

# **Face Recognition System – LBPH Algorithm**

Submitted in partial fulfilment of the requirements for the award of Bachelor of  
Technology Degree in Information Technology

by

**BALABHARATHI P (38120017)**

**YUVASRI J (38120097)**



**DEPARTMENT OF INFORMATION TECHNOLOGY  
SCHOOL OF COMPUTING**

**SATHYABAMA**

**INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)  
Accredited with Grade “A” by NAAC**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600 119**

**MAY 2022**



# **SATHYABAMA**

INSTITUTE OF SCIENCE AND TECHNOLOGY

**(DEEMED TO BE UNIVERSITY)**

Accredited with “A” grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

[www.sathyabama.ac.in](http://www.sathyabama.ac.in)



## **DEPARTMENT OF INFORMATION TECHNOLOGY**

### **BONAFIDE CERTIFICATE**

This is to certify that this Project Report is the bonafide work of **BALABHARATHI P (38120017)** and **YUVASRI J (38120097)** who carried out the project entitled “**FACE RECOGNITION SYSTEM-LBPH ALGORITHM**” under our supervision from November 2021 to April 2022.

#### **Internal Guide**

Dr.V.Nirmalrani, 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**

## **ACKNOWLEDGEMENT**

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** 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,** and **Dr.R.Subhashini M.E., Ph.D., Head of the Department, Department of Information Technology** for providing us 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.V.Nirmalrani.M.E.,Ph.D.,** for her 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.

## **DECLARATION**

I, **YUVASRI J (Reg no:38120097)** hereby declare that the Project Report entitled “**FACE RECOGNITION SYSTEM - LBPH ALGORITHM**” done by me under the guidance of **Dr.V.Nirmalrani, M.E., (Ph.D)** at “Sathyabama Institute of Science and Technology (Deemed to be University), Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600119” is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in INFORMATION TECHNOLOGY.

**DATE:**

**PLACE: CHENNAI**

**SIGNATURE OF THECANDIDATE**

## **ABSTRACT**

Face Recognition is a computer application that is capable of detecting, tracking, identifying or verifying human faces from an image or video captured using a digital camera. Although lot of progress has been made in domain of face detection and recognition for security, identification and attendance purpose, but still there are issues hindering the progress to reach or surpass human level accuracy. These issues are variations in human facial appearance such as; varying lighting condition, noise in face images, scale, pose etc. This research paper presents a new method using Local Binary Pattern (LBP) algorithm to improve the LBP codes, thus improve the accuracy of the overall face recognition system. In this paper we have proposed a system that is using Local Binary Patterns Histogram( LBPH) algorithm for identifying a face. It can recognize both front and side faces and upgrade the value of poor enlightened picture and also expands the recognition rate in real time.

## TABLE OF CONTENT

| CHAPTER NO | TITLE NAME                             | PAGE NO |
|------------|----------------------------------------|---------|
|            | ABSTRACT                               | I       |
|            | TABLE OF CONTENTS                      | II      |
|            | LIST OF ABBREVIATION                   | IV      |
|            | LIST OF FIGURES                        | V       |
| <b>1</b>   | <b>INTRODUCTION</b>                    |         |
|            | 1.1 GENERAL                            | 1       |
|            | 1.2 MOTIVATION                         | 1       |
|            | 1.3 OUTLINE OF THE PROJECT             | 2       |
|            | 1.4 PROBLEM STATEMENT                  | 2       |
|            | 1.5 OBJECTIVE OF THE PROJECT           | 2       |
| <b>2</b>   | <b>AIM AND SCOPE</b>                   |         |
|            | 2.1 AIM                                | 3       |
|            | 2.2 PROBLEM DEFINITATION               | 3       |
|            | 2.3 LITERATURE REVIEW                  | 3       |
|            | 2.4 SCOPE                              | 4       |
| <b>3</b>   | <b>ALGORITHM AND METHODS</b>           |         |
|            | 3.1 ALGORITHM                          | 5       |
|            | 3.2 EXISTING SYSTEM                    | 5       |
|            | 3.3 PROPOSED SYSTEM                    | 5       |
|            | 3.3.1 ADVANTAGES OF PROPOSED<br>SYSTEM | 5       |
|            | 3.4 MODULES                            | 6       |
|            | 3.4.1 FACE DETECTION                   | 6       |
|            | 3.4.2 FEATURE EXTRACTION               | 6       |
|            | 3.4.3 DATASET                          | 7       |
|            | 3.4.3.1 FACE RECOGNITION               | 7       |

|          |                                       |           |
|----------|---------------------------------------|-----------|
| <b>4</b> | <b>SYSTEM REQUIREMENTS AND DESIGN</b> |           |
|          | 4.1 REQUIREMENTS                      | <b>8</b>  |
|          | 4.1.1 HARDWARE REQUIREMENTS           | <b>8</b>  |
|          | 4.1.2 SOFTWARE REQUIREMENTS           | <b>8</b>  |
|          | 4.2 SOFTWARE DESCRIPTION              | <b>8</b>  |
|          | ROLE OF PYTHON                        | <b>8</b>  |
|          | NUMPY                                 | <b>9</b>  |
|          | FACE-RECOGNITION                      | <b>9</b>  |
|          | 4.3 DESIGN                            | <b>9</b>  |
|          | 4.3.1 OPEN CV                         | <b>9</b>  |
|          | 4.3.2 MACHINE LEARNING                | <b>10</b> |
|          | 4.3.3 IMAGE PROCESSING                | <b>10</b> |
|          | 4.4 SYSTEM ARCHITECTURE               | <b>14</b> |
|          | 4.4.1 USE CASE DIAGRAM                | <b>15</b> |
| <br>     |                                       |           |
| <b>5</b> | <b>RESULTS</b>                        |           |
|          | 5.1 SOFTWARE SETUP                    | <b>16</b> |
|          | 5.2 EXPERIMENTAL RESULTS              | <b>16</b> |

|          |                        |           |
|----------|------------------------|-----------|
| <b>6</b> | <b>CONCLUSION</b>      |           |
|          | <b>6.1 CONCLUSION</b>  | <b>17</b> |
|          | <b>6.2 FUTURE WORK</b> | <b>18</b> |
|          | <b>REFERENCE</b>       | <b>18</b> |
|          | <b>APPENDIX</b>        | <b>19</b> |
|          | <b>SOURCE CODE</b>     | <b>19</b> |

## **LIST OF ABBREVIATION**

- |    |      |                                |
|----|------|--------------------------------|
| 1. | LBPH | Local Binary Pattern Histogram |
| 2. | LBP  | Local Binary Pattern           |

## LIST OF FIGURE

| FIGURE NO | TITLE OF THE FIGURE  | PAGE NO |
|-----------|----------------------|---------|
| 3.1       | Feature Extraction   | 7       |
| 4.1       | python               | 8       |
| 4.2       | opencv               | 10      |
| 4.3       | Architecture diagram | 14      |
| 4.4       | use case diagram     | 15      |
| 5.1       | sample output 1      | 16      |
| 5.2       | sample output 2      | 17      |

# CHAPTER 1

## INTRODUCTION

### 1.1 GENERAL

Automatic face identification is an essential theme of computer vision and pattern recognition Analysis. It will distinguish the personalities and distinct data as indicated by the optical highlights of face picture, having an exceptionally wide possibility for advancement. It is typically used in validation, Criminal examination, police investigation, automaton intelligence, and healthful science, etc. There are various techniques utilized in face recognition system. Each distinctive properties underneath varied conditions like illumination, expression and posture amendment. Among them for our motivation that is confront discovery in the Video stream, we tend to focus on three noteworthy methods and that we looked their execution underneath modified improvement condition to find out an honest coordinating and low process expenses. The contributions of this work tend to develop a real time face-recognition system that will acknowledge face in poor illumination condition and fetch information about that image from the databases.

### 1.2 MOTIVATION

- The face detection Very few biometric technologies are attracting as much attention as facial recognition. Facial recognition technology offers a host of benefits such as in authentication, monitoring, access control, indexing, and maintenance of surveillance applications.
- Face recognition systems have led to the advancement of multimedia information access. Also, implementing network access control via face recognition not only makes it virtually impossible for hackers to steal a user's password but also improves human-computer interaction. This is one of the reasons why facial recognition attendance systems are gaining popularity.

- It is typically utilized in validation, criminal examination, police investigation, automation intelligence, and healthful science, etc.
- To detect individuals by using system

### 1.3 OUTLINE OF THE PROJECT

The project is all about detecting face for security system. This project helps the user to avoid insecurity. We are using Local Binary Pattern Histogram algorithm for face recognition. User need to store photos with name in database. When we log in with this system it shows our name in screen. In this case, the unknown log in with this system it shows warning signal. We are using this application in security system.

### 1.4 PROBLEM STATEMENT

The main problem of face recognition is its **high dimension space**, which is to be reduced by any dimension reduction techniques. The pattern recognition approach then tries to match the facial features, which are extracted from all the images present in the database.

### 1.5 OBJECTIVE OF THE PROJECT

The objective of face recognition is, from the incoming image, **to find a series of data of the same face in a set of training images in a database**. The great difficulty is ensuring that this process is carried out in real-time, something that is not available to all biometric facial recognition software providers.

## **CHAPTER 2**

### **AIM AND SCOPE**

#### **2.1 AIM**

The aim of our project is to design a system that will detect face and take necessary steps to avoid insecurity. The Face Recognition by using LBPH algorithm, which is been implemented in this project aims that it can be easily available and can be used with different types of organisations.

#### **2.2 PROBLEM DEFINITION**

Face recognition has many challenges due to illumination variations, large dimensionality, uncontrolled environments, pose variations and aging. In the recent years, Face recognition get remarkable improvement and accuracy to overcome these challenges, but illumination change is still changing.

#### **2.3 LITERATURE REVIEW**

According to the Survey on Face Recognition by using LBPH algorithm, the detection system includes the processes of face image extraction, face detection ,etc.

Ojala et al.[1] initially used Local Binary Pattern (LBP). LBP refers unremarkably to displacement pixels with an 8-bit dual code from the neighborhood of the pixels. For a 3 square, the center component estimate is ablated from that of each of its eight neighboring pixels and is relegated to a bit based on indication of the subtraction result One otherwise zero. The created bits are then connected for all neighboring pixels and encoded into double strings. The dual stringsinferred are referred to as Local Binary Patterns or LBP codes. The decimal estimate of the center component's LBP code (xc,yc). The LBP administrator  $LBP(P,R)$  produces  $2^p$  different yield values, about  $2^p$  distinctive two-fold examples formed by the P pixels within the R-space.

Ahonen et al. [2] presented an LBP primarily based facial illustration. They have bifurcated facial images into  $M$  minor non-covering areas  $R(0), R(1), \dots, R(M)$  to compare the facial structure. The LBP histograms extracted from each sub-region are then combined into a single, spatially enhanced histogram features.

In [3] as suggested by Ahonen et al, the texture classification LBP approach describes the occurrence of LBP codes in an image being collected into a histogram. The classification is then carried out by calculating the similarities between straight-forward histograms. However, a loss of spatial data results in considering the same methodology for illustration of the facial image. Bruce et al. [4] represented studies that shows distinctive faces are better retained in memory and are recognized better and faster than typical faces. However, if a decision has to be made as to whether an object is a face or not, it takes longer to recognize an a typical face than a typical face. This may be explained by different mechanisms being used for detection and for identification. Bruce [5] represents Facial expressions based on neurophysiological studies, it seems that analysis of facial expressions is accomplished in parallel to face-recognition. Some prosopagnosic patients who have difficulties in identifying familiar faces, nevertheless seem to recognize expression due to emotions. patients who suffers from "Organic brain syndrome" suffers from poor.

## **2.4 SCOPE**

- To automatically recognize the face of the user and its detection.
- To analyze the focus of driver while driving.
- To protect secured data or organisation.
- To detect whether the camera is focusing properly and the user's face.
- To verify the duplicate user.

## **CHAPTER 3**

### **ALGORITHM AND METHODS**

#### **3.1 ALGORITHM**

The overall algorithm is pretty straightforward one. First we have used a camera to capture the faces of humans and it will be stored in the dataset.

If face gets detected, the system checks whether the face is similar to the images which has been stored in the dataset. If yes, it will recognize the people else it will show the entered person is unknown.

For the functionalities of the system and to make it work efficiently we have used OpenCv and Python.

The implementation of the face recognition system includes machine learning algorithms which are in turn included in OpenCv ML algorithms. There are numerous ML algorithms but for our purpose we required only the LBPH algorithm.

#### **3.2 EXISTING SYSTEM**

Face recognition must be fast .However, performing facial good result in low -light conditions poses numerous challenges. We have suggested a system for detecting a face that uses the Local Binary Patterns Histogram technique in this project. It can distinguish full face , and it can improve a dimly lit image while simultaneously increasing the recognition rate in real time.

#### **3.3 PROPOSED SYSTEM**

##### ***3.3.1 ADVANTAGES OF PROPOSED SYSTEM***

- Enhanced Security
- Fast and Non-Invasive Identity Verification
- Reduces the number of touchpoints
- Improves photo organization
- Helps find missing people

### 3.4 MODULES

- Face detection
- Feature extraction
- Dataset
- Face recognition

#### 3.4.1 Face Detection

For face detection, OpenCV is used that introduces a Haar cascade classifier . Initially, it takes an input image using the camera and it will determines whether the image contains any faces in the frame or not. When any face is detected different facial features are checked and a square frame is drawn on the face. Otherwise, it starts reading other pictures.

#### 3.4.2 Feature Extraction

Extracting the facial features from Image the LBP operation is used that compares the intensity value of every component with the 8 nearest neighbor pixel values .If the value of the neighboring pixel is more than the value of the centered pixel, it will assign 1 to its neighboring pixel, otherwise it will assign 0. Each pixel this task provides an 8 bit string .A decimal value of an 8-bit pixel string determines the LBP value

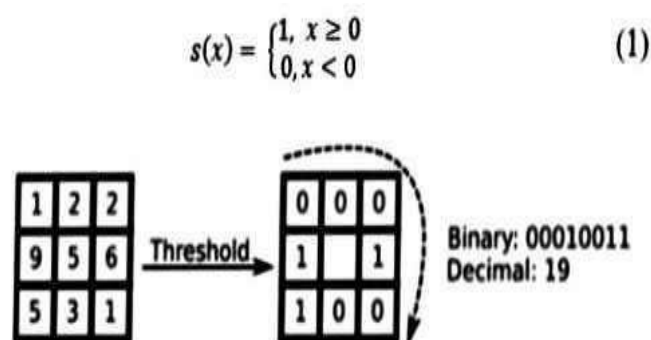


Fig.3.1 Extraction of LBP feature from the image

### **3.4.3 Dataset**

We designed our own dataset, with individual photos. Then those images are saved in the same folder to make face databases for extraction tasks.

### **3.4.4 Face Recognition**

For the face recognition process, Local Binary Pattern Histogram(LBPH) algorithm is applied. The LBP operator uses Local Binary Patterns to decrease the local spatial distribution of a face image. The LBP operator is a collection of binary pixel value ratios in the center at regular pixel intervals and is around eight pixels.

$$LBP(x_c, y_c) = \sum_{n=0}^7 S(i_n - i_c) 2^n$$

## **CHAPTER 4**

### **SYSTEM REQUIREMENT AND DESIGN**

#### **4.1 REQUIREMENT**

##### **4.1.1 HARDWARE REQUIREMENT**

- A system with minimum 8GB RAM.
- Camera

##### **4.1.2 SOFTWARE REQUIREMENT**

- Python version 3.8 installed in the system.
- Using pip command
  - Numpy
  - Opencv

#### **4.2 SOFTWARE DESCRIPTION**

##### **ROLE OF PYTHON**

Python is an interpreted, high-level, general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed AND supports multiple programming paradigms, including procedural, object-oriented, and functional programming.



Fig 4.1:Python

## NUMPY

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices.

## FACE RECOGNITION

Face-Recognition in Python is an open-source library used for **Automatically find all the faces in an image**. Automatically locate the facial features of a person in an image. Recognize faces in images and identify who they are.

## 4.3 DESIGN

### 4.3.1 OPEN CV

OpenCV is an open source computer vision library accessible in python coding language to code for visionary capabilities of our smart pc. OpenCV was expected for computational capability and having a high focus on ongoing picture location and distinguishing proof. OpenCV is coded with streamlined C and can take work with multicore processors. If we need progressively programmed improvement utilizing Intel models [Intel], you can purchase Intel's Integrated Performance Primitives (IPP) libraries [IPP]. These comprise of low-level schedules in different algorithmic regions which are streamlined. OpenCV consequently utilizes the IPP library, at runtime if that library is introduced.



Fig 4.2: Opencv

#### **4.3.2 IMAGE PROCESSING**

In computer science, digital image processing is the use of computer algorithms to perform image processing on digital images.

#### **4.3.3 MACHINE LEARNING**

Machine learning (ML) is the study of computer algorithms that improve automatically through experience. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. Machine learning involves computers discovering how they can perform tasks without being explicitly programmed to do so. It involves computers learning from data provided so that they carry out certain tasks. For simple tasks assigned to computers, it is possible to program algorithms telling the machine how to execute all steps required to solve the problem at hand; on the computer's part, no learning is needed. For more advanced tasks, it can be challenging for a human to manually create the needed algorithms. In practice, it can turn out to be more effective to help the machine develop its own algorithm, rather than having human programmers specify every needed step.

## CLASSIFIER

Classification is the process of predicting the class of given data points. Classes are sometimes called as targets/ labels or categories. Classification predictive modeling is the task of approximating a mapping function ( $f$ ) from input variables ( $X$ ) to discrete output variables ( $y$ ).

### *K-means Algorithm*

It is an unsupervised clustering algorithm which signifies a distribution of data w.r.t.  $K$  centers,  $K$  being chosen by the coder. The difference between K-means and expectation maximization is that in K-means the centers aren't Gaussian. Also the clusters formed look somewhat like soap bubbles, as centers compete to occupy the closest data points. All these cluster areas are usually used as a form of sparse histogram bin for representing the data.

### *Normal or Naïve Bayes classifier algorithm*

It is a generative classifier where features are often assumed to be of Gaussian distribution and also statistically independent from one another. This assumption is usually false. That's why it's usually known as a —naïve Bayes classifier. That said, this method usually works surprisingly well.

### *Decision trees algorithm*

It is a partially discriminative classifier. The tree we talk about just finds a singular data feature and determines a threshold value of the current node which best divides the data into different classes. The data is broken into parts and the procedure is recursively repeated through the left as well as the right branches of the decision tree. Even if it is not the top performer, it's usually the first thing we try as it is fast and has a very high functionality.

### *Boosting*

It is a discriminative group of classifiers. In boosting, the final classification decision is made by taking into account the combined weighted classification decisions

of the group of classifiers. We learn in training the group of classifiers one after the other. Each classifier present in the group is called a weak classifier. These weak classifiers are usually composed of single-variable decision trees known as —stumps. Learning its classification decisions from the given data and also learning a weight for its vote based on its accuracy on the data are things the decision tree learns during training. While each classifier is trained one after the other, the data points are reweighted to make more attention be paid to the data points in which errors were made. This continues until the net error over the entire data set, obtained from the combined weighted vote of all the decision trees present, falls below a certain threshold. This algorithm is usually effective when a very large quantity of training data is available. [4] *Random trees algorithm*

It is a discriminative woods of a grea t deal of choice trees, every one of which is worked down to a maximal part profundity. At the season of adapting, each hub of each tree is permitted a decision of part factors, however just from an arbitrarily produced subset of the considerable number of information highlights. This makes sure that all the trees become statistically independent and a decision maker. In the run mode, all the trees get an unweighted vote. Random trees are usually quite effective. They can *also perform regression by taking the average of the output numbers from every tree.*

#### *Local Binary Pattern Histogram (LBPH) algorithm*

LBPH (Local Binary Pattern Histogram) is **a Face-Recognition algorithm it is used to recognize the face of a person.** It is known for its performance and how it is able to recognize the face of a person from both front face and side face.This works remarkably well.

#### *Expectation maximization (EM) algorithm*

It is used for clustering. It is a generative unsupervised algorithm It fits N multidimensional Gaussians to the data, N being chosen by the user. It can act as an

efficient way for representing a more complex distribution using only a few parameters (i.e. means and variances). Usually used in segmentation, it can be compared with Kmeans.

### *K-nearest neighbors*

It is one of the simplest discriminative classifier. The training data is simply stored using labels. Then, a test data point is classified in accordance to the majority vote of the K nearest data points. K-nearest neighbours is probably the simplest algorithm we can use. It is usually effective but it can be slow. It also requires a lot of memory.

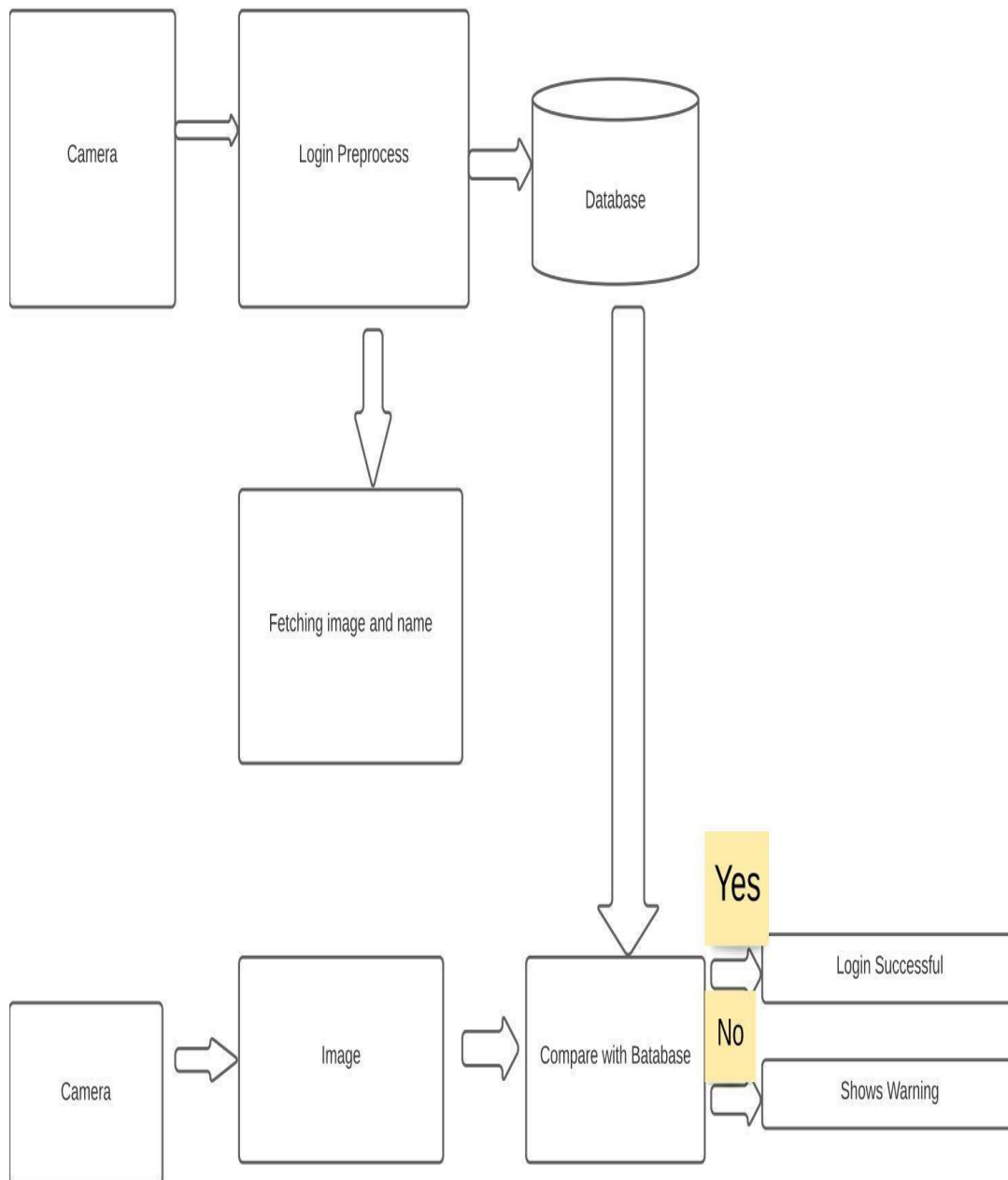
### *Neural networks / Multilayer perceptron (MLP) algorithm*

It is a discriminative algorithm which almost always contains hidden units in between the output and the input nodes for better representation of the input signal. It is slow to train, however it is quite fast to run. It remains the best performer for applications like letter recognition.

### *Support vector machine (SVM) algorithm*

It is a discriminative classifier that is likewise equipped for doing relapse. Here, a separation work in the middle of two information focuses is characterized in a higherdimensional space. (Anticipating information onto higher measurements helps in making the information more probable for direct division.) Support vector machine (SVM) gets the hang of isolating hyperplanes which maximally separate every one of the classes in the higher measurement. This will in general be the best when there is restricted information. Be that as it may, when huge informational indexes are accessible, boosting or arbitrary trees are liked.

## 4.4 SYSTEM ARCHITECTURE



#### 4.4.1 USE CASE DIAGRAM

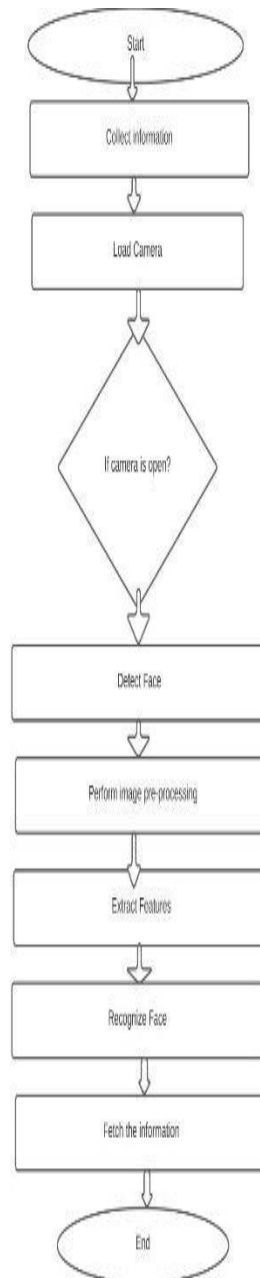


Fig 4.4 use case diagram

## CHAPTER 5

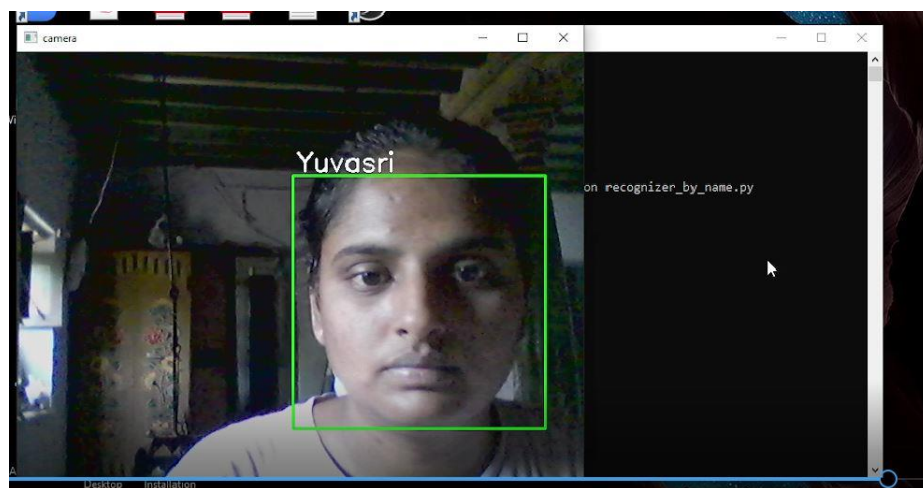
### RESULTS AND DISCUSSION

#### 5.1 SOFTWARE SETUP

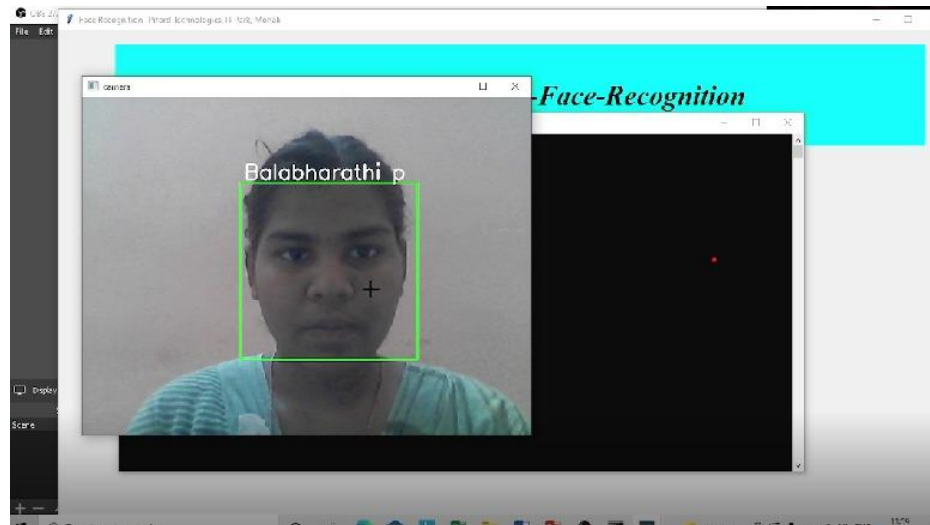
For the execution of the project we need to have a software called Python with version 3.8. After the installation of the software we need to have certain library function such as numpy, pip commands, opencv.

#### 5.2 EXPERIMENTAL RESULTS

To develop a face recognition system, first we have to prepare the face dataset for training. To develop face dataset, we used Face detection method that detects the face in real time camera and captured face images. That captured images are saved into dataset folder for feature extraction and training process. In the initial phase, system will open the camera and that captures images of a person on different face position and expression. The captured images are saved into dataset folder with same unique id. In the next process, the system extracts the LBP texture features from every input dataset image and train them with Haar Cascade classifier. Finally, the system performed face recognition process



Sample output 1



Sample output 2

## CHAPTER 6

## CONCLUSION

### 6.1 CONCLUSION

In the proposed face recognition system, we used Local Binary Patterns histogram algorithm for recognizing faces. The whole procedure is divided into three major components, Detection of faces, facial feature extraction, and classification of the image. The face detection process describes the face of a person in input image. In feature extraction, facial landmarks are extracted to make an LBPH histogram that gives the completely unique result and then in recognition process the histogram of the input image is compared with dataset histogram using the classifier. The result shows that the system can recognize a known and unknown person. This proposed system will assist security agencies in identifying criminals, for attendance purpose, to search any individual in a surveillance camera

## 6.2 FUTURE ENHANCEMENT

If the rate of change in the frame is very high and occlusion occurs, then the system will not robustly recognize faces. This issue has not yet been settled in near future it can be sorted out.

## REFERENCES

1. J. Hai qiang LONG, Tai zhe TAN. Computer Simulation, 2017, 34(1): 322-325.
2. J. HU Liqiao, QIU Runhe, Face recognition on adaptive weighted HOG. Computer Engineering and Applications, 2017, 53(3): 164-168.
3. J. Olshausan B.A, Field D J. Emergence of simple cell receptive field properties by learning a sparse code for natural images. Nature, 1996, 381(6583):607-609.
4. J. WU Qi, WANG Tang, LI Zhan li. Improved face recognition algorithm based on Gabor feature and collaborative representation. Computer Engineering and Design, 2016, 37(10):2769-2774.
5. W J. CHAO L, LIU J Z. Facial expression recognition based on improved Local Binary Pattern and class regularized locality preserving projection. Signal Processing, 2015, 117:1-10.
6. J. Yu Yan JIANG, Qing Wang. Labeled LDA model based on shared background topics.
7. Zheng Xiang, Hengliang Tan, Weinling Ye. The excellent properties of dense gird-based HOG features on face recognition compare to Gaborr and LBP, 2018.
8. Varun Garg, Kritika Garg, Face-recognition using Haar Cascade Classifier, Journal of Emerging Technologies and Innovative Research(JETI),, December 2015.
9. Viola, Paul and Michael J. Jones. "Robust real time Face Detection." International journal of computer vision 57.2(2004):138-154.

## APPENDIX

### SOURCE CODE

```
import numpy as np

import cv2

import os

import faceRecognition as fr

def faceDetection(test_img):

    gray_img=cv2.cvtColor(test_img,cv2.COLOR_BGR2GRAY)

    face_haar=cv2.CascadeClassifier(r'C:\Python37\Projects\models\research\obj
ect_detection\data\haarcascade_frontalface_alt.xml')

    faces=face_haar.detectMultiScale(gray_img,scaleFactor=1,minNeighbors=3)

return faces,gray_img

#Labels for training data has been created

def labels_for_training_data(directory):

    faces=[]

    faceID=[]

    for path,subdirnames,filenames in os.walk(directory):

        for filename in filenames:

            if filename.startswith("."):

                print("skipping system file")

                continue

            id=os.path.basename(path)
```

```

img_path=os.path.join(path,filename)

print ("img_path",img_path)

print("id: ",id)

test_img=cv2.imread(img_path)

if test_img is None:

    print ("Not Loaded Properly")

    continue

faces_rect,gray_img=faceDetection(test_img)

if len(faces_rect)!=1:

    continue

(x,y,w,h)=faces_rect[0]

roi_gray=gray_img[y:y+w,x:x+h]

faces.append(roi_gray)

faceID.append(int(id))

return faces,faceID

#Here training Classifier is called

def train_classifier(faces,faceID):

    face_recognizer=cv2.face.LBPHFaceRecognizer_create()

    face_recognizer.train(faces,np.array(faceID))

    return face_recognizer

#Drawing a Rectangle on the Face Function

def draw_rect(test_img,face):

```

```

(x,y,w,h)=face

cv2.rectangle(test_img,(x,y),(x+w,y+h),(0,255,0),thickness=3)

#Putting text on images

def put_text(test_img,text,x,y):

    cv2.putText(test_img,text,(x,y),cv2.FONT_HERSHEY_DUPLEX,3,(255,0,0),6)

cpt = 0

vidStream = cv2.VideoCapture(0)

while True:

    ret, frame = vidStream.read() # read frame and return code.

    cv2.imshow("test window", frame) # show image in window

    cv2.imwrite("Pycharm/train-images/0/image%04i.jpg" %cpt, frame)

    cpt += 1

    if cv2.waitKey(10)==ord('q'):

        break

test_img=cv2.imread(r'C:\Python37\Projects\Face Recognition\1.jpg') #Give path to
the image which you want to test

faces_detected,gray_img=fr.faceDetection(test_img)

print("face Detected: ",faces_detected)

faces,faceID=fr.labels_for_training_data(r'Give Path Here\train-images') #Give path to
the train-images folder which has both labeled folder as 0 and 1

face_recognizer=fr.train_classifier(faces,faceID)

face_recognizer.save(r'Give Path Here\trainingData.yml') #It will save the trained
model. Just give path to where you want to save

```

name={0:"Ashish",1:"Vijay Deverakonda"} #Change names accordingly. If you want to recognize only one person then write:- name={0:"name"} thats all. Dont write for id number 1.

for face in faces\_detected:

```
(x,y,w,h)=face
```

```
roi_gray=gray_img[y:y+h,x:x+h]
```

```
label,confidence=face_recognizer.predict(roi_gray)
```

```
print ("Confidence :",confidence)
```

```
print("label :",label)
```

```
fr.draw_rect(test_img,face)
```

```
predicted_name=name[label]
```

```
fr.put_text(test_img,predicted_name,x,y)
```

```
resized_img=cv2.resize(test_img,(1000,700))
```

```
cv2.imshow("face detection ", resized_img)
```

```
cv2.waitKey(0)
```

```
cv2.destroyAllWindows
```

```
test_img=cv2.imread(r'Pycharm/projects/2.jpg') #Give path to the image which you want to test
```

```
faces_detected,gray_img=fr.faceDetection(test_img)
```

```
print("face Detected: ",faces_detected)
```

```
face_recognizer=cv2.face.LBPHFaceRecognizer_create()
```

```
face_recognizer.read(r'Pycharm\trainingData.yml') #Give path of where trainingData.yml is saved
```

```
name={0:"Ashish",1:"Vijay Deverakonda"}
```

```

for face in faces_detected:

    (x,y,w,h)=face

    roi_gray=gray_img[y:y+h,x:x+h]

    label,confidence=face_recognizer.predict(roi_gray)

    print ("Confidence :",confidence)

    print("label :",label)

    fr.draw_rect(test_img,face)

    predicted_name=name[label]

    fr.put_text(test_img,predicted_name,x,y)

resized_img=cv2.resize(test_img,(1000,700))

cv2.imshow("face detection ", resized_img)

cv2.waitKey(0)

cv2.destroyAllWindows

face_recognizer=cv2.face.LBPHFaceRecognizer_create()

face_recognizer.read(r'Pycharm\trainingData.yml')

cap=cv2.VideoCapture(0)

name={0:"Ashish",1:"Vijay Deverakonda"}

while True:

    ret,test_img=cap.read()

    faces_detected,gray_img=fr.faceDetection(test_img)

    print("face Detected: ",faces_detected)

    for (x,y,w,h) in faces_detected:

```

```
cv2.rectangle(test_img,(x,y),(x+w,y+h),(0,255,0),thickness=5)
```

```
for face in faces_detected:
```

```
    (x,y,w,h)=face
```

```
    roi_gray=gray_img[y:y+h,x:x+h]
```

```
    label,confidence=face_recognizer.predict(roi_gray)
```

```
    print ("Confidence :",confidence)
```

```
    print("label :",label)
```

```
    fr.draw_rect(test_img,face)
```

```
    predicted_name=name[label]
```

```
    fr.put_text(test_img,predicted_name,x,y)
```

```
resized_img=cv2.resize(test_img,(1000,700))
```

```
cv2.imshow("face detection ", resized_img)
```

```
if cv2.waitKey(10)==ord('q'):
```

```
    break
```