SUMMER TRAINING REPORT

On

FACE RECOGNITION

Submitted to Guru Gobind Singh Indraprastha University, Delhi(India)

In partial fulfillment of the requirement for the award of the degree of

B.TECH

in

COMPUTER SCIENCE AND ENGINEERING

Submitted By

Vishal Jha

Roll no. 08215002716

DEPTT. OF COMPUTER SCIENCE

MAHARAJA SURAJMAL INSTITUTE Of TECHNOLOGY,

NEW DELHI-110058

OCTOBER 2018

ACKNOWLEDGEMENT

A research work owes its success from commencement to completion, to the people in love with researchers at various stages. Let me in this page express my gratitude to all those who helped us in various stage of this study. First, I would like to express my sincere gratitude indebtedness to **Ms.GEETIKA DHAND** (HOD, Department of Information Technology, Maharaja Surajmal Institute of Technology, New Delhi) for allowing me to undergo the summer training of 6 weeks at TCSion .

I am grateful to our guide **Mr. Aman Singh**, for the help provided in completion of the project, which was assigned to me. Without his friendly help and guidance it was difficult to develop this project.

I am also thankful to **Mr. Madan Mohan** for his true help, inspiration and for helping me to preparation of the final report and presentation.

I would not forget to remember **Ms. Lovely**, of Akatva for their encouragement and more over for their timely support and guidance till the completion of our project work.

Last but not least, I pay my sincere thanks and gratitude to all the Staff Members of TCSion for their support and for making our training valuable and fruitful.

Submitted By:

Vishal Jha

CERTIFICATE

TCS iON | Digital Learning Hub
Learn, Share, Collaborate

TATA CONSULTANCY SERVICES

This is to certify that

Vishal Jha

has successfully completed

College PRO - Technology Training with Projects : Python Programming

blended course offered by TCS iON

Grade Obtained: B

Course Duration: 60 **Hours** | **End Date:** 06/Aug/2018

Topics:

1) Technology Training - Python Programming 2) Mini Project

CERTIFIED

Cert. ID.: ICMP186853D5E1-330088

Dated: 06/Aug/2018

Head - Learning,
iON Digital Learning Hub

Grading:A+ >= 90%, A = 85% to 89.99%,A- = 80% to 84.99%, B+ = 75% to 79.99%,B = 70% to 74.99%, B- = 65% to 69.99%,C+ = 60% to 64.99%, C = 55% to 59.99%,C- = 50% to 54.99%,Fail < 50%

Akatva

CIN NO : U45402UP2016PTC085371
GSTIN No : 09AAECE3782C2Z0

TCS iON Training Partner

Mobile : +91-7599808080

Completion Certificate of Indust.Training

This is to certify that Mr. Vishal Jha has been undergoing 6 weeks training in Python with Machine Learning with us since 12th June 2018 and He has completed his training.

Akatva (TCS ION Training Partner) B-08, Sec-2, Noida Near Sec-15, Metro Station ,M:- 7599808080

Authorized Signatory

B-8 First Floor, Sec-2, Noida Near: Sec-15,Metro Station, Noida India (U.P.)
E-mail : Info@akatva.in I akatva.noida@gmail.com

Akatva

CIN NO : U45402UP2016PTC085371
GSTIN No : 09AAECE3782C2Z0

TCS iON Training Partner

Mobile : +91-7599808080

Project & Indust.Training Completion Letter

Date: 30th July 2018

Name- Vishal Jha
Maharaja Surajmal Institute Of Technology
Janakpuri, Delhi

Dear Mam,

This is to certify that **Vishal Jha** of **B.Tech(CSE)** of your prestigious **Maharaja Surajmal Institute Of Technology** has successfully completed the project based **Industrial Training** with us.

The Training was conducted from 12th June 2018 till 30th July 2018.

He has completed a project titled **Face Recognized Based Attendance System** on **Python With Machine Learning Technology**.

His performance was **Excellent** during the tenure. We wish him a bright and successful career.

With Warm Regards

Authorized Signatory

Akatva (TCS ION Training Partner) B-08, Sec-2, Noida Near Sec-15, Metro Station ,M:- 7599808080

CANDIDATE'S DECLARATION

I **Vishal Jha**, Roll No-08215002716, B.Tech (Semester- 5th) of the Maharaja Surajmal Institute of Technology, New Delhi hereby declare that the Training Report entitled **"FACE RECOGNITION"** is an original work and data provided in the study is authentic to the best of my knowledge.This report has not been submitted to any other Institute for the award of any other degree.

Vishal Jha

(Roll No. 08215002716**)**

Place: NEW DELHI

Date: 6th OCT 2018

ORGANIZATION INTRODUCTION

Akatva is a TCS iON Training Partner focused on providing high-end and high-quality trainings on cutting-edge technologies like Java, Python, Bigdata & Hadoop, MySQL, Microsoft.NET, PHP, Android and many more to the students who come to us.

TCS iON with its vast experience and expertise in digital and other technologies areas on a global scale provide the pedagogy and curricula of all the courses that are run in Training Centre. These trainings are delivered under the strict quality supervision of TCS iON by the "TCS iON Certified" faculty with the necessary technical infrastructure of par excellence for providing best trainings to the students.

The propellers of Akatva are educationists and academicians who nourish genuine desire in their hearts to uplift their students in every possible manner. They focus on employability of the students through quality trainings to make them stand on their feet in life. Akatva TCS iON is the outcome of this desire to provide best training in the IT and digital sphere. It is the best training hub existing today in Noida.

For helping its trained students, Akatva has a full-fledged Placement Division that takes care of all the training needs on the soft side of the candidates before they face interviews by the recruiters.

Vision

The vision is to act as a mediator between Technology and Students to improve their academic performance by providing non-syllabus inputs and Best Trainings on various Emerging Technologies.

ABSTRACT

The growing interest in computer vision of the past decade. Fueled by the steady doubling rate of computing power every 13 months, face detection and recognition has transcended from an esoteric to a popular area of research in computer vision and one of the better and successful applications of image analysis and algorithm based understanding. Because of the intrinsic nature of the problem, computer vision is not only a computer science area of research, but also the object of neuro-scientific and psychological studies, mainly because of the general opinion that advances in computer image processing and understanding research will provide insights into how our brain work and vice versa. Because of general curiosity and interest in the matter, I proposed to create an application that would allow user access to a particular machine based on an in-depth analysis of a person's facial features.

Identifying a person with an image has been popularized through the mass media. However, it is less robust to fingerprint or retina scanning. This report describes the face detection and recognition mini-project undertaken at Maharaja Surajmal Institute of Technology. It reports the technologies available in the Open-Computer-Vision (OpenCV) library and methodology to implement them using Python.

For face detection, Haar-Cascades were used and for face recognition Eigenfaces, Fisherfaces and Local binary pattern histograms were used. The methodology is described including flow charts for each stage of the system. Next, the results are shown including plots and screen-shots followed by a discussion of encountered challenges. The report is concluded with the authors' opinion on the project and possible applications.

ABBREVIATIONS AND ACRONYMS

OpenCV Open Source Computer Vision

LBPH Local Binary Pattern Histogram

LDA Linear Discriminant Analysis

PCA Principal Component Analysis

RGB Red Green Blue

Scrn Screen Shot

FIG Figure

TABLE OF CONTENTS

LIST OF TABLES

LIST OF SCREENSHOTS

LIST OF FIGURES

CONTENTS

Chapter-1

INTRODUCTION

1.1 About the Project

The following document is a report on the mini project for Face Recognition. It involved building a system for face detection and face recognition using several classifiers available in the open computer vision library(OpenCV). Face recognition is a noninvasive identification system and faster than other systems since multiple faces can be analyzed at the same time. The difference between face detection and identification is, face detection is to identify a face from an image and locate the face. Face recognition is making the decision ”whose face is it ? ”, using an image database. In this project both are accomplished using different techniques and are described below. The report begins with a brief history of face recognition. This is followed by the explanation of HAAR-cascades, Eigenface, Fisherface and Local binary pattern histogram (LBPH) algorithms. Next, the methodology and the results of the project are described. A discussion regarding the challenges and the resolutions are described. Finally, a conclusion is provided on the pros and cons of each algorithm and possible implementations

Chapter-2

HISTORY OF FACE RECOGNITION

2.1 History

Face recognition began as early as 1977 with the first automated system being introduced By Kanade using a feature vector of human faces [1]. In 1983, Sirovich and Kirby introduced the principal component analysis(PCA) for feature extraction [2]. Using PCA, Turk and Pentland Eigenface was developed in 1991 and is considered a major milestone in technology [3]. Local binary pattern analysis for texture recognition was introduced in 1994 and is improved upon for facial recognition later by incorporating Histograms(LBPH) [4], [5]. In 1996 Fisherface was developed using Linear discriminant analysis (LDA) for dimensional reduction and can identify faces in different illumination conditions, which was an issue in Eigenface method [6]. Viola and Jones introduced a face detection technique using HAAR cascades and ADABoost [7]. In 2007, A face recognition technique was developed by Naruniec and Skarbek using Gabor Jets that are similar to mammalian eyes [8], [9]. In This project, HAAR cascades are used for face detection and Eigenface, Fisherface and LBPH are used for face recognition.

Chapter-3

FACE DETECTION USING HAAR – CASCADES

3.1 Introduction to Haar-Cascades

A Haar wavelet is a mathematical fiction that produces square-shaped waves with a beginning and an end and used to create box shaped pattern to recognize signals with sudden transformations. An example is shown in figure 3.1. By combining several wavelets , a cascade can be created that can identify edges, lines and circles with different color intensities. These sets are used in Viola Jones face detection technique in 2001 and since then more patterns are introduced [10] for object detection as shown in figure 3.1. To analyse an image using Haar cascades, a scale is selected smaller than the target image. It is then placed on the image, and the average of the values of pixels in each section is taken. If the difference between two values pass a given threshold, it is considered a match. Face detection on a human face is performed by matching a combination of different Haar-like-features. For example, forehead, eyebrows and eyes contrast as well as the nose with eyes as shown below in figure A single classifier is not accurate enough. Several classifiers are combined as to provide an accurate face detection system as shown in the block diagram below in figure 3.3.

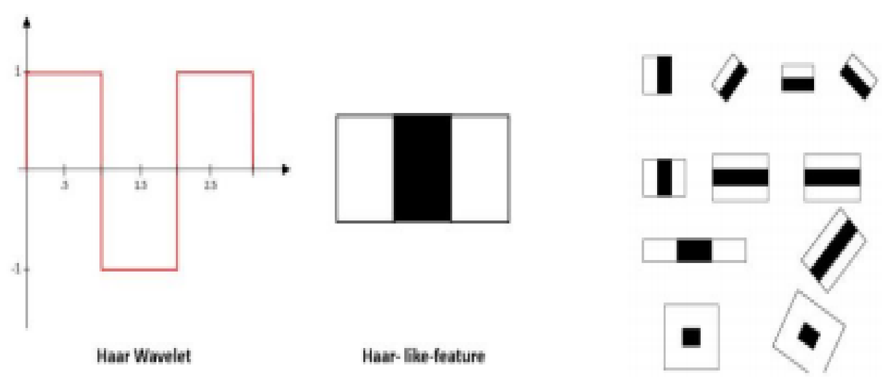


Fig 3.1 A Haar wavelet and resulting Haar-like features

In this project, a similar method is used effectively to by identifying faces and eyes in combination resulting better face detection. Similarly, in viola Jones method [7], several classifies were combined to create stronger classifiers. ADA boost is a machine learning algorithm that tests out several week classifiers on a selected location and choose the most suitable [7]. It can also reverse the direction of the classifier and get better results if necessary [7]. 2. Furthermore, Weight-update-steps can be updated.

Fig. 3.2 Several Haar-like-features matched to the features of authors face.

only on misses to get better performance. The cascade is scaled by 1.25 and re iterated in order to find different sized faces. Running the cascade on an image using conventional loops takes a large amount of computing power and time. Viola Jones [7] used a summed area table (an integral image) to compute the matches fast. First developed in 1984 [11], it became popular after 2001 when Viola Jones implemented Haar-cascades for face detection. Using an integral image enables matching features with a single pass over the image.

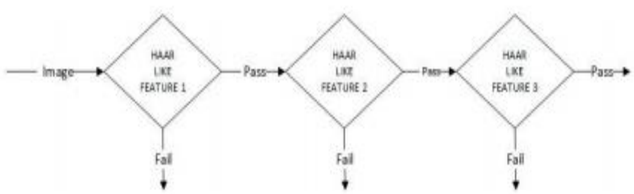


Fig 3.3 Haar-cascade flow chart

Chapter-4

FACE RECOGNITION

4.1 Brief

The following sections describe the face recognition algorithms Eigenface, Fisherface, Local binary pattern histogram and how they are implemented in OpenCV.

4.2 Eigenface

Eigenface is based on PCA that classify images to extract features using a set of images. It is important that the images are in the same lighting condition and the eyes match in each image. Also, images used in this method must contain the same number of pixels and in grayscale. For this example, consider an image with n x n pixels as shown in figure 4. Each raw is concatenated to create a vector, resulting a 1×n 2 matrix. All the images in the dataset are stored in a single matrix resulting a matrix with columns corresponding the number of images. The matrix is averaged (normalised) to get an average human face. By subtracting the average face from each image vector unique features to each face are computed. In the resulting matrix, each column is a representation of the difference each face has to the average human face. A simplified illustration can be seen in figure 4.1

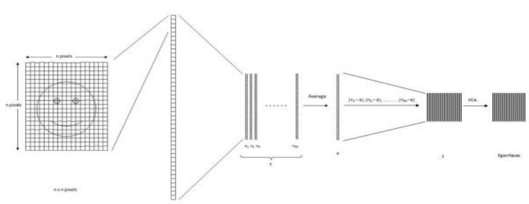


Fig 4.1 Pixels of the image are reordered to perform calculations for Eigenface

The next step is computing the covariance matrix from the result. To obtain the Eigen vectors from the data, Eigen analysis is performed using principal component analysis. From the result, where covariance matrix is diagonal, where it has the highest variance is considered the 1st Eigen vector. 2nd Eigen vector is the direction of the next highest variance, and it is in 90 degrees to the 1st vector. 3rd will be the next highest variation, and so on. Each column is considered an image and visualized, resembles a face and called Eigenfaces. When a face is required to be recognized, the image is imported, resized to match the same dimensions of the test data as mentioned above. By projecting extracted features on to each of the Eigenfaces, weights can be calculated. These weights correspond to the similarity of the features extracted from the different image sets in the dataset to the features extracted from the input image. The input image can be identified as a face by comparing with the whole dataset. By comparing with each subset, the image can be identified as to which person it belongs to. By applying a threshold detection and identification can be controlled to eliminate false detection and recognition. PCA is sensitive to large numbers and assumes that the subspace is linear. If the same face is analyzed under different lighting conditions, it will mix the values when distribution is calculated and cannot be effectively classified. This makes to different lighting conditions poses a problem in matching the features as they can change dramatically.

4.3 Fisherface

Fisherface technique builds upon the Eigenface and is based on LDA derived from Ronald Fishers' linear discriminant technique used for pattern recognition. However, it uses labels for classes as well as data point information [6]. When reducing dimensions, PCA looks at the greatest variance, while LDA, using labels, looks at an interesting dimension such that, when you project to that dimension you maximize the difference between the mean of the classes normalized by their variance [6]. LDA maximizes the ratio of the between-class scatter and within-class scatter matrices. Due to this, different lighting conditions in images has a limited effect on the classification process using LDA technique. Eigenface maximizes the variations while Fisherface maximizes the mean distance between and different classes and minimizes variation within classes. This enables LDA to differentiate between feature classes better than PCA and can be observed in figure 4.2 [12]. Furthermore, it takes less amount of space and is the fastest algorithm in this project. Because of these PCA is more suitable for representation of a set of data while LDA is suitable for classification.

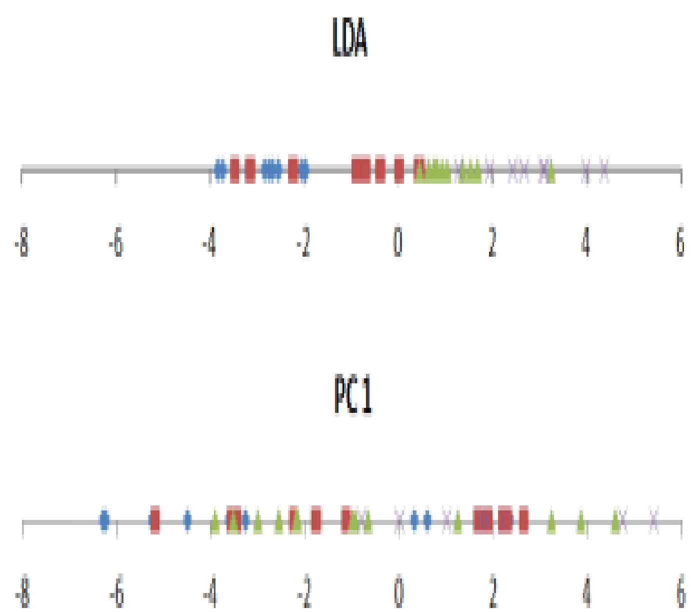


Fig 4.2 The first component of PCA and LDA. Classes in PCA looks more mixed than of LDA

4.3 Local Binary Pattern Histogram

Local binary patterns were proposed as classifiers in computer vision and in 1990 By Li Wang [4]. The combination of LBP with histogram oriented gradients was introduced in 2009 that increased its performance in certain datasets [5]. For feature encoding, the image is divided into cells (4 x 4 pixels). Using a clockwise or counter-clockwise direction surrounding pixel values are compared with the central as shown in figure 4.3. The value of intensity or luminosity of each neighbour is compared with the centre pixel. Depending if the difference is higher or lower than 0, a 1 or a 0 is assigned to the location. The result provides an 8-bit value to the cell. The advantage of this technique is even if the luminosity of the image.

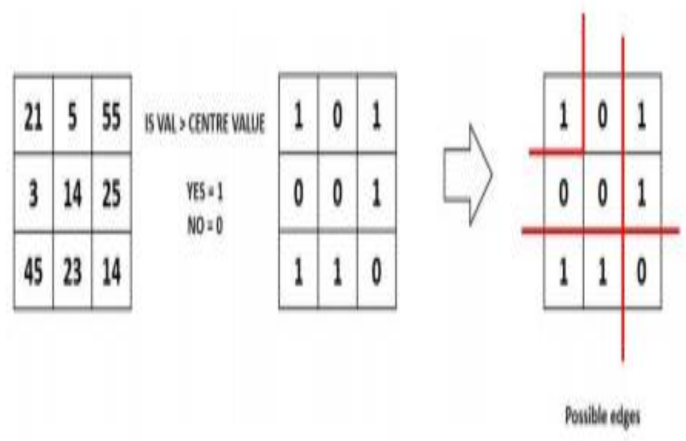


Fig 4.3 Local binary pattern histogram generating 8-bit number

is changed as in figure 4.4, the result is the same as before. Histograms are used in larger cells to find the frequency of occurrences of values making process faster. By analysing the results in the cell, edges can be detected as the values change. By computing the values of all cells and concatenating the histograms, feature vectors can be obtained. Images can be classified by processing with an ID attached. Input images are classified using the same process and compared with the dataset and distance is obtained. By setting up a threshold, it can be identified if it is a known or unknown face. Eigenface and Fisher face compute the dominant features of the whole training set while LBPH analyze them individually.

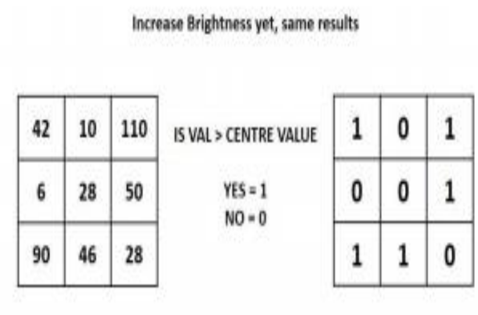


Fig 4.4 The results are same even if brightness is changed

Chapter-5

Methodology

5.1 Brief

Below are the methodology and descriptions of the applications used for data gathering, face detection, training and face recognition. The project was coded in Python using a mixture of IDLE and PYCharm IDEs.

5.2 Face Detection

First stage was creating a face detection system using Haar-cascades. Although, training is required for creating new Haar-cascades, OpenCV has a robust set of Haar-cascades that was used for the project. Using face-cascades alone caused random objects to be identified and eye cascades were incorporated to obtain stable face detection. The flowchart of the detection system can be seen in figure 5.1. Face and eye

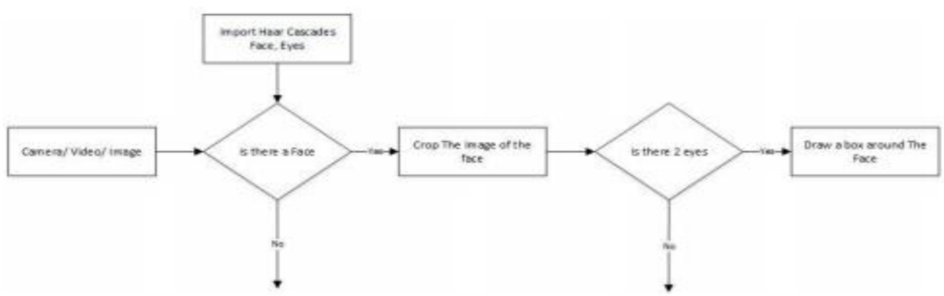


Fig 5.1 The Flow chart of the face detection application

classifier objects are created using classifier class in OpenCV through the cv2.CascadeClassifier() and loading the respective XML files. A camera object is created using the cv2.VideoCapture() to capture images. By using the CascadeClassifier.detectMultiScale() object of various sizes are matched and location is returned. Using the location data, the face is cropped for further verification. Eye cascade is used to verify there are two eyes in the cropped face. If satisfied a marker is placed around the face to illustrate a face is detected in the location.

5.2 Face Recognition Process

For this project three algorithms are implemented independently. These are Eigenface, Fisherface and Linear binary pattern histograms respectively. All three can be implemented using OpenCV libraries. There are three stages for the face recognition as follows: 1. Collecting images IDs 2. Extracting unique features, classifying them and storing in XML files 3. Matching features of an input image to the features in the saved XML files and predict identity

5.2.1 Collecting the image data

Collecting classification images is usually done manually using a photo editing software to crop and resize photos. Furthermore, PCA and LDA requires the same number of pixels in all the images for the correct operation. This time consuming and a laborious task is automated through an application to collect 50 images with different expressions. The application detects suitable expressions between 300ms, straightens any existing tilt and save them. The Flow chart for the application is shown in figure 5.2

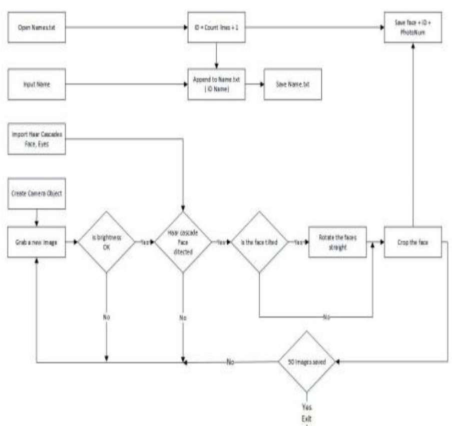


Fig 5.2 The Flowchart for the image collection

Application starts with a request for a name to be entered to be stored with the ID in a text file. The face detection system starts the first half. However, before the capturing begins, the application check for the brightness levels and will capture only if the face is well illuminated. Furthermore, after the face is detected, the position of the eyes are analysed. If the head is tilted, the application automatically corrects the orientation. These two additions were made considering the requirements for Eigenface algorithm. The Image is then cropped and saved using the ID as a filename to be identified later. A loop runs this program until 50 viable images are collected from the person. This application made data collection efficient.

5.2.2 Training the Classifiers

OpenCV enables the creation of XML files to store features extracted from datasets using the FaceRecognizer class. The stored images are imported, converted to grayscale and saved with IDs in two lists with same indexes. FaceRecognizer objects are created using face recogniser class. Each recogniser can take in parameters that are described below:

cv2.face.createEigenFaceRecognizer()

1. Takes in the number of components for the PCA for crating Eigenfaces. OpenCV documentation mentions 80

can provide satisfactory reconstruction capabilities.

2. Takes in the threshold in recognising faces. If the distance to the likeliest Eigenface is above this threshold, the

function will return a -1, that can be used state the face is unrecognisable

cv2.face.createFisherfaceRecognizer().

1. The first argument is the number of components for the LDA for the creation of Fisherfaces.OpenCV mentions it

to be kept 0 if uncertain.

2. Similar to Eigenface threshold. -1 if the threshold is passed.

cv2.face.createLBPHFaceRecognizer()

1. The radius from the centre pixel to build the local binary pattern.

2. The Number of sample points to build the pattern. Having a considerable number will slow down the computer.

3. The Number of Cells to be created in X axis.

4. The number of cells to be created in Y axis.

5. A threshold value similar to Eigenface and Fisherface. if the threshold is passed the object will return -1

Recogniser objects are created and images are imported, resized, converted into numpy arrays and stored in a vector. The ID of the image is gathered from splitting the file name, and stored in another vector. By using FaceRecognizer.train(Numpy Image, ID) all three of the objects are trained. It must be noted that resizing the images were required only for Eigenface and Fisherface, not for LBPH. Next, the configuration model is saved as

a XML file using FaceRecognizer.save(FileName). In this project, all three are trained and saved through one application for convenience. The flow chart for the trainer is shown in figure 5.3.

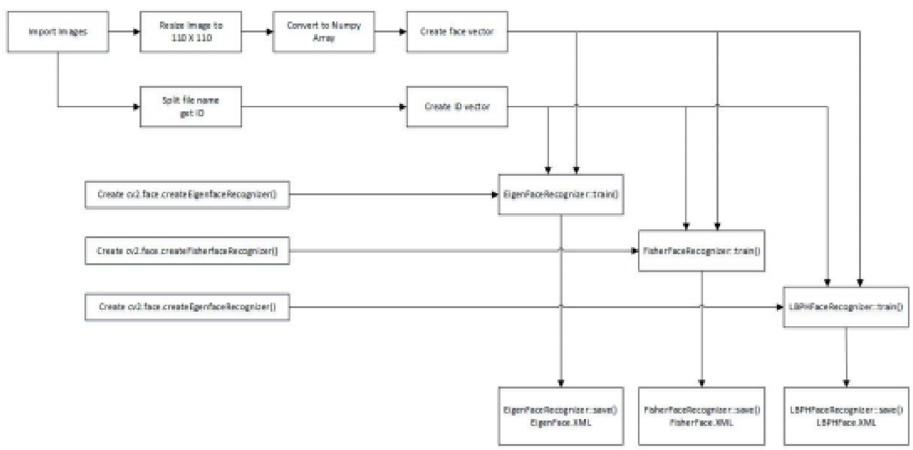


Fig 5.3 Flowchart of the training application

5.2.3 The Face Recognition

Face recogniser object is created using the desired parameters. Face detector is used to detect faces in the image, cropped and transferred to be recognised. This is done using the same technique used for the image capture application. For each face detected, a prediction is made using **FaceRecognizer.predict()** which return the ID of the class and confidence. The process is same for all algorithms and if the confidence his higher than the set threshold, ID is -1. Finally, names from the text file with IDs are used to display the name and confidence on the screen. If the ID is -1, the application will print unknown face without the confidence level. The flow chart for the application is shown in figure 5.4..

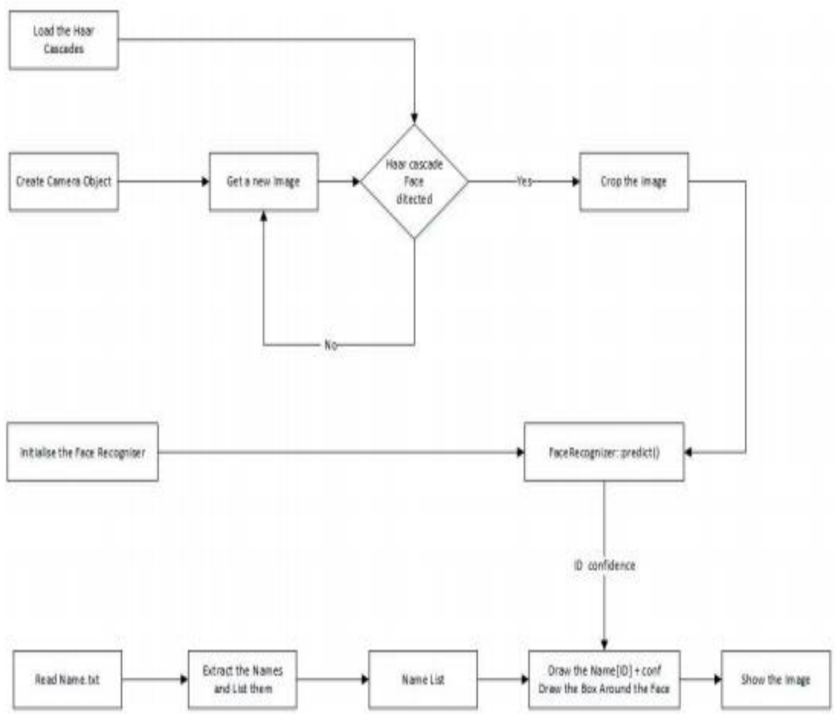


Fig 5.4 Flowchart of the face recognition application

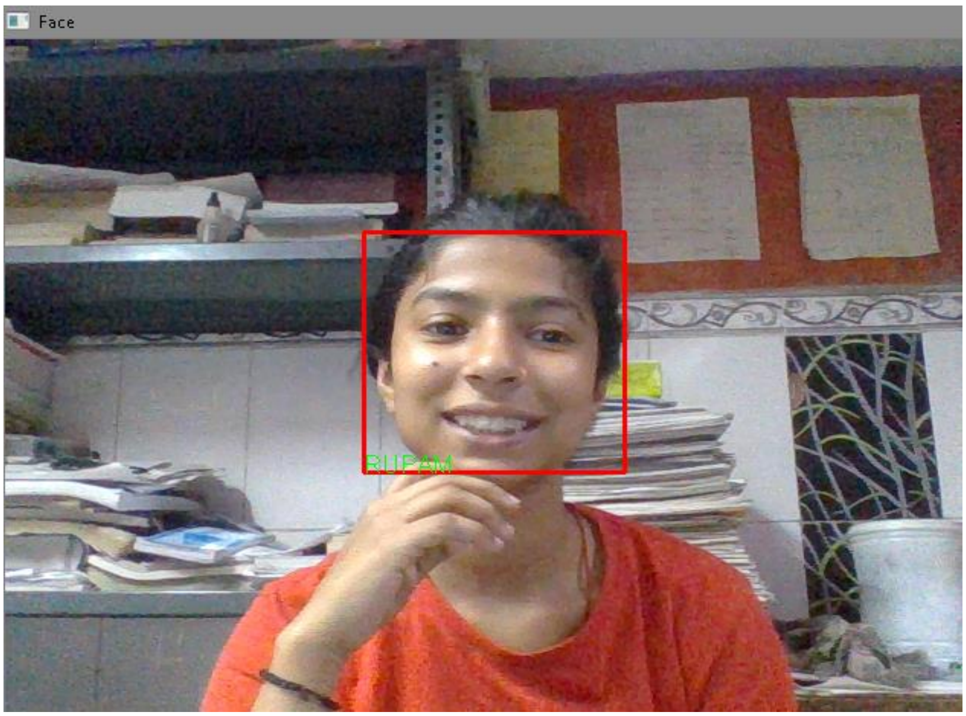
RESULTS

The collected images are shown below. Each face has 10 images. Three applications were written to iterate through the parameters of each algorithm. On each iteration, the algorithm is trained using different parameters and tested against a photo. The resulting data is plotted at the after finishing the tests. The applications are :

- TestDataCollector EigenFace.py
- TestDataCollector FisherFace.py
- TestDataCollector LBPH.py.

Image Analysis:-

1.Female Face

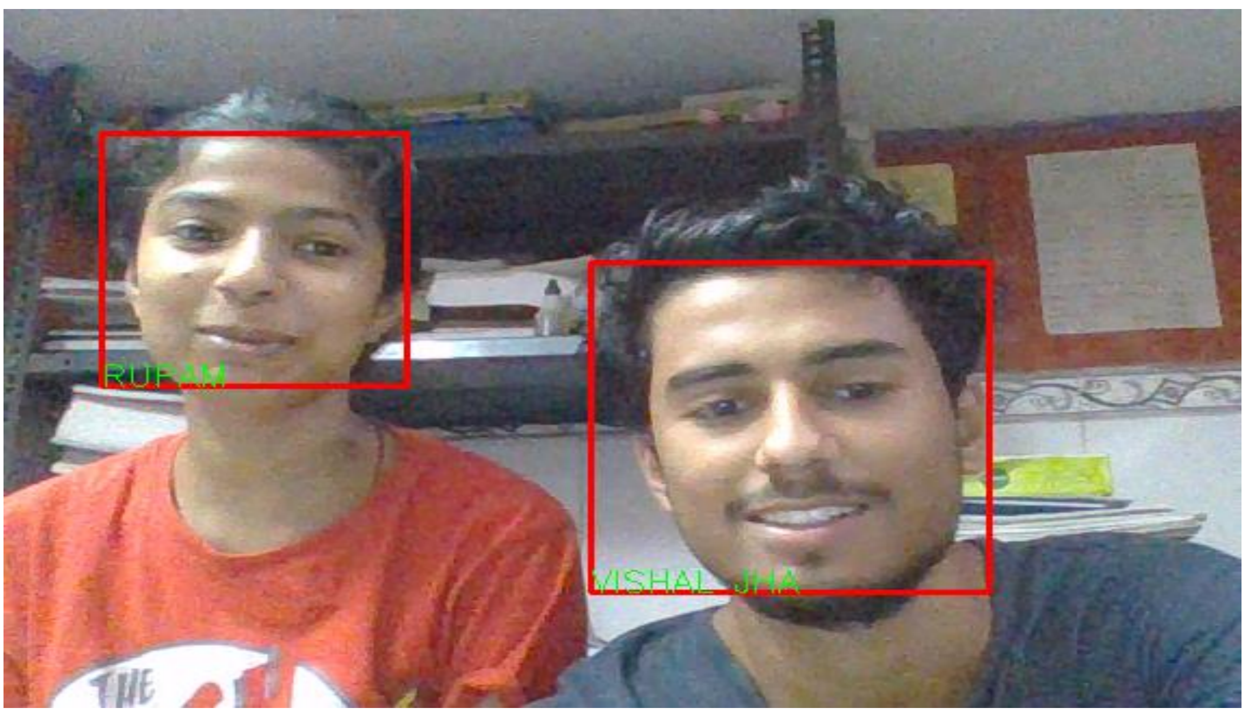


Scrn 1. Female Face Results

2.Male Face

Scrn 2. Male Face Results

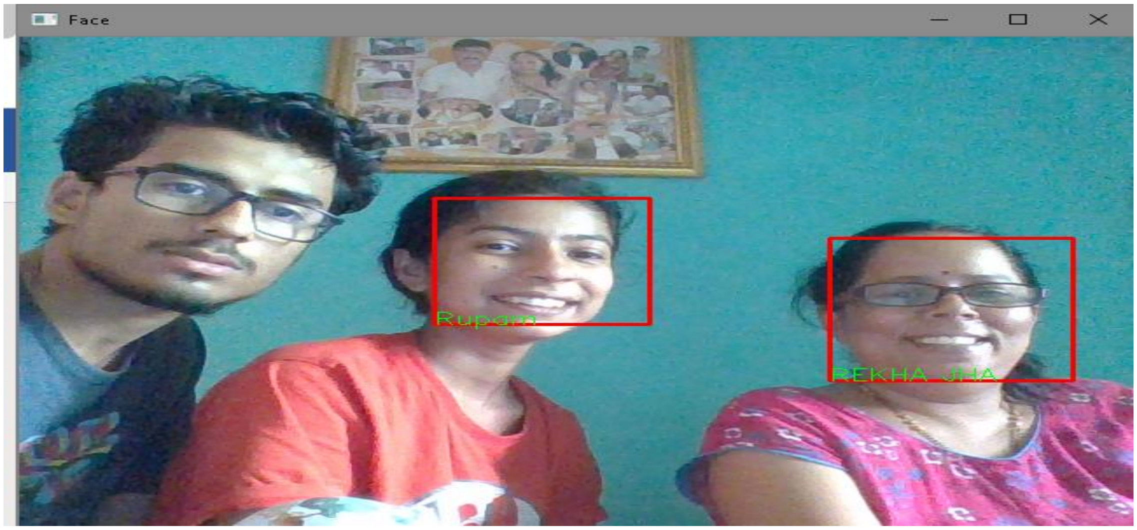
3. Multiple Faces



Scr 3. Multiple Faces Results

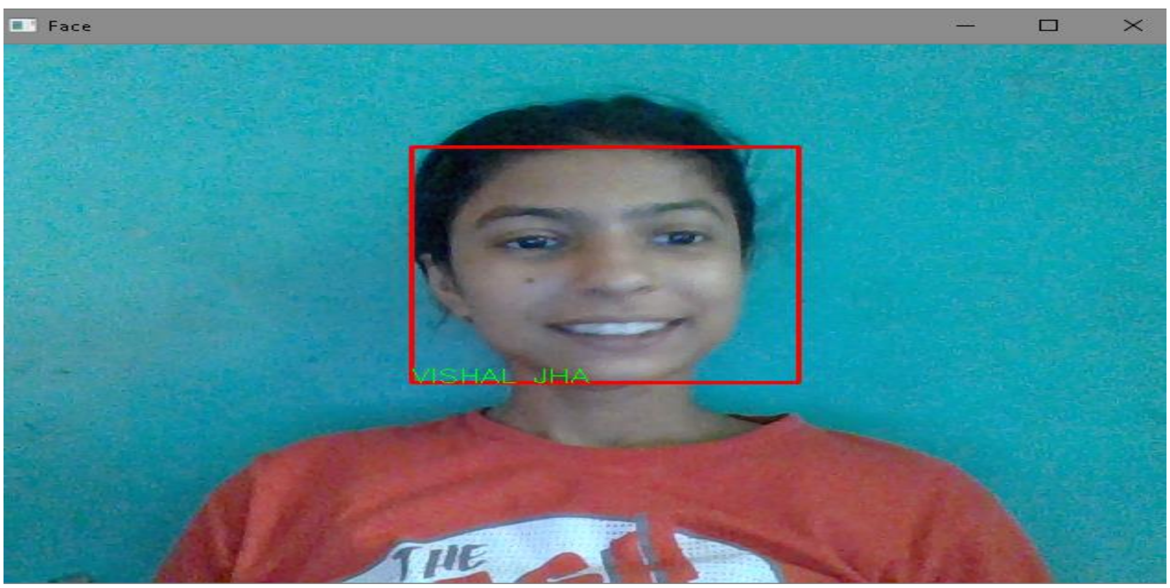
ERRORS:-

1.Not Recognizing Face



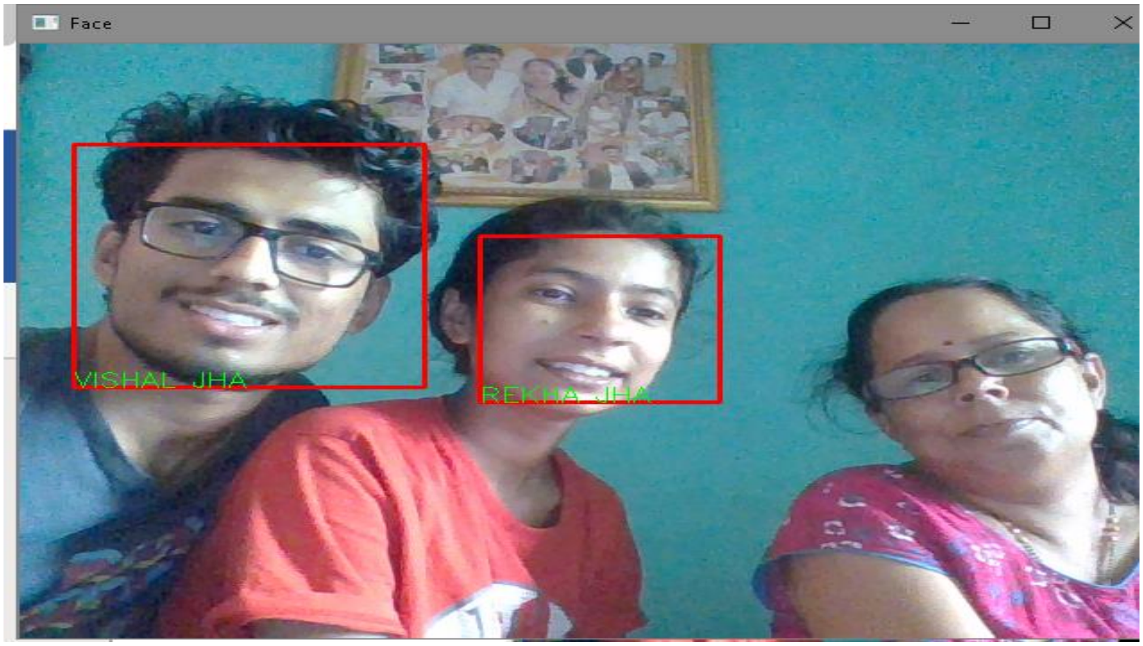
Scrn 4. Not recognizing Error

2.Wrong Face Recognition



Scrn 5. Wrong face Recognition Error

3. Recognizing Wrong Faces and Not Recognizing



Scrn 6. Recognizing Wrong Faces and Not Recognizing Faces

Conclusion

This paper describes the mini-project for visual perception and autonomy module. Next, it explains the technologies used in the project and the methodology used. Finally, it shows the results, discuss the challenges and how they were resolved followed by a discussion. Using Haar-cascades for face detection worked extremely well even when subjects wore spectacles. Real time video speed was satisfactory as well devoid of noticeable frame lag. Considering all factors, LBPH combined with Haar-cascades can be implemented as a cost effective face recognition platform. An example is a system to identify known troublemakers in a mall or 18 a supermarket to provide the owner a warning to keep him alert or for automatic attendance taking in a class.

References

[1] Takeo Kanade. Computer recognition of human faces, volume 47. Birkh•auser Basel, 1977.

[2] Lawrence Sirovich and Michael Kirby. Low-dimensional procedure for the characterization of human faces. Josa a, 4(3):519{524, 1987.

[3] M. Turk and A. Pentland. Eigenfaces for recognition. Journal of Cognitive Neuroscience, 3(1):71{86, Jan 1991}.