# Face Detection And Recognition

A report submitted for the course named Project I (CS-200)

*By*

**Ankit Tripathi**

**Bachelor of Technology, IV Semester**

**Roll No. 17010107**

*Under the Supervision and Guidance of*

**Dr. Kishorjit Nongmeikapam**

**Department of Computer Science and Engineering**

**Indian Institute of Information Technology Manipur**

April, 2019

# Abstract

In this work, Face recognition system is implemented for academic purpose. It is divided into the 3 parts consisting of Detecting face using viola jones algorithm ,training and labeling the face data and finally recognizing the new face.

In the first part I have used viola jones algorithm for detecting the face which is a machine learning approach. In the second step the detected region of face is converted into numpy array and this array is trained and labled for recognition. In the last part, the system recognize the new face using LBPH (Local Binary Pattern Histogram) algorithm

# Declaration

I declare that this submission represents my idea in my own words and where others' idea or words have been included, I have adequately cited and referenced the original source. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/sources in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and can also evoke penal action from the sources which have thus not been properly cited or from proper permission has not been taken when needed.

(Signature)

(Ankit Tripathi)

Date: (17010107)

Department of Computer Science & Engineering
Indian Institute of Information Technology Manipur

Dr. Kishorjit Nongmeimkapam Email: kishorjit@iiitmanipur.ac.in
Assistant Professor

# Certificate

This is to certify that the report entitled "Face Detection And Recognition", submitted to the Department of Computer Science and Engineering, Indian Institute of Information Technology, Manipur, in partial fulfillment for the award of the degree of **Bachelor of Technology** in Computer Science and Engineering, is a record of bona fide work carried out by Mr. **Ankit Tripathi** Roll no. 17010107, under my nsupervision and guidence.

No part of this report has been submitted elsewhere for award of any other degree.

Dr. Nongeimkapam Kishorjit Singh
Assistant Professor
Supervisor

Date:

Department of Computer Science & Engineering
Indian Institute of Information Technology Manipur

Dr. Kishorjit Nongeimkapam Email: Kishorjit@iiitmanipur.ac.in
Assistant Professor

# Certificate

This is to certify that the report entitled "Face Detection And Recognition", submitted to the Department of Computer Science and Engineering, Indian Institute of Information Technology, Manipur, in partial fulfillment for the award of the degree of **Bachelor of Technology** in Computer Science and Engineering, is a record of bona fide work carried out by Mr. **Ankit Tripathi** Roll no. 17010107

No part of this report has been submitted elsewhere for award of any other degree.

Dr. Nongeikapam Kishorjit Singh
Assistant Professor Head
Department of CSE
IIIT, Manipur

Department of Computer Science & Engineering
Indian Institute of Information Technology Manipur

## Certificate

This is to certify that the report entitled ”Face Detection And Recognition”, submitted to the Department of Computer Science and Engineering, Indian Institute of Information Technology, Manipur, in partial fulfillment for the award of the degree of **Bachelor of Technology** in Computer Science and Engineering, is a record of bona fide work carried out by Mr. **Ankit Tripathi** Roll no. 17010107

Examiners signature

# Acknowledgement

It is my privilege to express my sincerest regards to my project coordinator, Dr. Kishorjit Nongmeikapam, for their valuable inputs, able guidance, encouragement, whole-hearted cooperation and constructive criticism throughout the duration of our project. I deeply express our sincere thanks to our Computer Science Department for encouraging and allowing me to present the project on the topic "Face Recognition" at our department premises for the partial fulfillment of the requirements leading to the award of B-Tech degree. I take this opportunity to thank all our lecturers who have directly or indirectly helped our project. We pay our respects and love to our parents and all other family members and friends for their love and encouragement through out our career. Last but not the least we express our thanks to our friends for their cooperation and support.

- Ankit Tripathi

# Contents

Abstract ii

Declaration iii

Certificate iv

Certificate v

Certificate vi

Acknowledgement vii

Table of contents viii

List of tables xi

List of figures xii

List of abbreviations xiii

1 Introduction 1

1.0.1 Face Detection . . . . . . . . . . . . . . . . . . . . . . . . . . 3

1.0.2 Feature Extraction . . . . . . . . . . . . . . . . . . . . . . . . 3

1.0.3 Face Recognition . . . . . . . . . . . . . . . . . . . . . . . . . 4

**2 Existing System Study** **5**

2.1 FaceVACS . . . . . 5

2.2 FaceKey security features . . . . . 6

2.3 features versus templates . . . . . 6

2.4 Face Recognition using Unsupervised Feature Learning Approach . . . . 6

2.5 Understanding face recognition . . . . . 7

**3 Algorithms** **8**

3.1 Introduction . . . . . 8

3.2 Viola Jones algorithm for detecting face . . . . . 9

3.2.1 Haar Feature . . . . . 9

3.2.2 Adaboost . . . . . 10

3.2.3 Cascading Classifiers . . . . . 11

3.3 LBPH(Local Binary Pattern Histogram) for face recognition . . . . . . . 12

3.4 Summary . . . . . 13

**4 Coding and Implementation** **14**

4.1 code for detecting face . . . . . 15

4.1.1 Explaination . . . . . 15

4.2 code for training image dataset . . . . . 16

4.3 Code for recognizing . . . . . 18

**5 Conclusion** **19**

**Appendix A Screenshot and Description of the Implemented System** **21**

A.1 File structure . . . . . 21

A.2 Image DataSet . . . . . 22

A.3 Some output results . . . . . 23

**Appendix B User manual** **26**

B.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 26

B.2 Step to run your implemented system . . . . . . . . . . . . . . . . . . . . 26

**Bibliography** **27**

# List of Tables

# List of Figures

1.1 Steps of implementation . . . . . 2

3.1 Basic haar features(left) and extended features(right) . . . . . 10
3.2 features on human face . . . . . 11
3.3 LBPH procedure . . . . . 12

A.1 file structure of project . . . . . 21
A.2 Image dataset . . . . . 22
A.3 result 1 . . . . . 23
A.4 result 2 . . . . . 24
A.5 result 3 . . . . . 25

# List of abbreviations

**A**

# Chapter 1

# Introduction

“For starters, [face recognition] would help with customer service. If a shopper is standing in a store and looking upset, a store associate could be dispatched to help that customer. The technology could also play a role in product or selection support: Imagine a shopper is lingering in front of a shelf with 20 brands of detergent appearing perplexed or scowling. A message on their phone or a voice-activated shelf could ask them if they need help finding a product or require help in making a decision that best suits their stain-fighting needs.

“ – Maya Mikhailov

A facial recognition system is a technology capable of identifying a person from a digital image or from a video. Facial recognition is different from face detection. In detection we only detect the face. To implement face recognition we must detect the face first.
It has wide range of application such as robotics, biometrics, security etc. It is widely adopted due to its contactless process. Recently it has also become popular as a commercial identification and marketing tool. First commercial use can be seen in smartphones where it is used for locking/unlocking of phone.
Now a days almost all social media platform is using this technique to give good interaction to the users. As an example is Instagram where when you tag your friend in a group photo then it automatically assign the person's username to its face.so when you click on that photo ,you will be able to see who are thier in that photo. Same feature is applied by facebook.
In countries like Israel , these systems are widely used in security. The surveillance camera used by police have features to recognize the person so that they can easily catch the person doing any wrong activity. In India today we are using biometrics like fingerprint in our adhaar card. May be in future we government will store our face data instead of fingerprint.

Fig 1.1 show the working procedure of any facial recognition system. The first thing

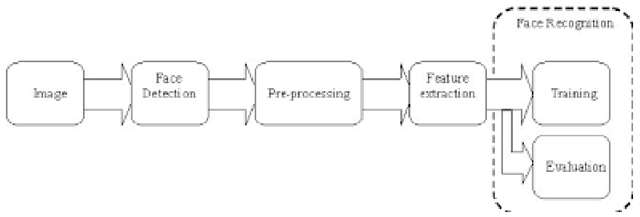


Figure 1.1: Steps of implementation

is to detect the face from the image/video. Next step is to extract the features which is haar feature for this project.Features can also be different.Some features like landmarks of the faces increase the accuracy of the system.These features are then trained in a classifier to recognize the face.So, basically we are using some kind of machine learning technique to do our task.Instead of doing everything from scratch, we will use some library like opencv which have some pre trained classifier which will do our task. So let see what face detection is.

### 1.0.1 Face Detection

Face detection is a computer technology being used in a variety of applications that identifies human faces in digital images. It is the first step for facial recognition.

Face-detection algorithms focus on the detection of frontal human faces. For this we need to make the image dataset which have front faces in it. Side face may cause problem in implementing. Their are many approches to do face detection but almost every algorithm do a basic things. Firstly, the possible human eye regions are detected by testing all the valley regions in the gray-level image. Then the algorithm is used to generate all the possible face regions which include the eyebrows, the iris, the nostril and the mouth corners. We extract lots of features from the face then we train them by any machine learning algorithm. For this project we will use viola jones alogorithm. This algorithm is used mainly for object detection but can be used for detecting face.The key point of this algorithm is that it has high accuracy.After detecting the face we will extract features from the face.

### 1.0.2 Feature Extraction

Everything in this world have some features. In our real life we recognize anything using those features. These features are extracted by our brain and then only we can recognize that object. Similarly, to let our computer recognize anything we first extract those features. One very important area of application is image processing, in which algorithms are used to detect and isolate various desired portions or shapes (features) of a digitized image or video stream. It is particularly important in the area of optical character recognition.In face recognition also we have to extract features. The most used feature is the facial landmarks which is 68 points of the face by which our system can differentiate the people.But for making our system more simple we will extract haar feature.This feature is also used to object detection. These faetures are actually found using image processing.These haar features are like convolutuional kernel which is applied on our image to extract the facial features.The basic idea behind all these is edge detection so to understand this system we must have some basic idea of image processing.Now after extracting features we will train the face data. To do this we will first convert these data into numpy array as our system works better with numbers.Then we will train these numpy array so that it can recognize the person. Also we will make a labeling to those data which will appear as name when system will recognize the face.

### 1.0.3 Face Recognition

A facial recognition system is a technology capable of identifying a person from a digital image or from a video. It is different from face detection. After the process of detection and feature extraction, we finally do recognition. Our extracted facial features from the image dataset are trained and when we show new face to our system then it again do all the process and this time extracted features are matched with the trained features.

For our system we will use LBPH(Local Binary Pattern Histogram) to match the train data with our new image data.This algorithm is also used to detect face from the image.We will learn more on this later in this report.

# Chapter 2

# Existing System Study

## 2.1 FaceVACS

Cognitec developed FaceVACS Engine which enables user to develop new face recognition applications. It provides a lots of API to integret it in other softwares. With software developement kit we can use this engine which is very costimized and is easy to work with. Specific use cases include of this system is image quality check, verification for document issuance, and verification for access control.

## 2.2 FaceKey security features

Facekey have applied lot of features in its face recognition system. According to thier website it is mainly used for security and biometric purpose.

## 2.3 features versus templates

In paper [1] is to compare two simple but general strategies on a common database (frontal images of faces of 47 people: 26 males and 21 females, four images per person). We have developed and implemented two new algorithms; the first one is based on the computation of a set of geometrical features, such as nose width and length, mouth position, and chin shape, and the second one is based on almost-grey-level template matching. The results obtained on the testing sets (about 90% correct recognition using geometrical features and perfect recognition using template matching) favour our implementation of the templatematching approach

## 2.4 Face Recognition using Unsupervised Feature Learning Approach

It is a face recognition system that uses joint feature learning that helps us learn feature representation directly from raw pixels. Unsupervised feature learning enables us to recognize faces even in unconstrained environment like varying poses and expressions. Firstly, they input an image or video into our system. Then They pre-process it by converting it into a gray-scale image. This is to reduce the complexity of the computation. Then we apply Face Detection algorithm (Viola Jones Algorithm) to detect the faces in the input image. After this process, They got all the faces in the image along with the count of it. Then adopt Gradient Boost Algorithm (Feature Extraction) to extract the features from the faces obtained in the previous step. This is followed by Correlation matching in which, the features obtained are matched with a template of the face to be recognized. If it matches, it signifies that the face is available in the input frame. Or else the face to be recognized is not available in the given image. The final step is using Support Vector Machines classifiers that are used for the actual authentication process. If this step is complete, the recognized face is given as output. Here, the usage of Viola

Jones Algorithm has shown an overall increase in accuracy and reduction in computation time. You can refer to paper[2] in bibliography.

## 2.5 Understanding face recognition

A functional model [3] is proposed in which structural encoding processes provide descriptions suitable for the analysis of facial speech, for analysis of expression and for face recognition units. Recognition of familiar faces involves a match between the products of structural encoding and previously stored structural codes describing the appearance of familiar faces, held in face recognition units. Identity-specific semantic codes are then accessed from person identity nodes, and subsequently name codes are retrieved. It is also proposed that the cognitive system plays an active role in deciding whether or not the initial match is sufficiently close to indicate true recognition or merely a 'resemblance'; several factors are seen as influencing such decisions. This functional model is used to draw together data from diverse sources including laboratory experiments, studies of everyday errors, and studies of patients with different types of cerebral injury. It is also used to clarify similarities and differences between processes responsible for object, word and face recognition

# Chapter 3

# Algorithms

## 3.1 Introduction

In this chapter we will understand the algorithms used in making this project.We will start with viola jones algorithm for detecting the face. Although this algorithm was given for object detection by Paul viola and Michael jones in their research paper ”Rapid object detection using a boosted cascade of simple features”.After that we will understand LBPH(local binary pattern histogram) algorithm for recognizing the new person.

## 3.2 Viola Jones algorithm for detecting face

The Viola–Jones algorithm is the first object detection approch to provide object detection rates in real-time proposed in 2001 by Paul Viola and Michael Jones. Although it can be trained to detect a variety of object classes, it was primarily used for face detection.

The characteristics of Viola–Jones algorithm which make it a good detection algorithm are:
1)Robust – very high detection rate (true-positive rate)  very low false-positive rate always.
2)Real time – For practical applications at least 2 frames per second must be processed.
3)Face detection only (not recognition) - The goal is to distinguish faces from non-faces (detection is the first step in the recognition process).

The algorithm has four stages:
1)Haar Feature Selection
2)Creating an Integral Image
3)Adaboost Training
4)Cascading Classifiers

But before extracting features we convert the image into greyscale.The reason behind this step is that we don't need any colour data to detect face. Also all color image have 3 channels of RGB. These are 3 matrix each for red,green and blue. While greyscale image have only 1 channel(1 matrix to represent image).So greyscale image is more easier to playwith as compared to colored image.

### 3.2.1 Haar Feature

All human faces share some similar properties. These may be matched using Haar Features.

A few properties common to human faces:
1)The eye region is darker than the upper-cheeks.
2)The nose bridge region is brighter than the eyes.
3)Location and size: eyes, mouth, bridge of nose.

Haar features is basically a convolutional kernel .The fig 3.1 shows the four basic haar features and extended features. In these feature all the black pixels are given high value

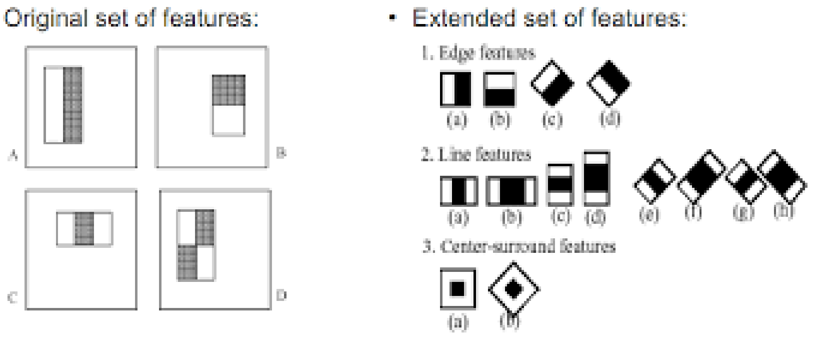

• Original set of features:
A
B
C
D
• Extended set of features:
1. Edge features
(a) (b) (c) (d)
2. Line features
(a) (b) (c) (d) (e) (f) (g) (h)
3. Center-surround features
(a) (b)

Figure 3.1: Basic haar features(left) and extended features(right)

and all white pixel are given low value. These features are then applied on our image to extract features by simply subtracting the sum of all white pixel from the sum of all the black pixel.Now, all possible sizes and locations of each kernel are used to calculate lots of features. (Just imagine how much computation it needs? Even a 24x24 window results over 160000 features). For each feature calculation, we need to find the sum of the pixels under white and black rectangles. To solve this, they introduced the integral image. However large your image, it reduces the calculations for a given pixel to an operation involving just four pixels. It makes things fast.

But among all these features we calculated, most of them are irrelevant. For example, consider the fig 3.2. The top row shows two good features. The first feature selected seems to focus on the property that the region of the eyes is often darker than the region of the nose and cheeks. The second feature selected relies on the property that the eyes are darker than the bridge of the nose. But the same windows applied to cheeks or any other place is not a good idea. So our aim is to select only proper features from 160000+ features. It is achieved by Adaboost.

### 3.2.2 Adaboost

AdaBoost, short for Adaptive Boosting, is a machine learning meta-algorithm formulated by Yoav Freund and Robert Schapire.For detection we apply each and every feature on all the training images. For each feature, it finds the best threshold which will classify the faces to positive and negative. Obviously, there will be errors. We select the features with minimum error rate, which means they are the features that most accurately classify

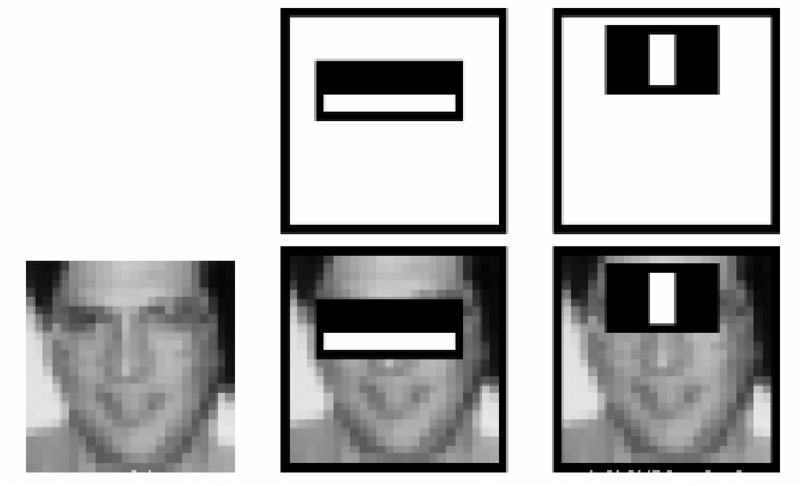


Figure 3.2: features on human face

the face and non-face images. (The process is not as simple as this. Each image is given an equal weight in the beginning. After each classification, weights of misclassified images are increased. Then the same process is done. New error rates are calculated. Also new weights. The process is continued until the required accuracy or error rate is achieved or the required number of features are found).

The final classifier is a weighted sum of these weak classifiers. It is called weak because it alone can't classify the image, but together with others forms a strong classifier. The paper says even 200 features provide detection with 95 percent accuracy. Their final setup had around 6000 features.

### 3.2.3 Cascading Classifiers

So now you take an image. Take each 24x24 window. Apply 6000 features to it. Check if it is face or not but still it is very time consuming process. To solve this we use Cascading Classifier.

In an image, most of the image is non-face region. So it is a better idea to have a simple method to check if a window is not a face region. If it is not, discard it and don't process it again. Instead, focus on regions where there can be a face. This way, we spend more time checking possible face regions.

For this viola-jones[4] introduced the concept of Cascade of Classifiers. Instead of applying all 6000 features on a window, the features are grouped into different stages of classifiers and applied one-by-one. (Normally the first few stages will contain very many fewer features). If a window fails the first stage, discard it. We don't consider the remaining features on it. If it passes, apply the second stage of features and continue the process. The window which passes all stages is a face region.

According to viola-jones their detector had 6000+ features with 38 stages with 1, 10, 25, 25 and 50 features in the first five stages. (The two features in the above image are actually obtained as the best two features from Adaboost). According to the authors, on average 10 features out of 6000+ are evaluated per sub-window.

So this is a simple intuitive explanation of how Viola-Jones face detection works.

## 3.3 LBPH(Local Binary Pattern Histogram) for face recognition

the Local Binary Patterns methodology has its roots in 2D texture analysis. The basic idea of Local Binary Patterns is to summarize the local structure in an image by comparing each pixel with its neighborhood. Take a pixel as center and threshold its neighbors against. If the intensity of the center pixel is greater-equal its neighbor, then denote it with 1 and 0 if not. You'll end up with a binary number for each pixel, just like 11001111.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.So with 8 surrounding pixels you'll end up with 2 power 8 possible combinations, called Local Binary Patterns or sometimes referred to as LBP codes. The first LBP operator described in literature actually used a fixed 3 x 3 neighborhood just like fig 3.5: At the end of this

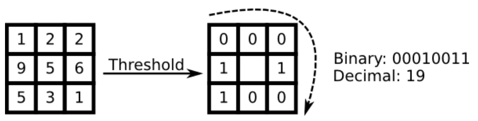


Figure 3.3: LBPH procedure

procedure (LBP procedure), we have a new image which represents better the characteristics of the original image.Now after doing lbp operation each region of the face is converted into histogram and we get a set of histogram. Now to recognize the face we compare the new face histogram with the histogram of faces from dataset.

## 3.4 Summary

So, in this chapter we learn about various algorithm we have used.First we understand Viola Jones algorithm where we learned the concept of haar features, integral images, adaboost and cascade classifier. Then we learned the concept behind lbph algorithm in facial recognition which is based on creating histogram of each region of face and then coparing that histogram set with our the histogram set of our image dataset .

# Chapter 4

# Coding and Implementation

The final part of this project is to apply all these algorithm to achieve our goal. For our coding part we will use opencv library which is widely used in computer vision. This library have lots of pre trained classifier which decrease labour and time in implementing these algorithm. Doing everything from scratch is very tedious.We will use python as our programming language.

Caution: Codes are not arranged properly only some part of code is explained.

## 4.1 code for detecting face

The first goal is to capture your video from web camera and then detecting face from it.
So we will import opencv library.
Code for capturing video and detecting face

```
video = cv2.VideoCapture(0)
face_cascade = cv2.CascadeClassifier('libraries\cv2\data\haarcascade_frontalface_alt2.xml')
while True:
    check,frame = video.read()
    #converting the frame to greyscale and detecting face
    gray = cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray,scaleFactor=1.6,minNeighbors=8)
    for (x,y,w,h) in faces:
        roi_gray = gray[y:y+h , x:x+w]
        roi_color = frame[y:y+h , x:x+w]
        rec_color = (255,0,0)
        rec_width = 2
        rec = cv2.rectangle(frame,(x,y),((x+w),(y+h)),rec_color,rec_width)
        cv2.imshow("capture",frame)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break;
video.release()
cv2.destroyAllWindows()
```

### 4.1.1 Explaination

So first we are capturing video frame by frame by our web camera using opencv VideCapture(0) and read() function. We will save it in a file. 0 indicate the web cam. we can write 1 or path of a video file to read any external camera or video file respectively.After that we make a face cascade and used pre trained model of viola-jones face detector.This training data can be found in the directory /data of opencv library as lots of xml file. We are loading haarcascade_frontalface_alt2.xml file to extract haar feature from the face . We can use other files to detect different part as per requirement. Now we will convert the each frame into greysacle using cvtColor() function. Then we will detect the face using detectMultiScale() function. This detectMultiScale function have two important parameter to understand. First one is scaleFactor. Basically the scale factor is used to create your scale pyramid. In short. Your model has a fixed size defined during training. This means that this size of face is detected in the image if occuring. However, by rescaling the input image, you can resize a larger face towards a smaller one, making it detectable for the algorithm.Using a small step for resizing, for example 1.05 which means you reduce size by 5%, you increase the chance of a matching size with the model for detection is found.

The second parameter is minNeighbour. Haar cascade classifier works with a sliding window approach. If you look at the cascade files you can see a size parameter which usually a pretty small value like 20 20. This is the smallest window that cascade can detect. So by applying a sliding window approach, you slide a window through out the picture than you resize it and search again until you can not resize it further. So with every iteration haar's cascaded classifier true outputs are stored. So when this window is slided in picture resized and slided again; it actually detects many many false positives. So, it reduce any false face detection and increase accuracy.

Now we have detected the face , so we will draw rectangle over the face. This is done by rectangle() function where we can give our color and witdh of our rectangle. Finally we will use imshow() function to show the captured frame. We will use "q" button to stop the capturing.

## 4.2 code for training image dataset

Now we will make a dataset of person to be recognized. For this we will click almost 10 picture of the person and place them under a folder image_name. During training the face data from images we will do labelling to each data. This label will apear on the screen when the person will be recognized. We will use the person_name for labelling. All these will be done in another python file.

#we will extract the path of image directory.

```
file_dir = os.path.dirname(os.path.abspath(___file___))
image_dir = os.path.join(file_dir,"images")


#Again we will load haar cascade to detect face from the image dataset
face_cascade = cv2.CascadeClassifier('libraries\cv2\data\haarcascade_frontalface_alt2.xml')

# we will create LBPH recognizer for training the face data.
recognizer = cv2.face.LBPHFaceRecognizer_create()

# We will label the image data.
for root , dirs , files in os.walk(image_dir):
        for file in files:
                path = os.path.join(root,file)
                label = os.path.basename(root)
                 if label not in label_id:
                        label_id[label] = current_id
                        current_id = current_id + 1
                id_for_label = label_id[label]

# Next we will be extracting whole image info and converting it in numpy array.
pil_image = Image.open(path).convert("L")
size = (500,550)
final_image = pil_image.resize(size , Image.ANTIALIAS)
image_array = np.array(final_image , "uint8")

#extracting region of intrest from the images and appending for training purpose
faces = face_cascade.detectMultiScale(image_array, scaleFactor = 1.6 ,minNeighbors=3)
for (x,y,w,h) in faces:
        roi = image_array[y:y+h , x:x+w]
        x_train.append(roi)
        y_labels.append(id_for_label)

# Now we will dump the labels data into labels.pkl file
with open("labels.pkl" , "wb") as file:
        pkl.dump(label_id,file)
```

# Next we will train the face data using LBPHFACERECOGNIZER and save it as yaml file.
recognizer.train(x_train , np.array(y_labels))
recognizer.save(”trainer.yml”)

So, we are simply extracting face from image dataset ,assign label to the data and train the data using train() function.

## 4.3 Code for recognizing

Now we will recognize the new face. To do so we will again open the file in which we have code to capture video and detecting face. Their we will create the LBPH recognizer and then will load the YML file which we have saved during training. To do so we will code
recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer.read(”trainer.yml”)

Now we will recognize the person using predict() function.
id_for_label ,conf = recognizer.predict(roi_gray)

Now we will write the label associated with the person using putText() function.
font = cv2.FONT_HERSHEY_SIMPLEX
name = labels[id_for_label]
color = (255,255,255,0)
stroke = 2
cv2.putText(frame , name, (x,y), font , 1 ,color,stroke,cv2.LINE_AA)

So, after coding we will run both the files(training file and detecting file). Training will take some time as per the size of your image dataset. Running the first file(having detection code) we can test the system.

# Chapter 5

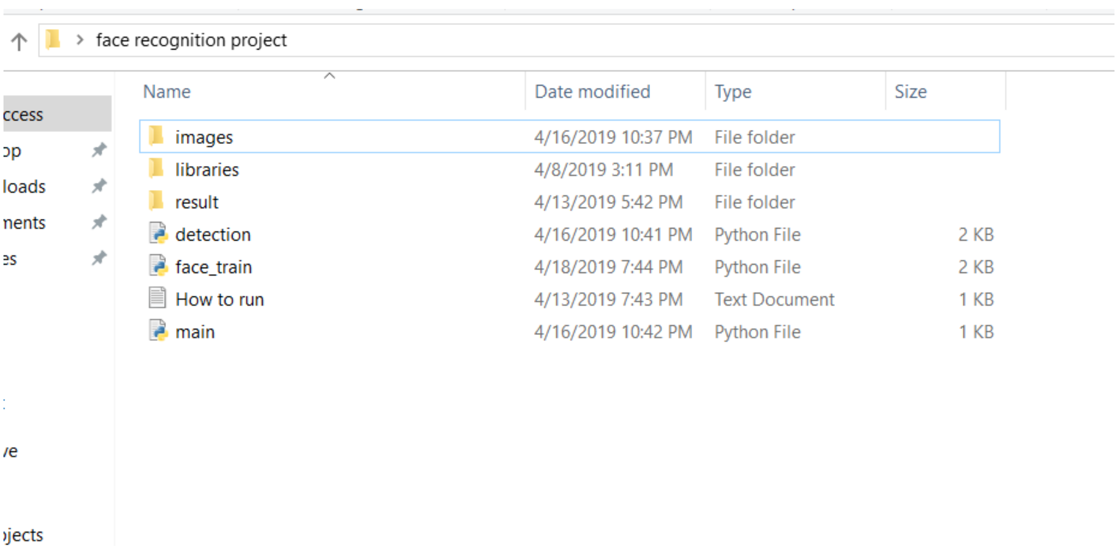
# Conclusion

This is a very basic face recognition system with not very good accuracy. That's why people moved to deep learning and convolutional neural network model. The main reason behind poor accuracy may be the quality of image dataset we have provided. It can be improved using some image processing technique for better extraction of data from image. Also during implementation I realise that the image data taken from high quality camera gives us better result. Like the images taken of Naman in the dataset is of poor quality. Also the orientation of his front face is not proper in his images. So our system find difficulties in recognizing Naman.Another problem is with the training process. The time taken for around 50 image data is almost 1 min. which is very high. So, if we will work with data of many people, the system become slower. Another problem is the false negative results. It also recognize a person not in dataset.

# Appendix A

# Screenshot and Description of the Implemented System

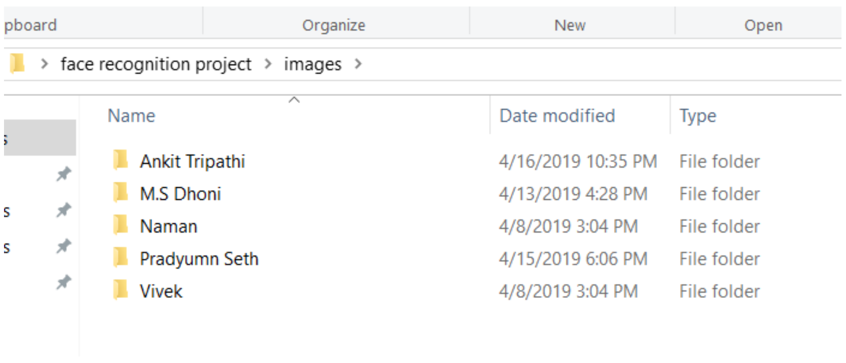
## A.1 File structure



| Name | Date modified | Type | Size |
|---|---|---|---|
| images | 4/16/2019 10:37 PM | File folder | |
| libraries | 4/8/2019 3:11 PM | File folder | |
| result | 4/13/2019 5:42 PM | File folder | |
| detection | 4/16/2019 10:41 PM | Python File | 2 KB |
| face_train | 4/18/2019 7:44 PM | Python File | 2 KB |
| How to run | 4/13/2019 7:43 PM | Text Document | 1 KB |
| main | 4/16/2019 10:42 PM | Python File | 1 KB |

Figure A.1: file structure of project

## A.2 Image DataSet



| Name | Date modified | Type |
|---|---|---|
| Ankit Tripathi | 4/16/2019 10:35 PM | File folder |
| M.S Dhoni | 4/13/2019 4:28 PM | File folder |
| Naman | 4/8/2019 3:04 PM | File folder |
| Pradyumn Seth | 4/15/2019 6:06 PM | File folder |
| Vivek | 4/8/2019 3:04 PM | File folder |

Figure A.2: Image dataset

## A.3 Some output results

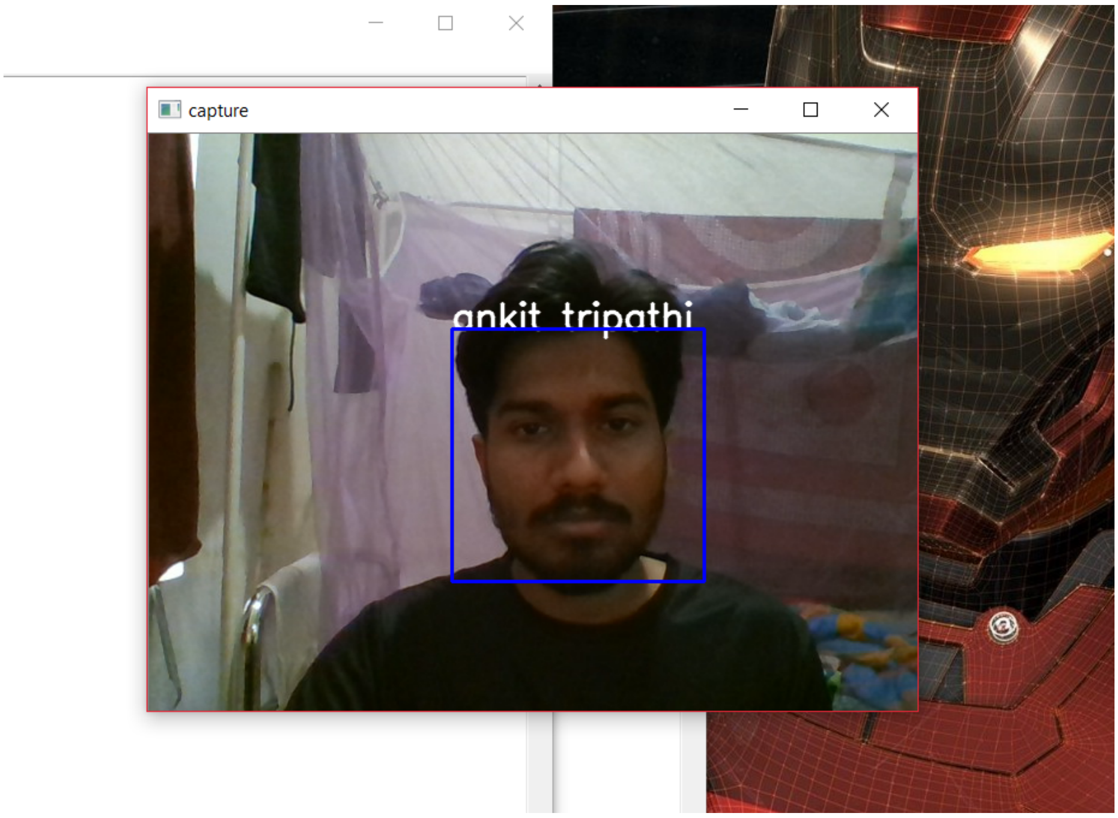


Figure A.3: result 1

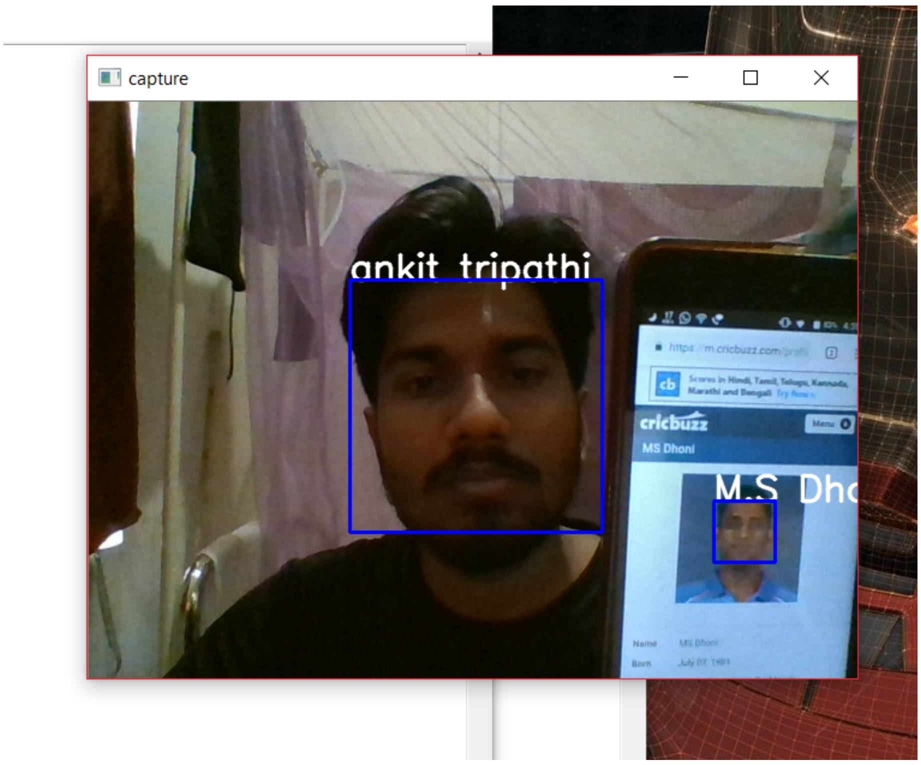


Figure A.4: result 2

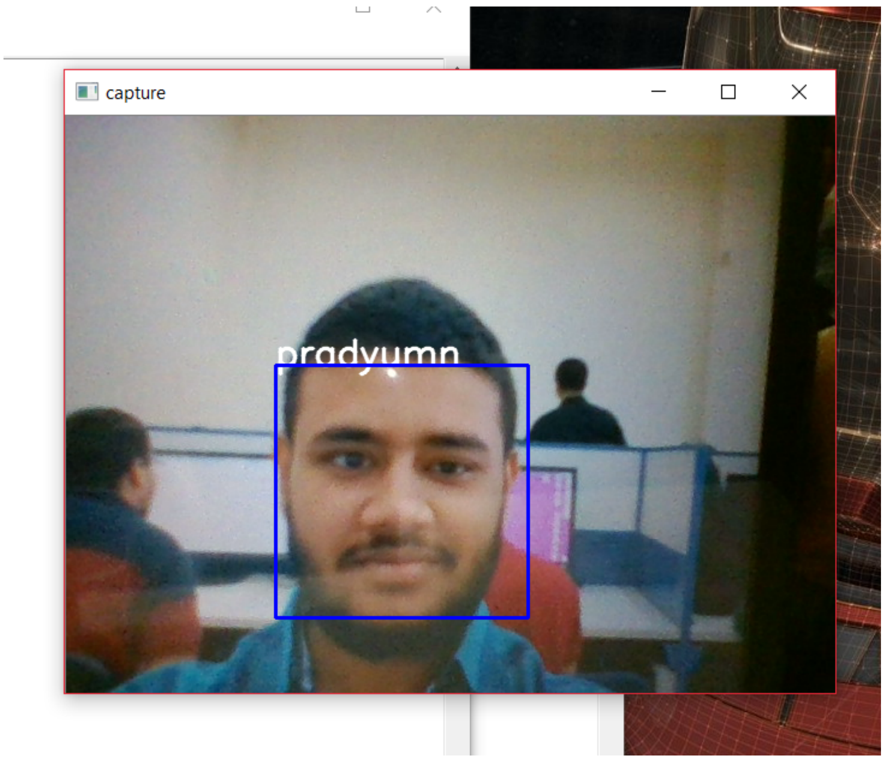


Figure A.5: result 3

# Appendix B

# User manual

## B.1 Introduction

Description of the system.

## B.2 Step to run your implemented system

First install the required libraries using pip command.pip install opencv-contrib-python
pip install numpy

Making Your own Image Data set-

Click atlest 10 picture of the person you want to recognize.
Inside image directory, make a new directory of the person's name.
Run main.py file.
After running the main.py your web camera will open and will recognize the person front of it.

# Bibliography

[1]  Face recognition: features versus templates,  in IEEE Transactions on Pattern Analysis and Machine Intelligence.

[2]  Face Recognition using Unsupervised Feature Learning (UFL) Approach,  M. Therasa, S.M. Poonkuzhali 242, 2016

[3]  Understanding face recognition,  Vicki Bruce Andy Young.

[4]  Rapid Object Detection using a Boosted Cascade of Simple Features