has been Recognized, 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.



(a) deterministic embedding (b) probabilistic embedding

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.