

PROFESSIONAL TRAINING REPORT
at
Sathyabama Institute of Science and Technology (Deemed to be
University)

Submitted in partial fulfillment of the requirements for the award of Bachelor of
Technology Degree in

Information Technology

By

Abhishek kumar
(Reg. No.38120002)



DEPARTMENT OF INFORMATION TECHNOLOGY SCHOOL OF
COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY JEPPIAAR NAGAR, RAJIV
GANDHI SALAI,
CHENNAI – 600119. TAMIL NADU

JANUARY 2021



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



SCHOOL OF COMPUTING

BONAFIDE CERTIFICATE

This is to certify that this Professional Training Report is the bonafide work of **ABHISHEK KUMAR (Reg.No. 38120002)** who underwent the professional training in "FACE DETECTION USING PYTHON-OPENCV" under our supervision from July 2020 to January 2021.

Internal Guide

Dr.Senduru Srinivasulu M.E., Ph.D.,

Head of the Department

Dr. R. SUBHASHINI M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, **ABHISHEK KUMAR (Reg.No.38120002)** hereby declare that the Professional Training Report on “**FACE DETECTION USING PYTHON-OPENCV**” done by me

under the guidance of **Dr.Senduru Srinivasulu M.E., Ph.D.**, at Sathyabama Institute Of Science And Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of technology degree in Information Technology.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** 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, Dean**, School of Computing , **Dr.SUBHASHINI M.E., Ph.D. ,** Head of the Department of Information and Technology 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.Senduru Srinivasulu M.E., Ph.D.**,for his 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 Information Technology** who were helpful in many ways for the completion of the project.



Certificate of Participation

This certificate is proudly presented to

ABHISHEK KUMAR

for participating in the

EDUFABRICA WINTER CAMP 2020

on the topic

DATA SCIENCE WITH PYTHON

conducted by

ETHICAL EDUFABRICA PVT. LTD.

at their DELHI Center

Niraj Singh
Director

Ethical Edufabrica Pvt. Ltd.

INTERNSHIP CERTIFICATE



12th Jan 2020

TO WHOM IT MAY CONCERN

This is to certify that ABHISHEK KUMAR Student of Sathyabama IST College had registered for **Edufabrica Winter Camp 2020 @ Delhi** with **Ethical Edufabrica Pvt. Ltd** which was held during the period of 25th Dec 2019 to 12th Jan 2020. As part of the **Edufabrica Winter Camp 2020**, the candidate was required to undertake a project showcasing the aspects of the chosen course curriculum as a criteria towards finishing the Program.

So hereby with this letter, we acknowledge that the student has done his project on the topic of "Face Detection using Python" as part of training in "Python Development Program" and the project has exhibited the student's ability and skill in adapting the course structure into its practical nature and focusing on the core elements of the curriculum. The Project was satisfactory and had catered to all the key briefs handed to the student as an outline for the project

The candidate, during training with us has, shown diligence and perseverance towards adhering to the training protocols set by the company and to making sure that the given tasks were handled and performed with great outcome.

We Wish the candidate all the very best for the future



Ethical Edufabrica Pvt. Ltd.

Ethical Edufabrica Pvt Ltd

Plot No 12, First Floor, Hargobind Enclave, Karkardooma,
Anand Vihar, New Delhi, Delhi 110092



info@edufabrica.net



www.edufabrica.net



+91-93548 23927

ABSTRACT

Identifying a person with an image has been popularised through the mass media. However, it is less robust to fingerprint or retina scanning. This report describes the face detection mini-project undertaken for the visual perception and autonomy module at Plymouth university. 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. The methodology is described including flow charts for each stage of the system. 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.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO
	ABSTRACT	VII
1.	INTRODUCTION	1
	1.1 OUTLINE OF THE PROJECT	1
	1.2 LITERATURE OVERVIEW	1
	1.3 HISTROY	2
2.	SYSTEM REQUIREMENTS	3
	2.1SOFTWARE REQUIREMENTS	3
	2.2HARDWARE REQUIREMENTS	3
	2.3APPLICATION	3
	2.4PLATFORM USED	5
3.	ALGORITHMS	8
	3.1EAGEN FACE	8
	3.2FISHER FACE	9
	3.3LBPH	9
4.	IMPLEMENTATION	14
	4.1 HOW FACE RECOGNIZATION WORK	14
	4.2 PROBLEBS AND CHALLENGES	16
5.	CONCLUSION AND FUTURE WORK	20
	REFERENCES	22
	APPENDIX	23
	A. SOURCE CODE	23
	B. SCREENSHOTS	27

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
3.1	LBPH OPERATION	10
3.2	LBPH METRIX	11
3.3	LBPH OPERATION	12
3.4	LBPH OPERATION IN HISTOGRAM	13
4.1	GENERIC OPERATION	14
4.2	FACE DETECTION	15
4.3	FACE RECOGNIZER	16
4.4	ILLUSION	17
4.5	IDENTICAL FACE	17
4.6	LOW RESOLUTION	18
4.7	EXPRESSION	18
5.1	FACE DETECTION CODE	27
5.2	OUTPUT SCREENSHOT	28

CHAPTER 1

INTRODUCTION

1.1 OUTLINE OF THE PROJECT

The following document is a report on the mini project for Robotic visual perception and autonomy. 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 non-invasive identification system and faster than other systems since multiple faces can be analysed 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., 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.

1.2 LITERATURE OVERVIEW

This section gives an overview on the major human face recognition techniques that apply mostly to frontal faces, advantages and disadvantages of each method . The methods considered are eigenfaces (eigenfeatures), neural networks, dynamic link architecture, hidden Markov model, geometrical feature matching, and template matching. The approaches are analyzed in terms of the facial representations they used.

1.3 HISTORY

Face recognition began as early as 1977 with the first automated system being introduced By Kanade using a feature vector of human faces. In 1983, Sirovich and Kirby introduced the principal component analysis(PCA) for feature extraction . Using PCA, Turk and Pentland Eigenface was developed in 1991 and is considered a major milestone in technology. Local binary pattern analysis for texture recognition was

introduced in 1994 and is improved upon for facial recognition later by incorporating Histograms. 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.

CHAPTER 2

SYSTEM REQUIREMENT

2.1 SOFTWARE REQUIREMENT

- Operating system :Windows or Linux and MacOS
- Coding Language :Python
- Tool :SublimeText2

2.2 HARDWARE REQUIREMENTS

- Processor: Intel Core(TM) i5 2520M CPU @ 2.50Ghz (Dual core)
- RAM: 8.00 GB
- Operating System: Windows 7/8/10 or Linux

2.3 APPLICATION

2.3.1 FACE DETECTION AND RECOGNIZATION

ON SOCIAL MEDIA FACEBOOK

We can see the advances of facial recognition technology on Facebook in the U.S.A. Facebook just recently launched its facial recognition feature called the Photo Review which alerts users every time a photo with their face is posted. This makes it even easier to tag yourself in new photos, keep an eye on unflattering photos without tagging yourself or to reach out to your friend to remind them that they promised not to post it.

VK.COM

In Russia, VK.com has launched their own version of Photo Review called findface. For those of you unfamiliar with VK.com, it's basically the Russian version of Facebook. FindFace takes it a bit further than Photo Review and does exactly as its name implies, it finds faces.

If you want to find someone on VK.com but only have a photo of them, you can simply log in and search with a jpeg or png photo, as long as it is under 5 Mb.

PicWISER

When it comes to facial recognition software and its uses, finding photos is not all such technology has to offer.

What if you could use facial recognition software to protect your designs and assets? Well, that is exactly what PicWISER has set out to do.

Primarily marketed as a tool for designers and photographers, PicWiser works like a google reverse image search but specializes in infringement.

It collects information on any instances your design or photo appears in a search and reports it back to you.

2.3.2 FACE RECOGNIZATION APP

GOOGLE

Taking facial recognition a step further, google has now used its software to enable grouping photos by faces. Not only that, but they have introduced pet recognition so you can easily peruse photos of your dog or cat instead of having to scroll through all of your photo gallery. You can also search terms like, "Dog" and photos of any dogs you have in your phone will appear. For pet lovers like me, this really is a great feature. If you have a pet, over half of your photos are just of them sleeping. But you often want to find a specific one to prove to your coworkers just how cute they really are.

SNAPCHAT

Speaking of animals in photos, have you checked the latest snapchat filters? Of course you have! Being the tech-savvy person you are, you probably know that snapchat uses facial recognition software.

But did you know that it actually takes three separate but instantaneous processes for you to get that realistic goth makeup look or Santa hat balanced just so? First, the program runs its facial detection to find the boundaries of your face and sets a grid

FACE2GANE

There are also more altruistic uses for facial recognition software. Face2Gene is a healthcare app which uses facial recognition technology to help detect certain genetic disorders or syndromes from photos of patients. The software is able to detect certain phenotypes that shed light on potential causes of certain facial and non-facial features the patient has. Using facial recognition technology in the healthcare field is an incredibly innovative idea, one on which I could write another lengthy blog about. But I really do hope that people begin to invest in such technology because lives in healthcare are endless.

LUXAND

And last but not least, making headway in automation, applications from Luxand FaceCrop SDK are using facial recognition software to predict the appearance of children as they age, automating post-processing tasks such as skin tone adjustment and red-eye removal.

You can also use your face to create 3D avatars for games and build an AR (augmented reality) application based on your real-time, real-life face transformations.

2.4 PLATFORM USED

Python:

Python is a general-purpose interpreted, interactive, high level, object-oriented scripting language. Fully dynamic type system and automatic memory management. Python is designed to be highly readable. It uses English keywords frequently. It was created by Guido van Rossum during 1985- 1990. Open Source : Support all OS like (Windows, Linux, Mac). Minimal code. Clarity. Consistency. Easy to teach. It is a general-purpose language, which means it can be used to build just about anything, which will be made easy with the right build just about anything, which tools/libraries

Beginner Friendliness

Python was designed to be easy to understand and fun to use Python. and coding in Python a satisfying experience. Thus, Python has gained popularity for being a beginner-friendly language

Easy to understand

python reads like English, which takes a lot of syntax-learning stress off coding beginners. Python handles a lot of complexity for you, so it is very beginner-friendly

Advantages of Python

- Extensive Support Libraries
- Integration Feature
- Improved Programmer's Productivity
- Productivity

Limitations or Disadvantages of Python

- Gets Slow in Speed
- Weak in Mobile Computing
- Run-time Errors
- Not Easy to Maintain

FACE DETECTION IN OPENCV

OpenCV (Open Source Computer Vision) is a library of programming functions for real-time computer vision. The face detection part of the project was made using an OpenCV Library for Scala. The reason was that most Face APIs are restricted to doing detection on pictures only, whereas the project was required to have face detection done on a live video footage to speed up the process of checking student attendance and prevent queues before lectures. The OpenCV library proved to be

flexible enough for the project as it can accurately detect a face in real time and

highlight it by drawing a rectangle around the faces of the students passing by. This all happens in a window separate from the face recognition so the lecturer can keep track of both students passing by while having their faces detected and the feedback from the recognition part of the system. While faces are being detected, the application takes a snapshot of the live footage every second and then sends it to the recognition system.

Sublime Text 3

Sublime Text is a shareware cross-platform source code editor with a Python application programming interface (API). It natively supports many programming languages and markup languages, and functions can be added by users with plugins, typically community-built and maintained under free-software licenses.

Software genre: Source-code editor

Languages used: Python

CHAPTER 3

ALGORITHM

ALGORITHM

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.

TYPES OF ALGORITHMS

- 1. EAGEN FACES:1991**
- 2. FISHER FACES:1997**
- 3. LOCAL BINARY PATTERN HISTOGRAMS:1994**
- 4. SCALE INVARIANT FEATURE TRANSFORMS :1997**
- 5. SPEED UP REBUST FEATURE:2006**

3.1 EAGEN FACES

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. Each row is concatenated to create a vector, resulting a $1 \times n^2$ 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

3.2 FISHERFACES

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 maximise the difference between the mean of the classes normalised by their variance .. This enables LDA to differentiate between feature classes better than PCA .

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.

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.

3.3 LOCAL BINARY PATTERN

Local binary pattern histogram is a simple yet very efficient texture operator which able the pixels of an image by thresholding the neighbourhood of each pixel and considers the result as a binary number.

It was first described in 1994 and has since been found to be a powerful feature for texture classification. It has further been determined that when LBP is combined with histograms of oriented gradients descriptor, it improves the detection performance considerably on some datasets.

Using the LBP combined with histograms we can represent the face images with a simple data vector .As LBP is a visual descriptor it can also be used for face recognition tasks, as can be seen in the following step-by-step explanation.

3.3.2 TRAINING THE ALGORITHM:

First, we need to train the algorithm. To do so, we need to use a dataset with the facial images of the people we want to recognize. We need to also set an ID (it may be a number or the name of the person) for each image, so the algorithm will use this information to recognize an input image and give you an output. Images of the same person must have the same ID. With the training set already constructed, let's see the LBPH computational steps.

Applying the LBP operation:

The first computational step of the LBPH is to create an intermediate image that describes the original image in a better way, by highlighting the facial characteristics. To do so, the algorithm uses a concept of a sliding window, based on the parameter's **radius** and **neighbours**.

THE IMAGE BELOW SHOWS THIS PROCEDURE:

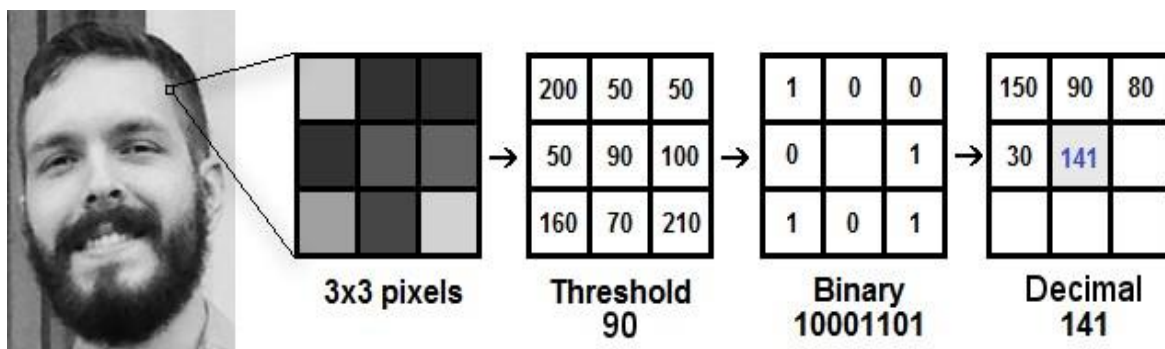


Fig:3.1 LBPH Operation

Based on the image above, let's break it into several small steps so we can understand it easily:

1. Suppose we have a facial image in grayscale.
2. We can get part of this image as a window of 3x3 pixels.
3. It can also be represented as a 3x3 matrix containing the intensity of each pixel (0~255).

4. Then, we need to take the central value of the matrix to be used as the threshold.
5. This value will be used to define the new values from the 8 neighbours.

For each neighbour of the central value (threshold), we set a new binary value. We set 1 for values equal or higher than the threshold and 0 for values lower than the threshold.

Now, the matrix will contain only binary values. We need to concatenate each binary value from each position from the matrix line by line into a new binary value. Some authors use other approaches to concatenate the binary values, but the final result will be the same. Then, we convert this binary value to a decimal value and set it to the central value of the matrix, which is actually a pixel from the original image.

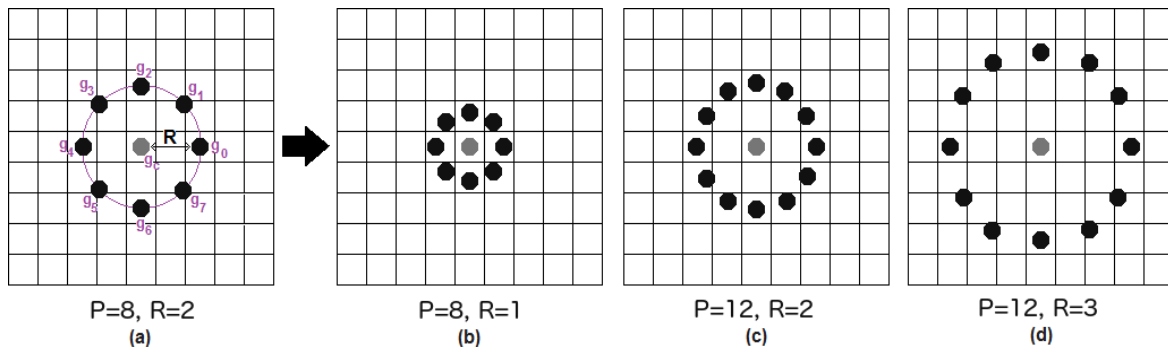


Fig:3.2 LBPH In matrix

It can be done by using bilinear interpolation. If some data point is between the pixels, it uses the values from the 4 nearest pixels (2x2) to estimate the value of the new data point.

3.3.3 Extracting the Histograms: Now, using the image generated in the last step, we can use the Grid X and Grid Y parameters to divide the image into multiple grids

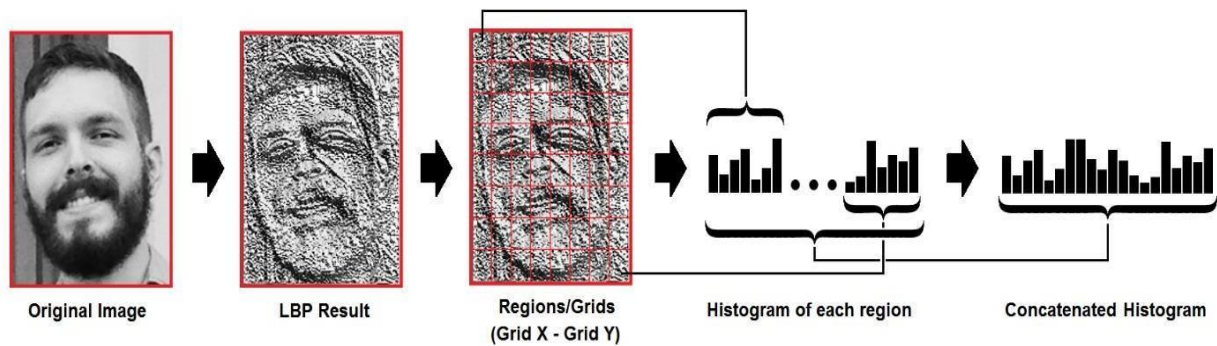


Fig:3.3 LBPH Operation In Histogram

Based on the image above, we can extract the histogram of each region as follows:

1. As we have an image in grayscale, each histogram (from each grid) will contain only 256 positions (0~255) representing the occurrences of each pixel intensity.
2. Then, we need to concatenate each histogram to create a new and bigger histogram. Supposing we have 8x8 grids, we will have $8 \times 8 \times 256 = 16,384$ positions in the final histogram. The final histogram represents the characteristics of the image original image.

3.4 Face Detection using 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 patterns to recognise signals with sudden transformations. By combining several wavelets, a cascade can be created that can identify edges, lines and circles with different colour intensities. These sets are used in Viola Jones face detection technique in 2001 and since then more patterns are introduced for. 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.



Fig:3.4 LBPH Operation In Histogram

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 used a summed area table to compute the matches fast.

CHAPTER 4

IMPLEMENTATION

HOW FACE DETECTION AND RECOGNITION WORK

A face recognition system (FRS) is a computer application for automatically identifying or verifying a person from a digital image or a video frame from a video source. A face image or image sequence is given as input to the FRS, and then FRS applies proposed methods to match the given input with face images stored in the database.

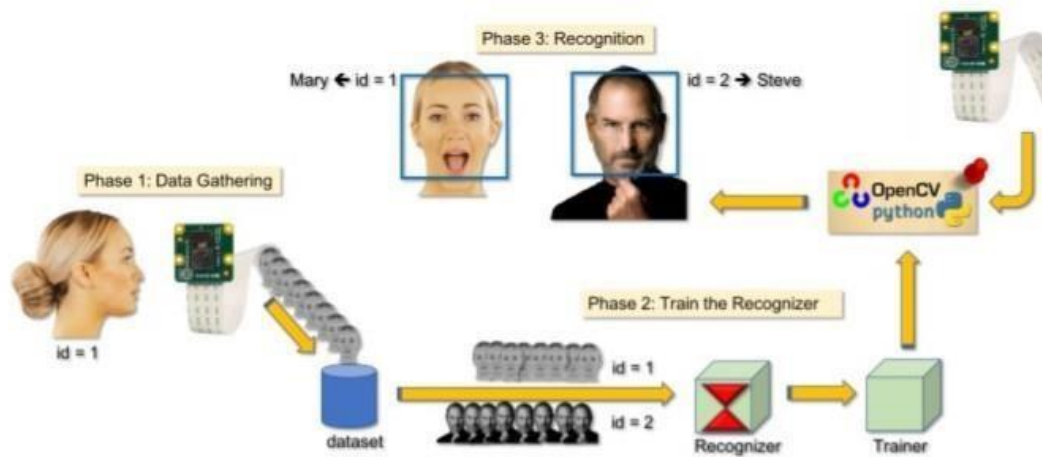


Fig:4.1Generic Representation

It is used for two primary tasks:

1. Verification (one-to-one matching):

When presented with a face image of an unknown individual along with a claim of identity, ascertaining whether the individual is who he/she claims to be.

2. Identification (one-to-many matching):

Given an image of an unknown individual, determining that person's identity by comparing (possibly after encoding) that image with a database of (possibly encoded) images of known individuals.

BASIC STEP FOR FACE RECOGNITION

As mentioned earlier, facial recognition methods vary slightly, depending on the application and manufacturer, but they generally involve a series of steps that serve to capture, process, analyze and match the captured face to a database of recorded images. These basic steps are:

1. Detection:

When the facial recognition system is attached to a video surveillance system, the recognition software scans the field of view of the camera for what it detects as faces. Upon the detection of each face-like image on a head-shaped form, it sends the face to the system to process it further. The system then estimates the head's position, orientation, and size. Generally, a face needs to be turned at least 35 degrees toward the camera for the camera to detect it.

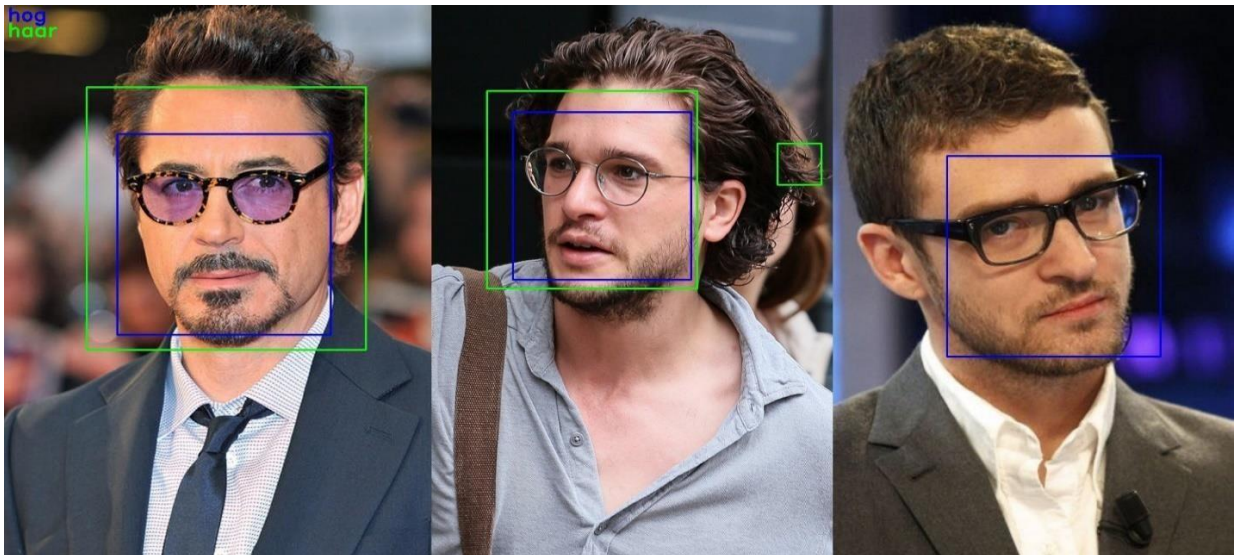


Fig:4.2 Face Detection

2. Matching:

This is the final stage in which newly acquired facial data is compared to the stored data; if it matches with one of the images in the database, the software returns the details of the matched face and notifies the end user.

3. Face Recognition:



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.

Fig:4.3 Face Recognition

4.2 PROBLEMS & CHALLENGES

Over the last decade, academic computer vision researchers and commercial product developers have improved the performance of automated face recognition algorithms on a variety of challenging face recognition tasks. Because humans currently perform face recognition tasks in most real-world security situations, it is unclear whether the use of algorithms improves security or puts it at greater risk. The real challenge in face detection and recognition technologies is the ability to handle all those scenarios where subjects are non-cooperative and the acquisition phase is unconstrained. There are numerous factors that cause the appearance of the face to vary. These sources of variation in the facial appearance can be categorized into two groups: Intrinsic factors and Extrinsic ones.

ILLUMINATION



Fig:4.4 Face lillumination

IDENTICAL FACES



Fig:4.5 Face Identical

LOW RESOLUTION



Fig:4.6 Face Low Resolution

EXPRESSION



Fig:4.7 Face Expression

1.6 OBJECTIVES

The main objective of the project is to develop an algorithm that will be used to detect faces. The need is to develop a database where all the related data will be stored and to develop a interface.

Chapter 5

CONCLUSION AND FUTURE WORK

SUMMARY

Following the project, the purpose of this report is to analyze the advantages and disadvantages, problems encountered and solutions found, when using face detection and recognition in an academic environment to keep track of the attendance of students. A small introduction will outline the content of the report and it will continue with the advantages and disadvantages of the face detection and recognition system compared to the other biometrics authentication methods. Subsequently, detailed description of the two main parts of the project and how they work will follow, as well as problems encountered during the project and how they were solved. Even though it is difficult to build a flawless system, solutions were found for the most impactful issues and work arounds were found for the problems caused by the limitations of the software used.

FUTURE SCOPE

PRE EVENT RETAIL CRIME

Face shopper enters a store that prevents a threat. Face recognition systems are already radically reducing retail crime According to our data,face recognition reduces external shrink by 34% and, more importantly, reduces violent incidents in retail stores by up to 91%.

FIND MISSING PERSON

Face recognition can be used to find missing children and victims of human trafficking. As long as missing individuals are added to a database, law enforcement can become alerted as soon as they are recognized by face recognition—be it an airport, retail store or other public space. In fact, 3000 missing children were discovered in just four days using face recognition in india!

recognition is currently being used to instantly identify when known shoplifters, organized retail criminals or people with a history of fraud enter retail establishments. Photographs of individuals can be matched against large databases of criminals so that loss prevention and retail security professionals can be instantly notified when a

IDENTIFYING PEOPLE ON SOCIAL MEDIA PLATFORM

Facebook uses face recognition technology to automatically recognize when Facebook members appear in photos. This makes it easier for people to find photos they are in and can suggest when particular people should be tagged in photos

TRACK SCHOOL ATTENDANCE

In addition to making schools safer, face recognition has the potential to track students' attendance. Traditionally, attendance sheets can allow students to sign another student, who is ditching class, in. But China is already using face recognition to ensure students aren't skipping class. Tablets are being used to scan students' faces and match their photos against a database to validate their identities.

FACILITATE SECURE TRANSACTIONS

In China, there is a financial services company called Ant Financial that enables customers to pay for meals by scanning their faces. Customers place orders through a digital menu, and then use face scan as a payment option. After providing their telephone number they can then purchase their meal.

VALIDATE IDENTIFICATION AT ATMS

It seems likely that face scans will eventually replace ATM cards completely since face recognition is such a powerful identity authentication tool. But in the meantime, face recognition can be used to make sure that individuals using ATM cards are who they say they are. Face recognition is currently being used at ATMs in Macau to protect people's identities.

RECOGNIZE DRIVERS

More car companies are experimenting with way to use face recognition.

One use of face recognition for automobiles is using a face to replace a key as a means of starting a car. Face recognition can also be used to change radio stations and seat preferences based on who is driving. Face recognition can even make drivers safer by recognizing and alerting drivers if they are drifting off or not focusing on the road.

CONTROL ACCESS TO SENSITIVE AREA

Face recognition can work as a means of access control to ensure that only authorized individuals get into facilities like labs, boardrooms, bank vaults, training centers for athletes and other sensitive locations.

MAKE AIR TRAVEL MORE CONVENIENT

Airlines have already started using face recognition to help people check bags, check into flights and board planes faster. It seems like we are quickly moving toward a future in which air travel is not only safer than ever before, but also more convenient than any period in history.

REFERENCE

1. Adelson, E. H., and Bergen, J. R. (1986) The Extraction of Spatio-Temporal Energy in Human and Machine Vision, Proceedings of Workshop on Motion: Representation and Analysis Charleston, SC
2. AAFPRS(1997). A newsletter from the American Academy of Facial Plastic and Reconstructive Surgery.
3. Baron, R. J. (1981). Mechanisms of human facial recognition. International Journal of Man Machine Studies.
4. Beymer, D. and Poggio, T. (1995) Face Recognition From One Example View, A.I. Memo No. 1536, C.B.C.L. Paper No. 121. MIT
5. Bichsel, M. (1991). Strategies of Robust Objects Recognition for Automatic Identification of Human Faces. PhD thesis, , Eidgenossischen Technischen Hochschule, Zurich.
6. Brennan, S. E. (1982) The caricature generator. M.S. Thesis. MIT.
7. Brunelli, R. and Poggio, T. (1993), Face Recognition: Features versus Templates. IEEE Transactions on Pattern Analysis and Machine Intelligence, 15(10):1042-1052
8. Craw, I., Ellis, H., and Lishman, J.R. (1987). Automatic extraction of face features. Pattern Recognition Letters, 5:183-187, February.
9. Deffenbacher K.A., Johanson J., and O'Toole A.J. (1998) Facial ageing, attractiveness, and distinctiveness. Perception. 27(10):1233-1243
10. Duntelman, G.H. (1989) Principal Component Analysis. Sage Publications. • Frank, H. and Althoen, S. (1994). Statistics: Concepts and applications. Cambridge University Press. p.110

APPENDIX

A.SOURCE CODE

```
import cv2

cascade_classifier =
cv2.CascadeClassifier(
'haarcascades/haarcas
cade_frontalface_defa
ult.xml')

cap =
cv2.VideoCapture(0)

while True:

    # Capture frame-by-
frame

    ret, frame =
cap.read()

    # Our operations on
the frame come here

    gray =
cv2.cvtColor(frame, 0)

    detections =
cascade_classifier.det
ectMultiScale(gray,scal
```

```
eFactor=1.3,minNeigh  
bors=5)  
    if(len(detections) >  
0):  
        (x,y,w,h) =  
detections[0]  
        frame =  
cv2.rectangle(frame,(x,  
y),(x+w,y+h),(255,0,0),  
2)
```

```
# Display the  
resulting frame
```

```
cv2.imshow('frame',fra  
me)
```

```
    if cv2.waitKey(1) &  
0xFF == ord('q'):  
        break
```

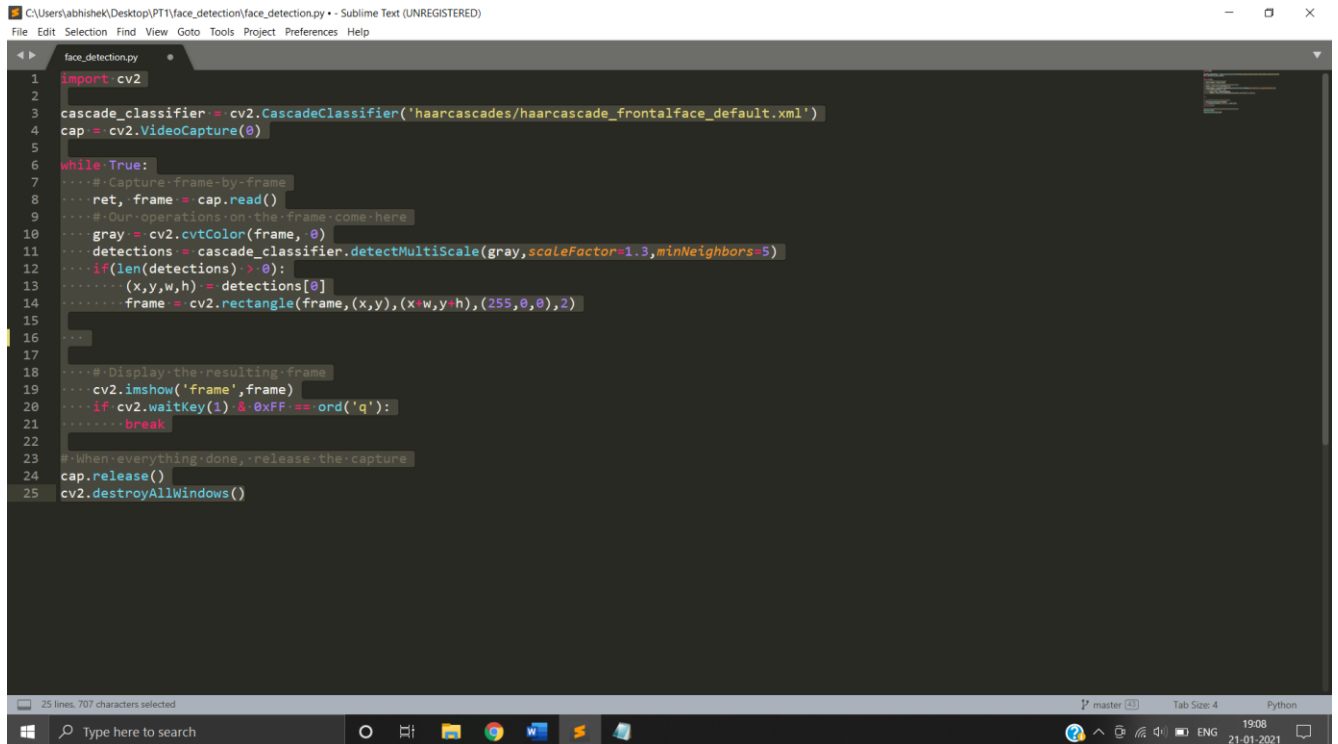
```
# When everything  
done, release the  
capture
```

```
cap.release()
```

```
cv2.destroyAllWindows
```

```
()
```

B.SCREENSHOT



```
1 import cv2
2
3 cascade_classifier = cv2.CascadeClassifier('haarcascades/haarcascade_frontalface_default.xml')
4 cap = cv2.VideoCapture(0)
5
6 while True:
7     # Capture frame-by-frame
8     ret, frame = cap.read()
9     # Our operations on the frame come here
10    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
11    detections = cascade_classifier.detectMultiScale(gray, scaleFactor=1.3, minNeighbors=5)
12    if len(detections) > 0:
13        (x,y,w,h) = detections[0]
14        frame = cv2.rectangle(frame, (x,y), (x+w,y+h), (255,0,0), 2)
15
16
17
18    # Display the resulting frame
19    cv2.imshow('frame', frame)
20    if cv2.waitKey(1) & 0xFF == ord('q'):
21        break
22
23 # When everything done, release the capture
24 cap.release()
25 cv2.destroyAllWindows()
```

Fig 5.1:FACE DETECTION

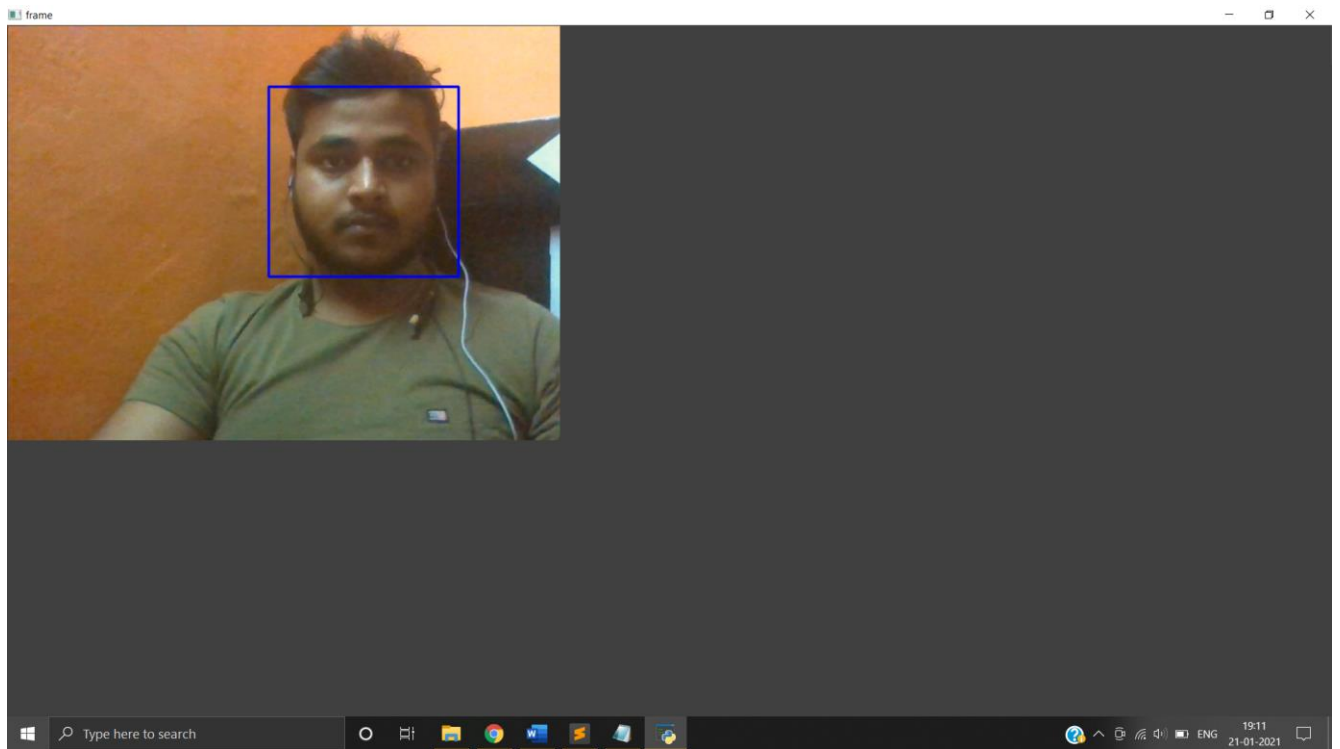


Fig 5.2:Output screenshot