IMAGE RECOGNITION SPEECH EMBEDDED SYSTEM

A Project Work

Submitted in the partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE ENGINEERING SPECIALISATION IN BIG DATA ANALYTICS

Submitted by:

AMAN SHAH, UID: 17BCS3784

Under the Supervision of:

CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

CHANDIGARH UNIVERSITY, GHARUAN, MOHALI

140413, PUNJAB

NOVEMBER 2020

DECLARATION

I, **'Aman Shah 17BCS3784'** student of **'Bachelor of Engineering in Computer Science Engineering Specialization in Big Data Analytics'**, **session: 2020**, Apex Institute of Technology, Chandigarh University, Punjab, hereby declare that the work presented in this Project Work entitled '**Image Recognition Speech Embedded System'** is the outcome of our own bona fide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. It contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Aman Shah
Candidate UID: 17BCS3784

Date: 9th of Nov 2020
Place: Chandigarh University

CERTIFICATE

This is to certify that the work embodies in this dissertation entitled ***'Image Recognition Speech Embedded System'*** being submitted by **Aman Shah 17BCS3784** for partial fulfillment of the requirement for the award of **Bachelor of Engineering** in ***Computer Science Engineering Specialization in Big Data Engineering*** discipline to Apex Institute of Technology, Chandigarh University, Punjab during the academic year 2020 is a record of bonafide piece of work, undertaken by him/her the supervision of the undersigned.

Approved and Supervised by

Signature of Supervisor

Forwarded by

Professor & Head of Department

EXTERNAL EXAMINER

Signature of External Examiner
(External Examiner's Name)

Table of Contents

CHAPTER 1

1.INTRODUCTION

1.1 ABOUT IMAGE RECOGNITION

Image recognition, in the context of machine vision, is the ability of software to identify objects, places, people, writing and actions in images. Computers can use machine vision technologies in combination with a camera and artificial intelligence software to achieve image recognition.

1.2 IMAGE RECOGNITION USES

Image recognition is used to perform a large number of machine-based visual tasks, such as labeling the content of images with meta-tags, performing image content search and guiding autonomous robots, self-driving cars and accident avoidance systems.

1.2 COMPUTER IMAGE RECOGNITION

While human and animal brains recognize objects with ease, computers have difficulty with the task. Software for image recognition requires deep machine learning. Performance is best on convolutional neural net processors as the specific task otherwise requires massive amounts of power for its compute-intensive nature. Image recognition algorithms can function by use of comparative 3D models, appearances from different angles using edge detection or by components. Image recognition algorithms are often trained on millions of pre-labeled pictures with guided computer learning.

Current and future applications of image recognition include smart photo libraries, targeted advertising, the interactivity of media, accessibility for the visually impaired and enhanced research capabilities. Google, Facebook, Microsoft, Apple and Pinterest are among the many companies that are investing significant resources and research into image recognition and related applications. Privacy concerns over image recognition and similar technologies are controversial as these companies can pull a large volume of data from user photos uploaded to their social media platforms.

CHAPTER 2

2.1 LITERATURE REVIEW:

Human beings perform face recognition automatically every day and practically with no effort.

Although it sounds like a very simple task for us, it has proven to be a complex task for a computer, as it has many variables that can impair the accuracy of the methods, for example: illumination variation, low resolution, occlusion, amongst other.

In computer science, face recognition is basically the task of recognizing a person based on its facial image. It has become very popular in the last two decades, mainly because of the new methods developed and the high quality of the current videos/cameras.

Note that face recognition is different of face detection:

Face Detection: it has the objective of finding the faces (location and size) in an image and probably extract them to be used by the face recognition algorithm.

Face Recognition: with the facial images already extracted, cropped, resized and usually converted to grayscale, the face recognition algorithm is responsible for finding characteristics which best describe the image.

The face recognition systems can operate basically in two modes:

Verification or authentication of a facial image: it basically compares the input facial image with the facial image related to the user which is requiring the authentication. It is basically a 1x1 comparison.

Identification or facial recognition: it basically compares the input facial image with all facial images from a dataset with the aim to find the user that matches that face. It is basically a 1xN comparison.

There are different types of face recognition algorithms, for example:

Eigenfaces (1991)

Local Binary Patterns Histograms (LBPH) (1996)

Fisher faces (1997)

Scale Invariant Feature Transform (SIFT) (1999)

Speed Up Robust Features (SURF) (2006)

Each method has a different approach to extract the image information and perform the matching with the input image. However, the methods Eigenfaces and Fisherfaces have a similar approach as well as the SIFT and SURF methods.

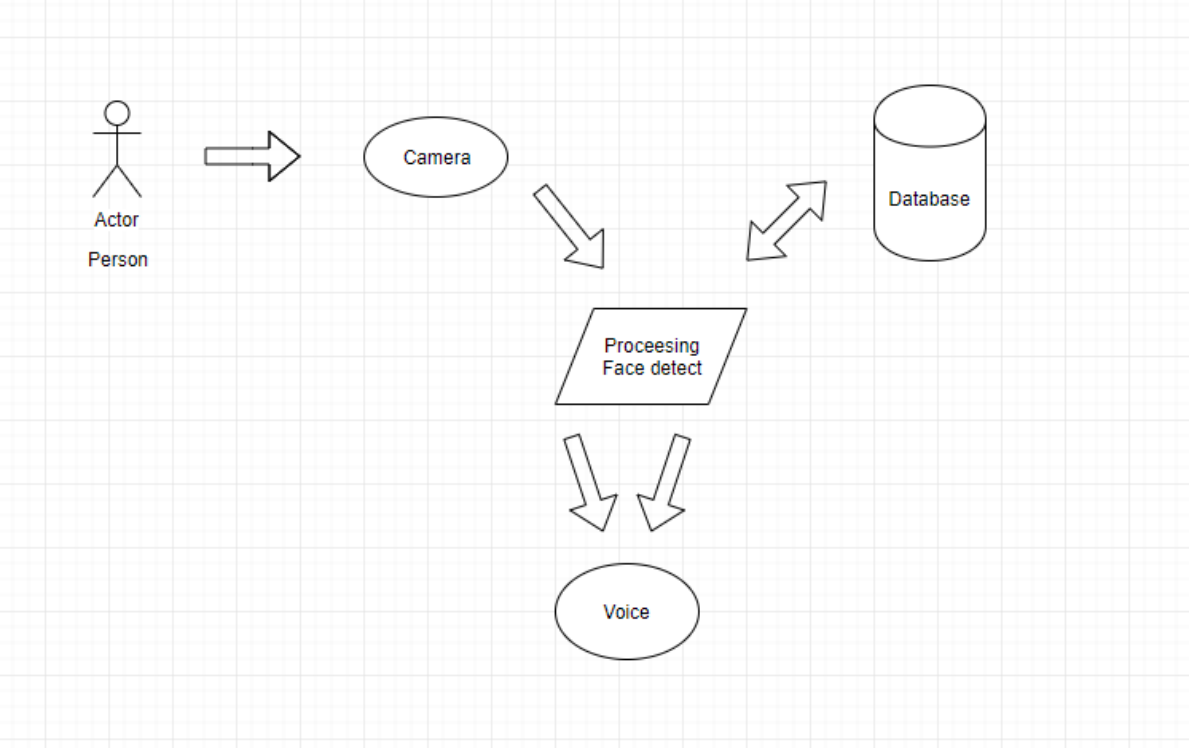
2.2 EXISTING FACIAL RECOGNITION PROJECTS:

A real time face recognition system is capable of identifying or verifying a person from a video frame. To recognize the face in a frame, first it needs to detect whether the face is present in the frame. If it is present, mark it as a region of interest (ROI), extract the ROI and process it for facial recognition.

Mainly this is done by using the Open Cv.

One of the most efficient algorithms developed for object detection was YOLO algorithm which is faster and more responsive but the problem is that it needs much more dedicated hardware to process the information. Mainly expensive GPU's are needed for parallel processing. But for this project we need something which is fast, reliable, responsive, cost efficient hardware as well as durable. So, we came up with this solution: -

2.3 PROPOSED SYSTEM DESIGN:



LIBRARIES USED:

2.4 ABOUT OPEN CV:

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 18 million. The library is used extensively in companies, research groups and by governmental bodies.

2.5 ABOUT SPEECH MODULES:

Text To Speech (TTS)

A computer system used to create artificial speech is called a speech synthesizer, and can be implemented in software or hardware products.

Pyttsx

Pyttsx is a cross-platform speech (Mac OSX, Windows, and Linux) library. You can set voice metadata such as age, gender, id, language and name. Thee speech engine comes with a large amount of voices.

2.6 PROBLEM STATEMENT

Blindness is one of the most, if not the most, misunderstood type of disability. The general masses have their notions about the blind people that they firmly believe to be true without even getting in touch with a blind person. Most of the members of the non-blind community believe that the blind people cannot do their work or live a normal life.

Blind people do lead a NORMAL LIFE with their own style of doing things. But they definitely face troubles due to inaccessible infrastructure and social challenges.

NAVIGATING AROUND PLACES

The biggest challenge for a blind person, especially the one with the complete loss of vision, is to navigate around places. Obviously, blind people roam easily around their house without any help because they know the position of everything in the house. People living with and visiting blind people must make sure not to move things around without informing or asking the blind person. Commercial places can be made easily accessible for the blinds with tactile tiles. But, unfortunately, this is not done in most of the places. This creates a big problem for blind people who might want to visit the place.

RECOGNIZING PEOPLE:

It is really difficult for a blind person to recognize people around them or any objects around them. So, with our approach we are trying to make use of the image recognition approach using computer vision to recognize objects around them and hence alerting them to their surroundings.

2.7 MAIN APPROACH:

Our main approach of the project is to recognizes faces around the surrounding of the particular person and alert the person by speaking out the name of the recognized face. This will not only help the blind person to become confident as machine don't lie and also with this, they would feel much confident and independent.

IMPLEMENTATION APPROACH:

- Automated the dataset creation by developing a python script.
- Using OpenCV developed a face recognition python script
- Interfaced the voice module with the face recognition algorithm.

2.8 HARDWARE SPECIFICATION:

- Dedicated graphics card.
- Image capturing device.
- Working system with not less than Intel i5 6th Gen processor.
- Ram minimum 8GB.
- Hard drive of at least 100GB storage.
- Speaker for speech output.

2.9 SOFTWARE SPECIFICATION:

- Python runnable environment
  1. Annaconda Navigator
     OR
  2. Python IDLE
     OR
  3. Spyder

- Some python dependencies:

1. OpenCv Library
2. Pyttsx2 or Pyttsx3
3. Numpy

CHAPTER:3

BACKGROUND AND PROPOSED METHOD

3.1 Technology used for the algorithm

Facial recognition is a way of recognizing a human face through technology. A facial recognition system uses biometrics to map facial features from a photograph or video. It compares the information with a database of known faces to find a match.

3.2 Working of Facial Recognition

You might be good at recognizing faces. You probably find it a cinch to identify the face of a family member, friend, or acquaintance. You're familiar with their facial features — their eyes, nose, mouth — and how they come together.

That's how a facial recognition system works, but on a grand, algorithmic scale. Where you see a face, recognition technology sees data. That data can be stored and accessed. For instance, half of all-American adults have their images stored in one or more facial-recognition databases that law enforcement agencies can search, according to a Georgetown University study.

So how does facial recognition work? Technologies vary, but here are the basic steps:

Step 1. A picture of your face is captured from a photo or video. Your face might appear alone or in a crowd. Your image may show you looking straight ahead or nearly in profile.

Step 2. Facial recognition software reads the geometry of your face. Key factors include the distance between your eyes and the distance from forehead to chin. The software identifies facial landmarks — one system identifies 68 of them — that are key to distinguishing your face. The result: your facial signature.

Step 3. Your facial signature — a mathematical formula — is compared to a database of known faces. And consider this: at least 117 million Americans have images of their faces in one or more police databases. According to a May 2018 report, the FBI has had access to 412 million facial images for searches.

Step 4. A determination is made. Your faceprint may match that of an image in a facial recognition system database.

CHAPTER 4:

METHODOLOGY

4.1 LIBRARY USED FOR IMAGE RECOGNITION:

OpenCV is a Computer Vision library with APIs that let your setup a pipeline for your Computer Vision project.

I/O. Loading data from image files, videos, capturing devices.

Performing feature extraction. OpenCV contains a long list of existing algorithms so you don't have to implement them yourself.

Applying machine learning algorithms for decision making, object recognition and detection How it works: You install the library on your computer. You start writing your code that will make use of the many features in OpenCV. You build your code and run it to perform the task you described.
In between writing your code and running your application you can assume some debugging and tweaking.

4.2 PICTORIAL REPRESENTATION OF THE ALGORITHM:

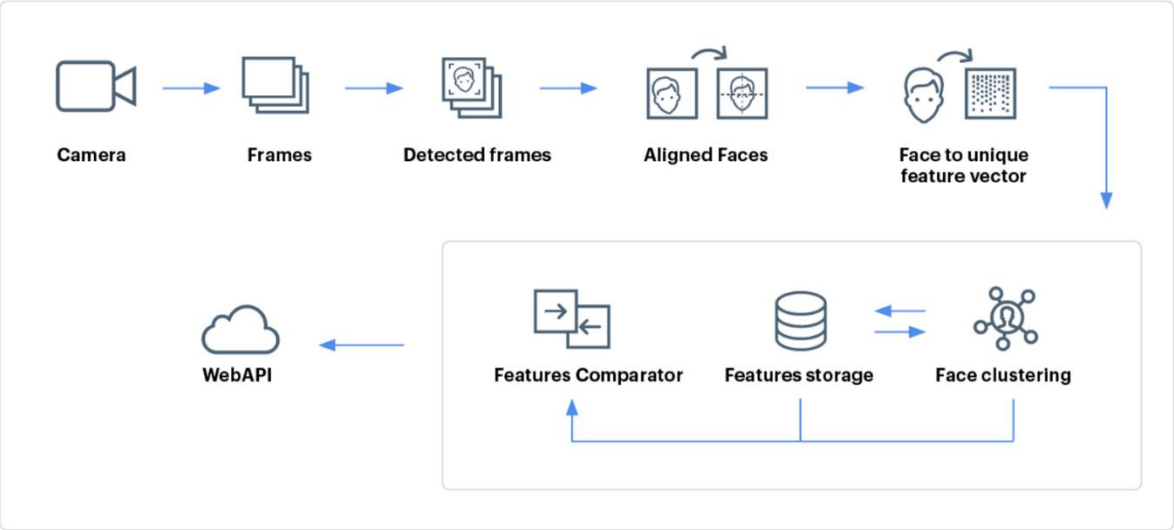


Fig 4.1: This Figure shows a brief of how image recognition is done.

Image source:
https://lh3.googleusercontent.com/rwfKXZCbIJAixo70ejDC_XEt4b26O4U8Pfg4mGSWACkSXHOr5GeajKz8iYcIwrUoPE8c=s170

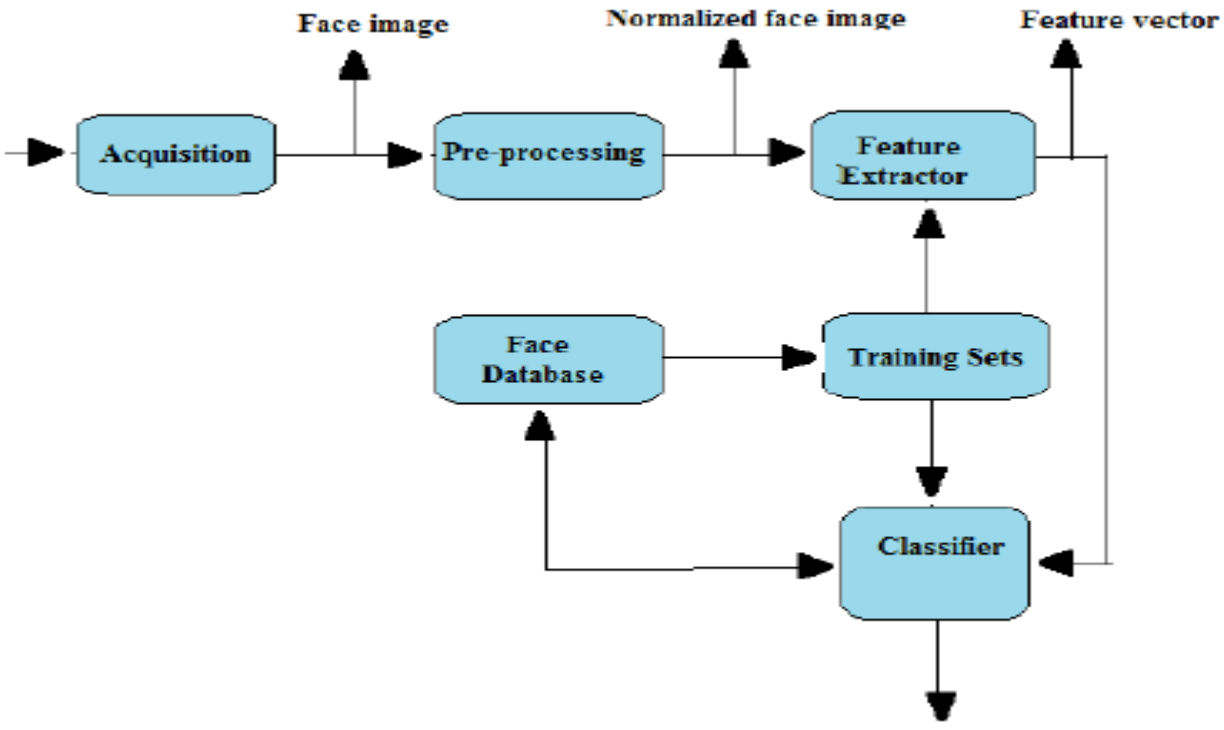


Fig 4.2: This Figure explains a brief of how image is extracted and classified

Image source:
https://www.researchgate.net/profile/Mahendra_Kumar45/publication/308789143/figure/fig1/AS:412931989753856@1475462066803/Block-diagram-of-the-face-recognition-system-15.png

4.3 DESIGNING OF RECOGNITION ALGORITHM:

Robust face recognition requires the ability to recognize identity despite many variations in appearance that the face can have in a scene. The face is a 3D object which is illuminated from a variety of light sources and surrounded by arbitrary background data (including other faces). Therefore, the appearance a face has when projected onto a 2D image can vary tremendously. If we wish to develop a system capable of performing non-contrived recognition, we need to find and recognize faces despite these variations. In fact, 3D pose, illumination and foreground-background segmentation have been pertinent issues in the field of computer vision as a whole.

Additionally, our detection and recognition scheme must also be capable of tolerating variations in the faces themselves. The human face is not a unique rigid object. There are billions of different faces and each of them can assume a variety of deformations. Inter-personal variations can be due to race, identity, or genetics while intra-personal variations can be due to deformations, expression, aging, facial hair, cosmetics and facial paraphernalia.

Furthermore, the output of the detection and recognition system has to be accurate. A recognition system has to associate an identity or name for each face it comes across by matching it to a large database of individuals. Simultaneously, the system must be robust to typical image-acquisition problems such as noise, video-camera distortion and image resolution.

Thus, we are dealing with a multi-dimensional detection and recognition problem. One final constraint is the need to maintain the usability of the system on contemporary computational devices ($\approx$ 100 MIPS). In other words, the processing involved should be efficient with respect to run-time and storage space.

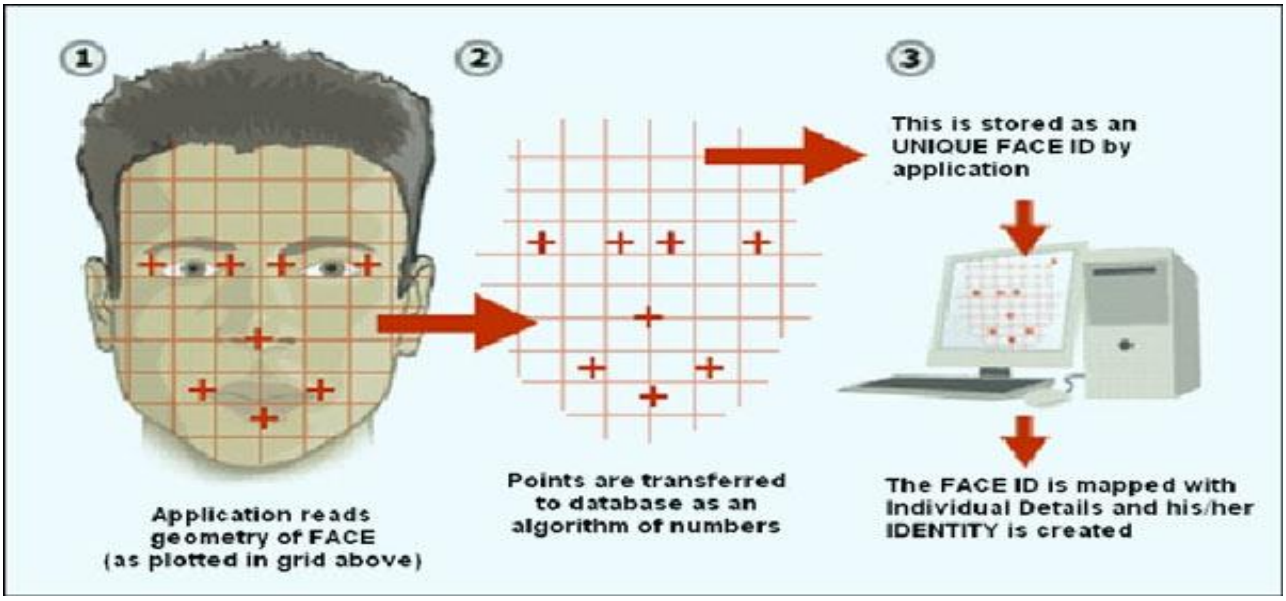


Fig 4.3: This Figure explains a brief of how image a face data is mapped and stored for unique representation

Image source: https://pbs.twimg.com/media/DJuIraPUQAUFnZe.jpg:large

CHAPTER 5:

EXPERIMENTAL SETUP:

5.1 STRUCTURE OF THE PROJECT

We are implementing a system that works on real time and analyses that information in real time.

5.2 Proposed system:

5.3 Dataset Creation:

We are developing a system which will take the live feed from a capturing device and start working on the images. For a facial recognition system to work. The first thing is we need to create a dataset. So, we have developed an automated python script that will create dataset by capturing images attained from the imaging device.

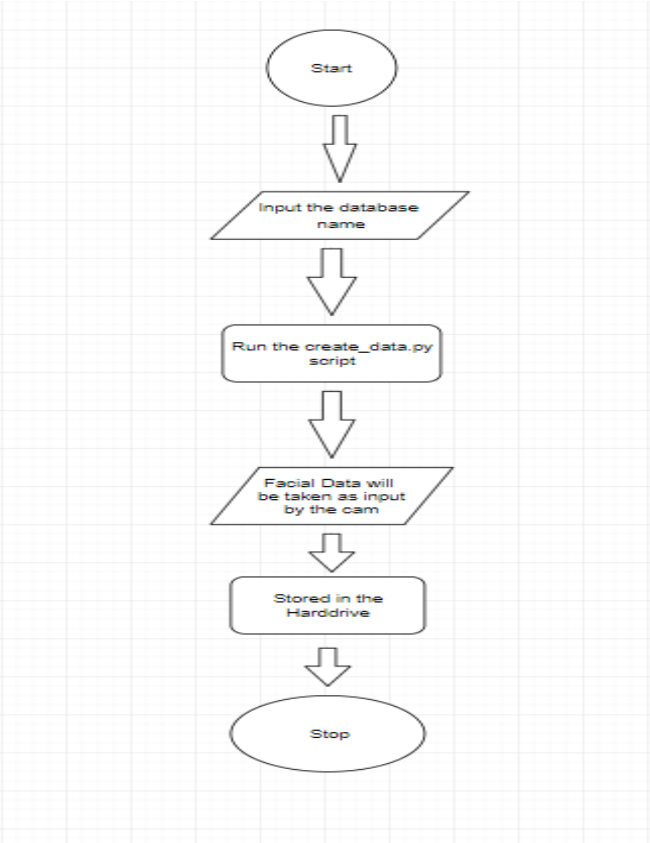


Fig 5.1: This Flowchart shows how image dataset is being generated and stored

5.4 The Recognition Part:

The recognition part works by taking input from the camera. Then we use the OpenCv library to develop the recognition algorithm. After a prediction is made with the highest probability from the dataset, we take the input from the recognizer and we deliver the input to the speech library to speak the name of the recognized face data. This happens in loop until and unless and exit command is pressed.

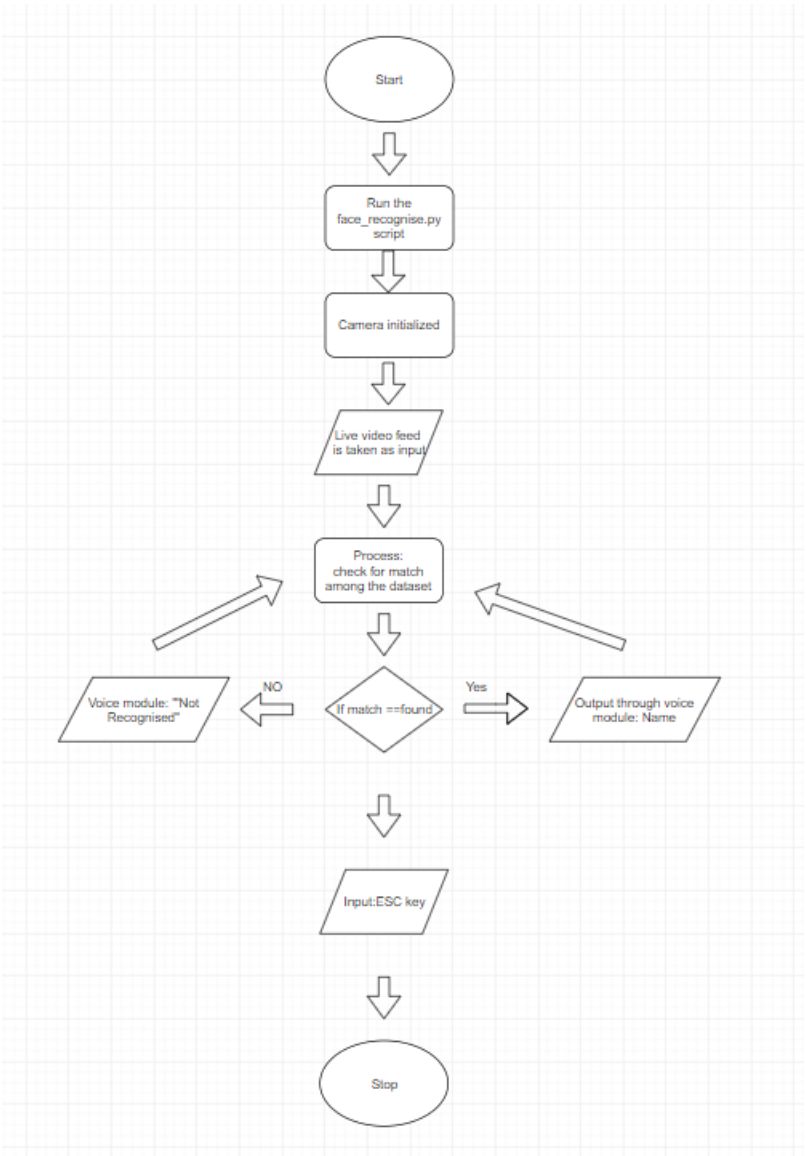


Fig 5.2: This Flowchart shows how recognition is done and interfaced with the speech module

5.5 BRIEF ABOUT WORKING OF OUR ALGORITHM:

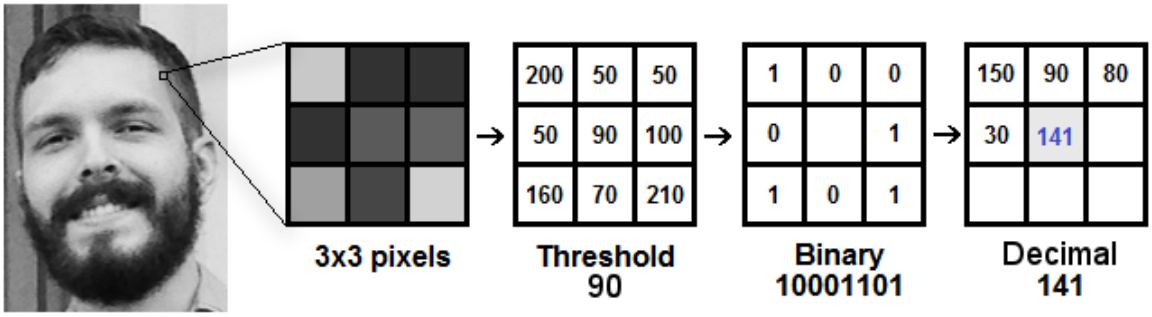


Fig 5.3: Image Data generation using Digital Image Processing

Image source: https://miro.medium.com/proxy/1*JOOavchVlRWOBbkAv0sSTA.png

Images are generally identified using array of pixel containing O's and 1's. Its the traditional way of doing it

It uses a single CNN network for both classification and localizing the object using bounding boxes. This is the architecture:

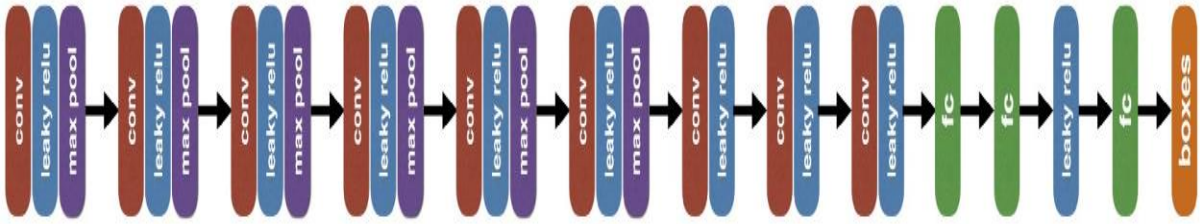


Fig 5.4: Classification and localization of the Image

Image source: medium.com

In the end, you will get a tensor value of 7*7*30.

For every grid cell, you will get two bounding boxes, which will make up for the starting 10 values of the 1*30 tensor. The remaining 20 denote the number of classes. The values denote the class score, which is the conditional probability of object belongs to class i, if an object is present in the box.

Next, we multiply all these class score with bounding box confidence and get class scores for different bounding boxes. We do this for all the grid cells. That is equal to 7*7*2 = 98.

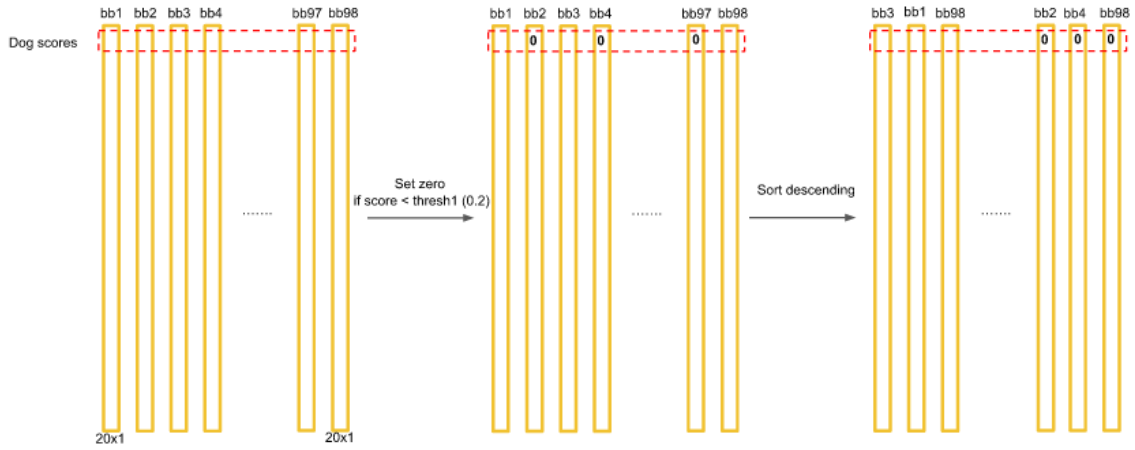


Fig 5.5: This figure shows how the most probable value is selected from the image data

Image source: https://lh3.googleusercontent.com/fXpI5R9iDnT5n-wnJVpW9hZ12QWz7WOV_wGSffmbAsLRKLBtdSmQnLLwhxYGKWX3j8Ggnw=s170

Now we will use Non-max suppression algorithm to set score to zero for redundant boxes.

Consider you have dog score for boundingbox1 as 0.5 and let this be the highest score and for box47 as 0.3. We will take an Intersection over Union of these values and if the value is greater than 0.5, we will set the value for box2 as zero, otherwise, we will continue to the next box. We do this for all boxes.

After all this has been done, we will be left with 2–3 boxes only. All others will be zero. Now, we select bbox to draw by class score value. This is explained in the image.

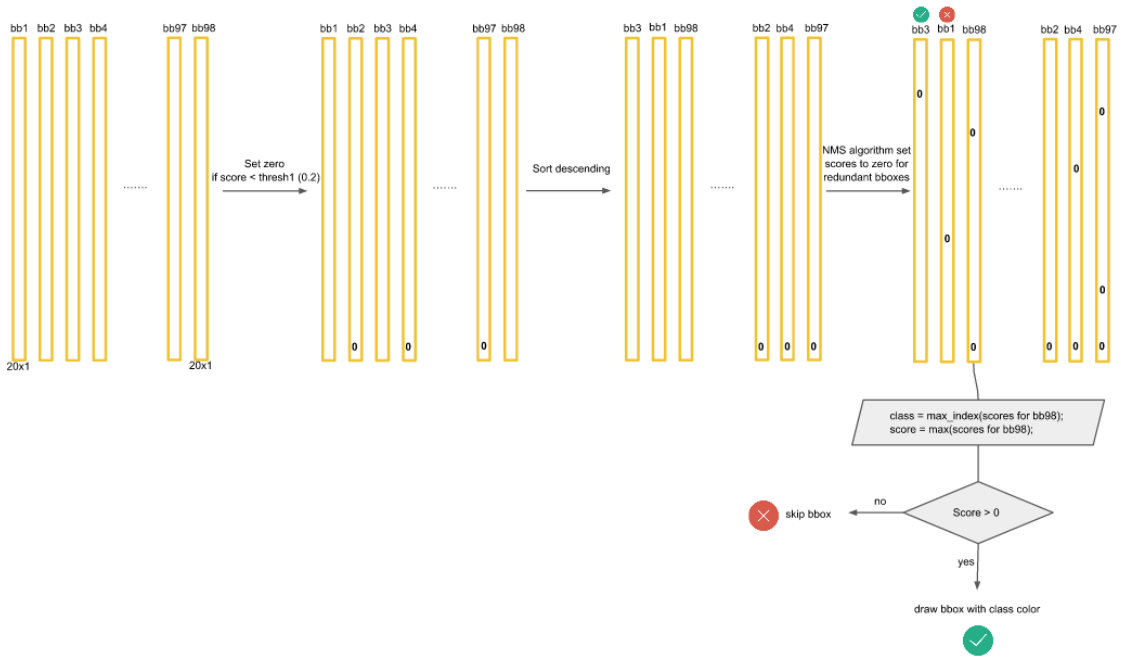


Fig 5.6: This figure shows how the most probable value is selected and image scoring is done

Image source: https://miro.medium.com/max/2222/1*bifa7gJ72omDAmYpazTS-A.png

Now we get to a conclusion by eliminating the least probable values and by arranging it in a decreasing series we get to a conclusion the most probable value by determining with the dataset taken as input.

This is how we get to a name from the recognition algorithm.

5.6 SPEECH INTERFACING WITH THE IMAGE RECOGNITION:

An application invokes the pyttsx3.init() factory function to get a reference to a pyttsx3.Engine instance. During construction, the engine initializes a pyttsx3.driver.DriverProxy object responsible for loading a speech engine driver implementation from the pyttsx3.drivers module. After construction, an application uses the engine object to register and unregister event callbacks; produce and stop speech; get and set speech engine properties; and start and stop event loops.

After the recognition algorithm we get an output from the prediction in a decreasing order. We take the highest output from it and input it into the **engine.say** part. This results in speaking up the name of the person being recognized.

CHAPTER 6

RESULT AND DISCUSSION:

With this project implementation we can make the blind people self-confident in navigating themselves in their day to day activities. It will also make a huge impact on the security of the physically challenged people as our algorithm can detect faces that comes near them and notify them using speech synthesis. Coming up for the solution of different key constrains this project can have a huge impact on the community.

SOME PICTURES OF THE PROJECT IMPLEMENTATION:

DATABASE CREATION:

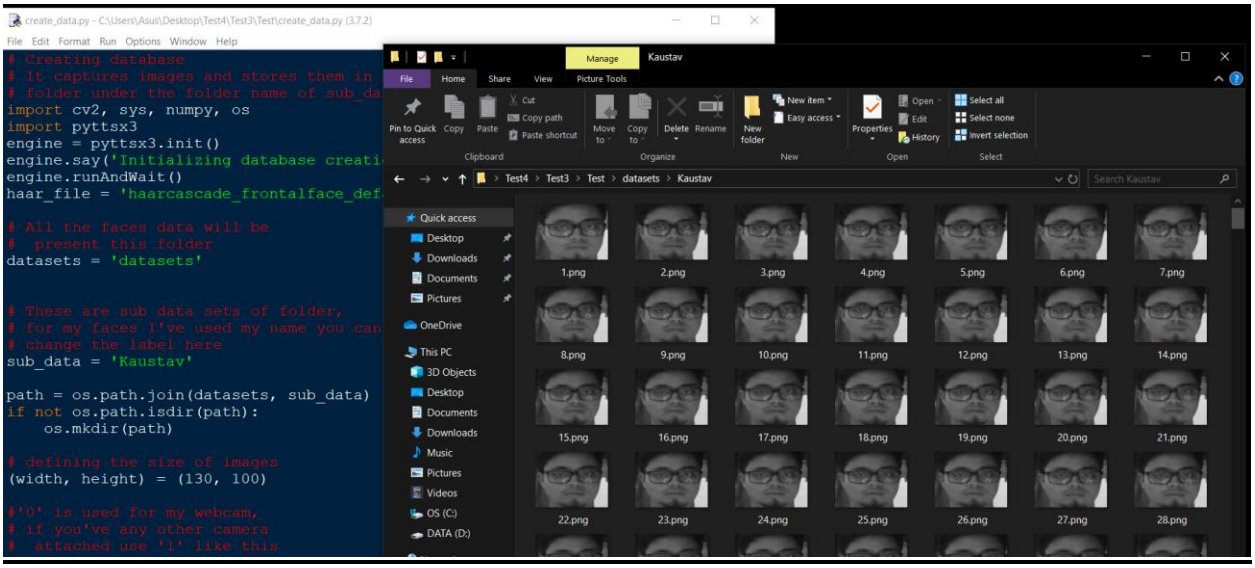


Fig 6.1: This figure shows the creation of database using create_data python script

RECOGNITION AND SPEECH IMPLEMENTATION PART:

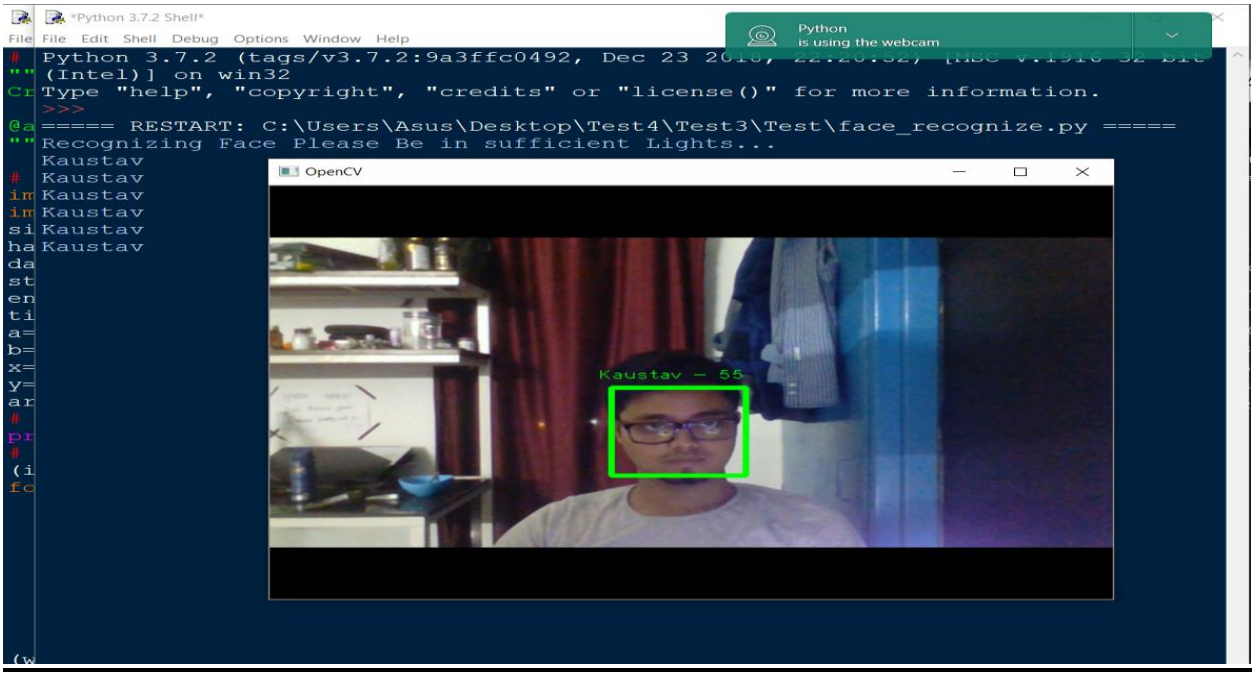


Fig 6.2: This figure shows the face recognition and speech part

CHAPTER 7

CONCLUSION AND FUTURE SCOPE:

This project can be a game changer in the market as blind people will get a huge benefit from it. Also, we can extend the project by taking it further to implementing in autonomous vehicles and many other yet to be researched on

This project can be made portable by using Raspberry pi and by encapsulating the whole hardware in a box to make it user friendly and reliable.

Raspberry Pi:

Fig 7.1: This is an image of raspberry pi chip

Image source: https://upload.wikimedia.org/wikipedia/commons/f/f1/Raspberry_Pi_4_Model_B_-_Side.jpg

Camera module:

Fig 7.2: This is an image of raspberry pi camera module

Image source: https://www.raspberrypi.org/homepage-9df4b/static/621b26de7977c5b8d765b3003b341a49/bc3a8/68fe7e4cb53767ad6c033bf3b46da3452188a24a_pi-camera-front-1-1426x1080.jpg

CHAPTER 8

REFERENCES

- Opencv.org
- Learning OpenCV: Computer Vision with the OpenCV Library By Gary Bradski, Adrian
- https://pythonspot.com/speech-engines-with-python-tutorial/
- https://searchenterpriseai.techtarget.com/definition/image-recognition
- https://www.tensorflow.org/
- https://www.osti.gov/biblio/6537037
- Google Scholar.
- Research Papers:
    1. Image Recognition Methods based on Deep Learning
       Publisher: IEE
       Author: Xia Jin

    2. Image Classification using Deep Learning
       Author: M Manoj Krishna
               M Neelima