FACE RECOGNITION USING MACHINE LEARNING

Minor Project Report

Submitted in partial fulfilment of the
requirements for the award of the degree

Of

Bachelor of Technology
In

Computer Science and Engineering

BY

Ms. Somya Sharma(Roll No.-8817103008)

Mr. Animesh Kumar Jain(Roll No.-8817103008)

Under the guidance of

Dr. Nishant Srivastava

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

JAYPEE UNIVERSITY ANOOPSHAHR – 223390, UP (INDIA)

NOV, 2019

JAYPEE UNIVERSITY ANOOPSHAHR

CERTIFICATE

This is to certify that the report titled **“Facial Recognition Using Machine Learning”** submitted by **Ms. Somya Sharma** and **Mr. Animesh Kumar Jain** in partial fulfilment of the requirements for the award of Bachelor of Technology degree in **Computer Science and Engineering** during session 2019-2020 at **Japyee University Anoopshahr** is an authentic work by them under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other university / institute for the award of any Degree.

Date:______________________

Dr. Nishant Srivastava

Head of Department

Dept. of Computer Science and Engineering

Japyee University Anoopshahr

____

ACKNOWLEDGEMENT

We would like to express our gratitude to our mentor **Dr. Nishant Srivastava** for his guidance, advice and constant support throughout our project work. We would like to thank him for being our advisor.

Next, we want to express our respects to **Mr. Gaurav Raj, Dr. Nirmal Kumar Gupta and Dr. Mohammad Qasim Rafiq** for teaching us and also helping us how to learn. They have been great sources of inspiration to us and we thank them from the bottom of my heart.

We would like to thank all our friends and especially our classmates for all the thoughtful and mind stimulating discussions we had, which prompted us to think beyond the obvious. We are especially indebted to our parents for their love, sacrifice, and support and would like to thank our parents for raising us in a way to believe that we can achieve anything in life with hard work and dedication.

Ms. Somya Sharma (Roll Number: - 8817103032)

Mr. Animesh Kumar Jain (Roll Number: - 8817103008)

INDEX

CONTENTS **PAGE**

Abstract………..1

1. Introduction………………………………………………………………….2

1.1 Problem Statement…………………………………………………..2

1.2 Objective…………………………………………………………….2

1.3 Face Detection………………………………………………………2

1.3.1 Pre-Processing………………………………………………3

1.3.2 Classification………………………………………………..3

1.3.3 Localization…………………………………………………3

1.4 Face Recognition……………………………………………………3

1.4.1 Holistic Matching Methods…………………………………4

1.4.2 Feature Based Methods……………………………………..4

1.4.3 Hybrid Methods…………………………………………….4

1.5 Applications OF Face Recognition…………………………………5

1.5.1 Face Identification………………………………………....5

1.5.2 Access Control……………………………………………...5

1.5.3 Security……………………………………………………..6

1.5.4 Validate Identity At ATMs…………………………………6

1.5.5 Disease Diagnoses………………………………………….6

1.6 Dataset Discription…………………………………………………..7

2. Technique Used…………………………………………………………….....8

2.1 Haar Cascade Classifier……………………………………………..8

2.1.1 Haar Feature Selection……………………...………………8

2.1.2 Creating Integral Images……………………………………9

2.1.3 Adaboost Training………………………………………….10

2.1.4 Cascading Clasifier…………………………………………11

2.2 Local Binary Pattern Histogram……………………………………..12
2.2.1 Parameters…………………………………………………..12
2.2.2 Training The Algorithm……………………………………13
2.2.3 Appling The LBP Operation………………………………..13
2.2.4 Performing Recognition……………………………………15
3. Experimental Results……………………………………………………...16
3.1 Implementation Details………………………………………….....16
3.2 Training Images………………………………………………………...16
3.3 Testing Images……………………………………………………….18
3.4 Prediction Results……………………………………………………19
4. Limitations………………………………………………………………...20
5. Future Enhancement……………………………………………………20
6. Conclusion………………………………………………………………...20
7. References…………………………………………………………………21

ABSTRACT

The face is one of the easiest ways to distinguish the individual identity of each other. Face recognition is a personal identification system that uses personal characteristics of a person to identify the person's identity. Human face recognition procedure basically consists of two phases, namely face detection, where this process takes place very rapidly in humans, except under conditions where the object is located at a short distance away, the next is the recognition, which recognize a face as individuals. Efficient and accurate face recognition has been an important topic in the advancement of computer vision systems. The project aims to incorporate state-of-the-art technique for face recognition with the goal of achieving high accuracy in recognition and detection of the face. In this project, we use a completely machine learning based approach to solve the problem of face recognition in an end-to-end fashion. The machine is trained on the pre-processed dataset Yalefaces. The resulting system is fast and accurate, thus aiding those applications which require face recognition.

1 INTRODUCTION

1.1 PROBLEM STATEMENT

Everyday actions are increasingly being handled electronically, instead of pencil and paper or face to face. This growth in electronic transactions results in great demand for fast and accurate user identification and authentication. Face recognition technology may solve this problem since face is undeniably connected to its owner. The availability of numerous commercial face recognition systems attests to the significant progress achieved in the research field. Despite these achievements face recognition continues to be an active topic in the field of computer vision research. This is due to the fact that current systems perform well under relatively controlled environments but tend to suffer when variations in different factors (such as pose, illumination etc.) are present. Therefore, the goal is to increase the robustness of the systems against different factors.

1.2 OBJECTIVE

We aim to learn about the machine learning techniques for face detection and recognition and to implement a system using a pre-processed dataset for better accuracy in face recognition.

1.3 FACE DETECTION

Face detection can be regarded as a specific case of object-class detection. In object-class detection, the task is to find the locations and sizes of all objects in an image that belong to a given class. The primary aim of face detection algorithms is to determine whether there is any face in an image or not. It is difficult because although commonalities exist between faces, they can vary considerably in terms of age, skin color and facial expression. The problem is further complicated by differing lighting conditions, image qualities and geometries, as well as the possibility of partial occlusion and disguise. An ideal face detector would therefore be able to detect the presence of any face under any set of lighting conditions, upon any background. The face detection task can be broken down into two steps. The first step is a classification task that takes some arbitrary image as input and outputs a binary value of yes or no, indicating whether there are any faces present in the image. The second step is the face localization task that aims to take an image as input and output the location of any face or faces within that image as some bounding box with (x, y, width, height).

The face detection system can be divided into the following steps:-

1.3.1 PRE-PROCESSING

To reduce the variability in the faces, the images are processed before they are fed into the system. All positive examples that is the face images are obtained by cropping images with frontal faces to include only the front view. All the cropped images are then corrected for lighting through standard algorithms.

1.3.2 CLASSIFICATION

Algorithms are used to classify the images as faces or non-faces by training on these examples. We have implemented haar cascade classifier for detected whether an image consist of a face or not.

1.3.3 LOCALIZATION

The trained algorithm is then used to search for faces in an image and if present, localize them in a bounding box and return the position on that region.

1.4 FACE RECOGNITION

Face recognition is the task of identifying an already detected face as a known or unknown face. In order to develop a useful and applicable face recognition system several factors need to be take in hand.

1. The overall speed of the system from detection to recognition should be acceptable.
2. The accuracy should be high.
3. The system should be easily updated and enlarged, that is easy to increase the number of subjects that can be recognized.

There are number of methods for face recognition some of them are as follows:

1.4.1 HOLISTIC MATCHING METHODS

In holistic approach, the complete face region is taken into account as input data into face catching system. One of the best example of holistic methods are Eigenfaces (most widely used method for face recognition), Principal Component Analysis, Linear Discriminant Analysis and independent component analysis etc.

1.4.2 FEATURE-BASED (STRUCTURAL) METHODS

In this methods local features such as eyes, nose and mouth are first of all extracted and their locations and local statistics (geometric or appearance) are fed into a structural classifier. A big challenge for feature extraction methods is feature restoration, this is when the system tries to retrieve features that are invisible due to large variations. These methods are distinguished between three different extraction methods:

1. Generic methods based on edges, lines, and curves.
2. Feature-template-based methods
3. Structural matching methods that take into consideration geometrical constraints on the features.

1.4.3 HYBRID METHODS

Hybrid face recognition systems use a combination of both holistic and feature extraction methods. Generally 3D Images are used in hybrid methods. The image of a person's face is caught in 3D, allowing the system to note the curves of the eye sockets or the shapes of the chin or forehead and so on. Even a face in profile would serve because the system uses depth, and an axis of measurement, which gives it enough information to construct a full face. Thus, this method comprises of the following stages:

1. Detection - Capturing a face either a scanning a photograph or photographing a person's face in real time.

2. Position - Determining the location, size and angle of the head.

3. Measurement - Assigning measurements to each curve of the face to make a template with specific focus on the outside of the eye, the inside of the eye and the angle of the nose.

4. Representation - Converting the template into a code (a numerical representation of the face).

5. Matching - Comparing the received data with faces in the existing database.

1.5 APPLICATIONS OF FACE RECOGNITION

1.5.1 FACE IDENTIFICATION

Face recognition systems identify people by their face images. Face recognition systems establish the presence of an authorized person rather than just checking whether a valid identification (ID) or key is being used or whether the user knows the secret personal identification numbers (Pins) or passwords. The following are example. To eliminate duplicates in a nationwide voter registration system because there are cases where the same person was assigned more than one identification number. The face recognition system directly compares the face images of the voters and does not use ID numbers to differentiate one from the others. When the top two matched faces are highly similar to the query face image, manual review is required to make sure they are indeed different persons so as to eliminate duplicates.

1.5.2 ACCESS CONTROL

In many of the access control applications, such as office access or computer logon, the size of the group of people that need to be recognized is relatively small. The face pictures are also caught under natural conditions, such as frontal faces and indoor illumination. The face recognition system of this application can achieve high accuracy without much co-operation from user. The following are the example. Face recognition technology is used to monitor continuously who is in front of a computer terminal. It allows the user to leave the terminal without closing files and logging out. When the user leaves for a predetermined time, a screen saver covers up the work and disables the mouse & keyboard. When the user comes back and is recognized, the screen saver clears and the previous session appears as it was left. Any other user who tries to logon without authorization is denied.

1.5.3 SECURITY

Today more than ever, security is a primary concern, especially at airports for airline staff office and passengers. Airport protection systems that use face recognition technology have been implemented at many airports around the world. In October, 2001, Fresno Yosemite International (FYI) airport in California deployed Visage's face recognition technology for airport security purposes. The system is designed to alert FYl airport public safety officers whenever an individual matching the appearance of a known terrorist suspect enters the airport's security checkpoint. Anyone recognized by the system would have further investigative processes by public safety officers. Computer security has also seen the application of face recognition technology. To prevent someone else from changing files or transacting with others when the authorized individual leaves the computer terminal for a short time, users are continuously authenticated, checking that the individual in front of the computer screen or at a user is the same authorized person who logged in.

1.5.4 VALIDATE IDENTITY 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 ATMs cards are who they say they are. Face recognition is currently being used at ATMs in Macau to protect peoples' identities.

1.5.5 DISEASE DIAGNOSES

Face recognition can be used to diagnose diseases that cause detectable changes in appearance. As an example, the National Human Genome Institute Research Institute in USA, uses face recognition to detect a rare disease called DiGeorge syndrome, in which there is a portion of the 22nd chromosome missing. Face recognition has helped diagnose the disease in 96% of cases. As algorithms get even more sophisticated, face recognition will become an invaluable diagnostic tool for all sorts of conditions.

1.6 DATASET DESCRIPTION

In this project, we have downloaded the dataset of facial images from Yale University site known as Yalefaces which contains 165 grayscale images in GIF format of 15 individuals. There are 11 images per subject, one per different facial expression or configuration: center-light, w/glasses, happy, left-light, w/no glasses, normal, right-light, sad, sleepy, surprised, and wink.

Fig: Yalefaces Dataset

- This data set has been downloaded from

  http://cvc.cs.yale.edu/cvc/projects/yalefaces/yalefeces.html

- In our dataset each subject is having 10 images to get trained.
- So having 15 subjects we get 15 * 10 = 150 total images to get trained.
- 1 image from each subject has been taken for testing process.

2 TECHNIQUE USED

There are many algorithms which are used for detecting and recognizing the face in an image. But we have implemented only two main algorithms which are used by us for detecting and recognizing.

2.1 HAAR CASCADE CLASSIFIER

Haar Cascade is a machine learning object detection algorithm used to identify objects in an image or video and based on the concept of features proposed by Paul Viola and Michael Jones in their paper "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images. The algorithm has four stages:

2.1.1 HAAR FEATURE SELECTION

First step is to collect the Haar Features. A Haar feature considers adjacent rectangular regions at a specific location in a detection window, sums up the pixel intensities in each region and calculates the difference between these sums. This difference is then used to categorize subsections of an image. For example, let us say we have an image database with human faces. It is a common observation that among all faces the region of the eyes is darker that the region of the cheeks. Therefore, a common Haar feature for face detection is a set of two adjacent rectangles that lie above the eye and the cheek region. The position of these rectangles is defined relative to a detection window that acts like a bounding box to the target object (face in this case).

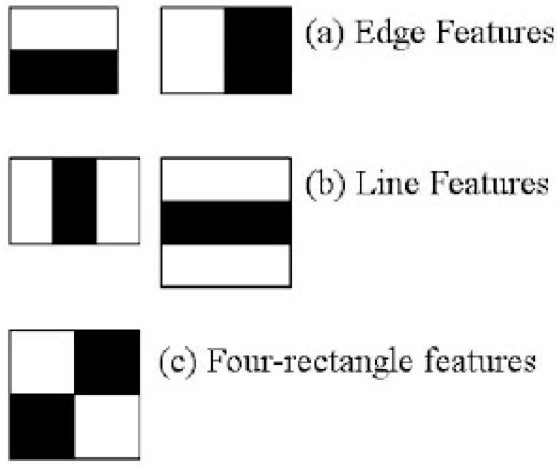


Fig: Haar Features

2.1.2 CREATING INTEGRAL IMAGES

An integral image is summed-area table is a data structure and algorithm for quickly and efficiently generating the sum of values in a rectangular subset of a grid. It is used to speed up the calculation of differences between the pixels of the selected region.

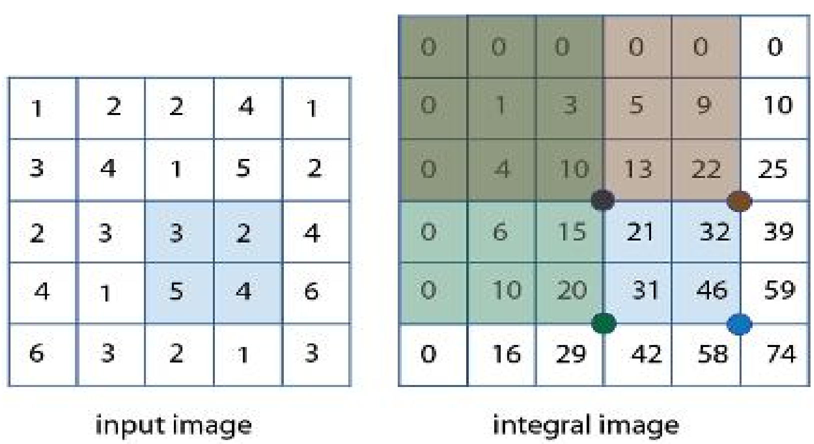


Fig: Integral Image

2.1.3 ADABOOST TRAINING

Among all these features we calculated, most of them are irrelevant. For example, consider the image below. 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 applying on cheeks or any other place is irrelevant.

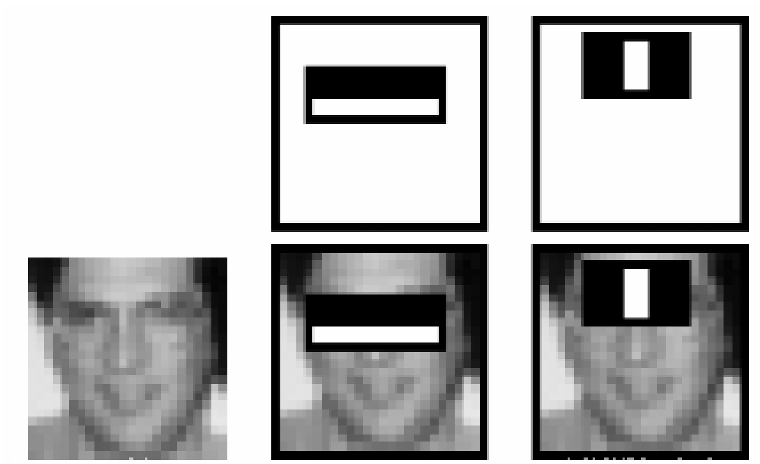


Fig: Selecting Haar Features

This is accomplished using a concept called **Adaboost** which both selects the best features and trains the classifiers that use them. This algorithm constructs a “strong” classifier as a linear combination of weighted simple “weak” classifiers.

During the detection phase, a window of the target size is moved over the input image, and for each subsection of the image and Haar features are calculated. This difference is then compared to a learned threshold that separates non-objects from objects. Because each Haar feature is only a "weak classifier" (its detection quality is slightly better than random guessing) a large number of Haar features are necessary to describe an object with sufficient accuracy and are therefore organized into **cascade classifiers** to form a strong classifier.

2.1.4 CASCADING CLASSIFIERS

The cascade classifier consists of a collection of stages, where each stage is an ensemble of weak learners. The weak learners are simple classifiers called *decision stumps*. Each stage is trained using a technique called boosting. *Boosting* provides the ability to train a highly accurate classifier by taking a weighted average of the decisions made by the weak learners.

Each stage of the classifier labels the region defined by the current location of the sliding window as either positive or negative. *Positive* indicates that an object was found and *negative* indicates no objects were found. If the label is negative, the classification of this region is complete, and the detector slides the window to the next location. If the label is positive, the classifier passes the region to the next stage. The detector reports an object found at the current window location when the final stage classifies the region as positive.

The stages are designed to reject negative samples as fast as possible. The assumption is that the vast majority of windows do not contain the object of interest. Conversely, true positives are rare and worth taking the time to verify.

- A *true positive* occurs when a positive sample is correctly classified.
- A *false positive* occurs when a negative sample is mistakenly classified as positive.
- A *false negative* occurs when a positive sample is mistakenly classified as negative.

To work well, each stage in the cascade must have a low false negative rate. If a stage incorrectly labels an object as negative, the classification stops, and you cannot correct the mistake. However, each stage can have a high false positive rate. Even if the detector incorrectly labels a non-object as positive, we can correct the mistake in subsequent stages. Adding more stages reduces the overall false positive rate, but it also reduces the overall true positive rate.

Cascade classifier training requires a set of positive samples and a set of negative images. We must provide a set of positive images with regions of interest specified to be used as positive samples. We can use the Image Labeler to label objects of interest with bounding boxes. The Image Labeler outputs a table to use for positive samples. We also must provide a set of negative images from which the function generates negative samples automatically.

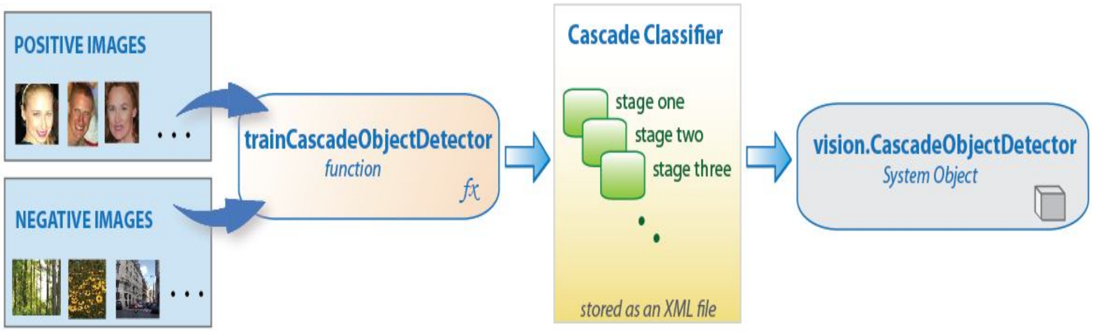


Fig: Cascade Classifier

2.2 LOCAL BINARY PATTERN HISTOGRAM(LBPH)

Local Binary Pattern (LBP) is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number.

It was first described in 1994 (LBP) 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 (HOG) 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. The algorithm is into the following steps:

2.2.1 PARAMETERS

LBPH uses four parameters.

Radius: The radius is used to build the circular local binary pattern and represents the radius around the central pixel. In our project we have set it to 1.

Neighbors: The number of sample points to build the circular local binary pattern. Keep in mind: the more sample points you include, the higher the computational cost. In our project we have set it to 8.

Grid X: The number of cells in the horizontal direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. In our project we have set it to 8.

Grid Y: The number of cells in the vertical direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. In our project we have set it to 8.

2.2.2 TRAINING THE ALGORITHM

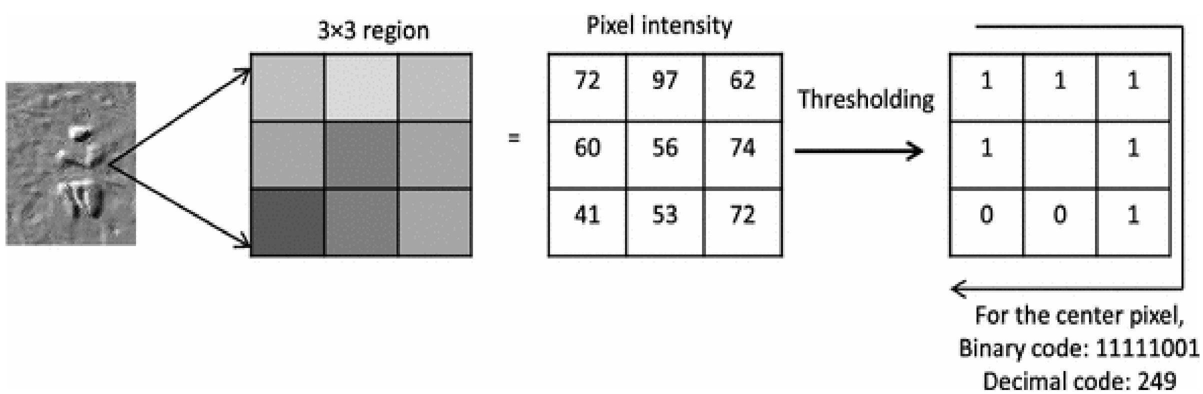
To do so we have taken the yalefaces dataset in our project. The detected faces from the Haar Cascade Classifier are used to train the LBPH algorithm along with the subject names as labels to identify the image.

2.2.3 APPLYING THE LBP OPERATION

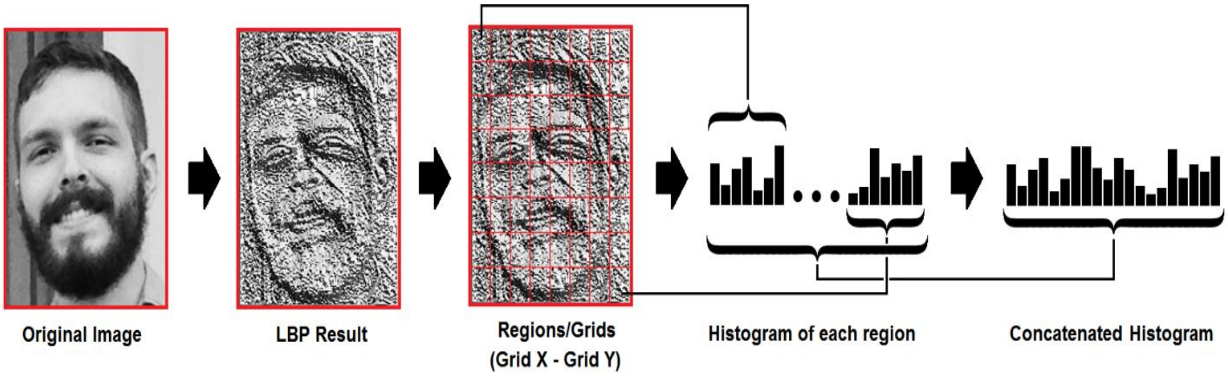
STEP 1: We take part of an image as a 3x3 matrix containing the intensity of each pixel.

STEP 2: Now, we take the central pixel as threshold and if the intensity of the center pixel is greater or equal to its neighbor, then we denote it with 1 and 0 if not.

STEP 3: The resulting matrix will contain only binary values. Then we concatenate a binary value from each position of the matrix and convert this binary value to a decimal value and set it to the central value of the matrix.



STEP 4: We perform the same procedure for all the pixels of the image and at the end, we have a new image which represents all the Linear Binary Codes.



STEP 5: We divide the resulting image into several regions and generate a histogram for each region.

STEP 6: Then, we concatenate each histogram to create a new histogram which represents the characteristics of the original image.

A more formal description of the LBP operator can be given as:

$$LBP\ (x_c, y_c) = \sum_{n=0}^{8} s(i_n - i_c) 2^n$$

$$where\ \ s(x) = \begin{cases} 1 & if\ x \geq 0 \\ 0 & otherwise \end{cases}$$

With (x_c, y_c) as central pixel with intensity i_n and i_c being the intensity of the the neighbor pixel and **is** the sign function.

2.2.4 PERFORMING RECOGNITION

The algorithm is already trained. Each histogram created is used to represent each image from the training dataset. So, given an input image, we performed the steps again for this new image and created a histogram which represents the image.

So to find the image that matches the input image we just needed to compare two histograms and return the image with the closest histogram. For this we used Euclidean distance algorithm is our project.

$$d(\mathbf{p},\mathbf{q}) = d(\mathbf{q},\mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

Fig: Euclidean Distance Formula

The algorithm output is the ID from the image with the closest histogram. The algorithm should also returns the calculated minimum distance between the histograms.

3 EXPERIMENTAL RESULT

3.1 IMPLEMENTATION DETAILS

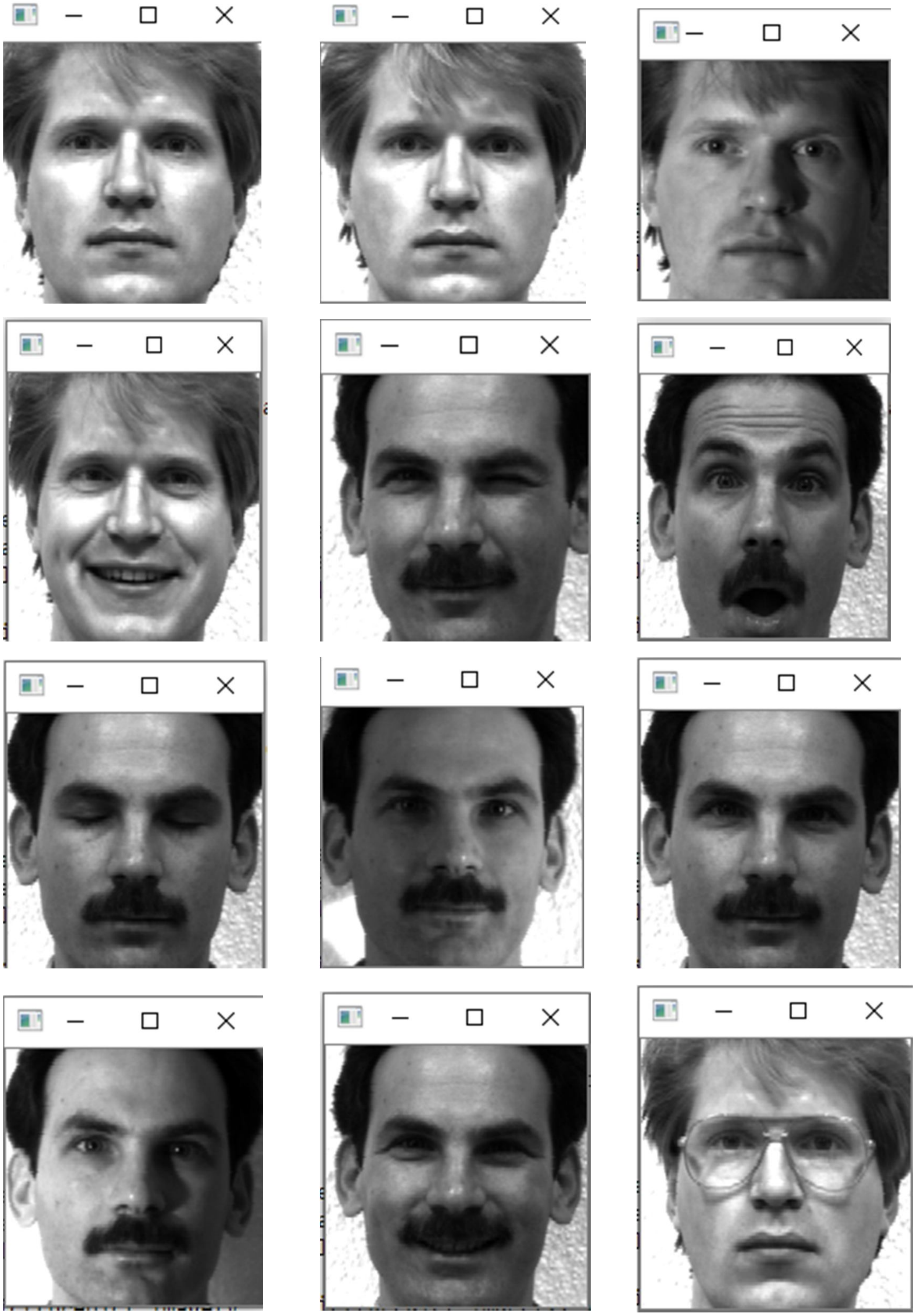
The project is implemented in python 3. Python libraries like OpenCV (Computer Vision 2) along with Numpy, OS, SKLearn and Pillow (PIL) was used for implementation. The system specifications on which the model is trained and evaluated are mentioned as follows:

CPU - Intel Core i5- 3320M 2.60 GHz

RAM – 8 GB DDR3.

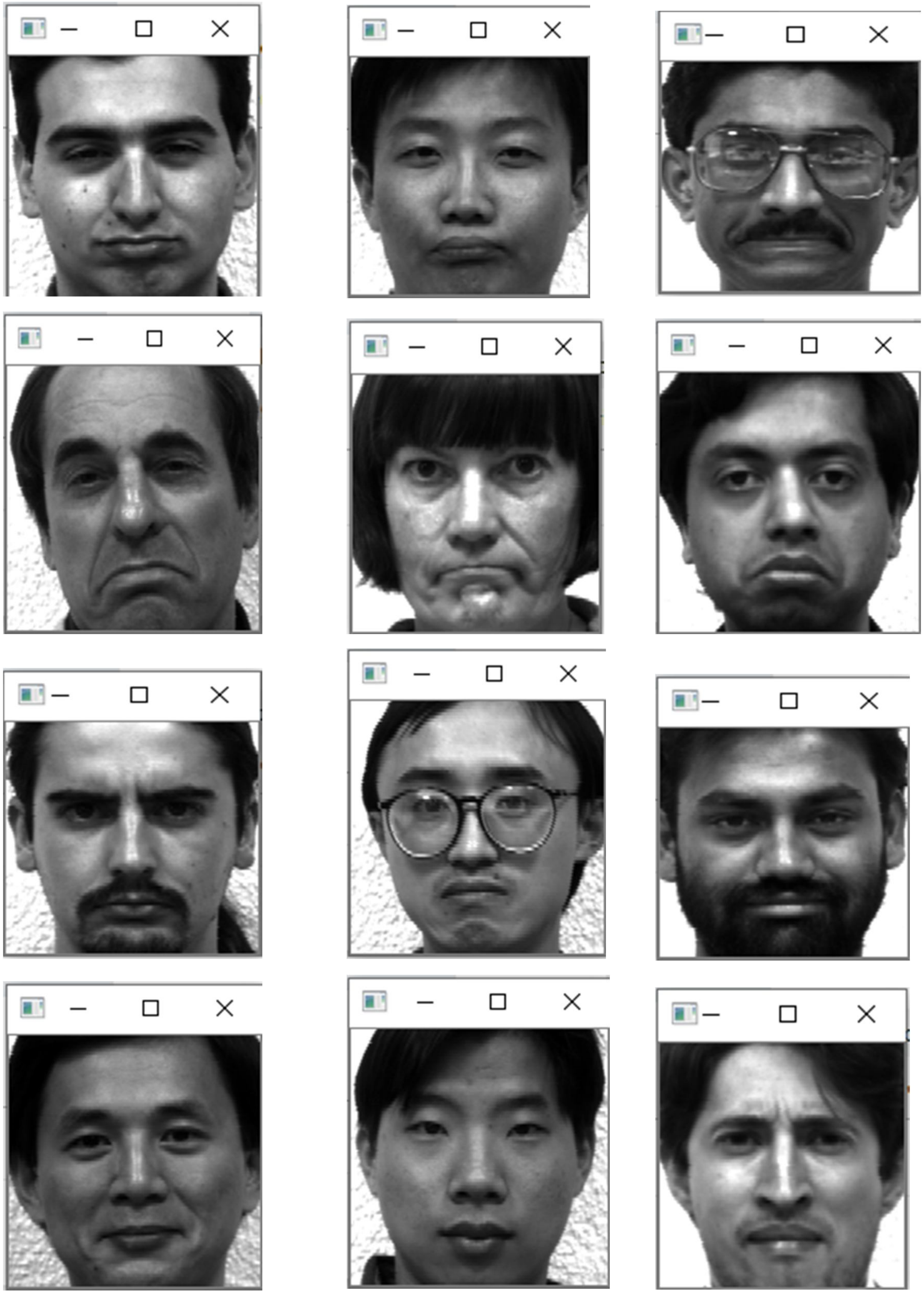
3.2 TRAINING IMAGES

These are few of the total 150 images used from yalefaces dataset for training the LBPH algorithm.



3.3 TESTING IMAGES

These are the total 15 images used from yalefaces dataset for testing the LBPH algorithm.



3.4 PREDICTION RESULTS

```
Test Subject- 1  is Correctly Recognized as Train Subject- 1  with minimum distance between histogram 39.651251744453255
Test Subject- 2  is Correctly Recognized as Train Subject- 2  with minimum distance between histogram 35.49467125513262
Test Subject- 3  is Correctly Recognized as Train Subject- 3  with minimum distance between histogram 43.47629904102858
Test Subject- 4  is Correctly Recognized as Train Subject- 4  with minimum distance between histogram 0.0
Test Subject- 5  is Correctly Recognized as Train Subject- 5  with minimum distance between histogram 32.31531084262701
Test Subject- 6  is Correctly Recognized as Train Subject- 6  with minimum distance between histogram 30.617649295244288
Test Subject- 7  is Correctly Recognized as Train Subject- 7  with minimum distance between histogram 42.07498858952902
Test Subject- 8  is Correctly Recognized as Train Subject- 8  with minimum distance between histogram 67.78277828058381
Test Subject- 9  is Correctly Recognized as Train Subject- 9  with minimum distance between histogram 48.12459852823689
Test Subject- 10  is Correctly Recognized as Train Subject- 10  with minimum distance between histogram 25.485826004911853
Test Subject- 11  is Correctly Recognized as Train Subject- 11  with minimum distance between histogram 37.70928857285139
Test Subject- 12  is Correctly Recognized as Train Subject- 12  with minimum distance between histogram 33.884765520689534
Test Subject- 13  is Correctly Recognized as Train Subject- 13  with minimum distance between histogram 36.261214553130074
Test Subject- 14  is Correctly Recognized as Train Subject- 14  with minimum distance between histogram 36.65706957155049
Test Subject- 15  is Correctly Recognized as Train Subject- 15  with minimum distance between histogram 32.86989791386682
```

Fig: Predicted images with id and distance between histograms.

```
Prediction Accuracy- 100.0 %
```

Fig: Accuracy of the predicted images as compared to test images.

4 LIMITATIONS

Our prediction model will not give such accuracy if training and testing is done with 50% of the dataset. And computation time will increase for comparatively larger datasets. Our system is not suitable for the real world environment as both testing and training as both the processes are done using pre-processed dataset which is unavailable in real world environment.

5 FUTURE ENHANCEMENTS

Other approaches with different algorithms like Eigenfaces and Fisherfaces can be done for better accuracy over comparatively larger datasets.

6 CONCLUSION

An accurate and efficient Face Recognition system has been developed which achieves comparable metrics with the existing state-of-the-art system. This project uses recent techniques in the field of computer vision and machine learning. Already studied and famous dataset Yalefaces was used and the evaluation was consistent. This can be used in real-time applications which require facial recognition. After training and testing we get 100% accuracy of our model.

7 REFERENCES:

- Sudha Narang, Kriti Jain, Megha Saxena, Aashna Arora, **"Comparison of Face Recognition Algorithms Using Opencv for Attendance System"** in International Journal of Scientific and Research Publications, Volume 8, Issue 2, February 2018.

- Md. Abdur Rahim, Md. Najmul Hossain, Tanzillah Wahid & Md. Shafiul Azam, **"Face Recognition using Local Binary Patterns (LBP)"** in Global Journal of Computer Science and Technology Graphics & Vision, Volume 13, Issue 4, Version 1.0, Year 2013.

- Paul Viola, Michael Jones, **"Rapid Object Detection using a Boosted Cascade of Simple Features"** at Accepted Conference On Computer Vision And Pattern Recognition 2001.

- Divyarajsinh N. Parmar, Brijesh B. Mehta **"Face Recognition Methods & Applications"** in International Journal of Computer Applications in Technology, Volume 4(1),84-86.

- Dataset-
  http://cvc.cs.yale.edu/cvc/projects/yalefaces/yalefaces.html

- Article-
  https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b
  http://www.willberger.org/cascade-haar-explained/

- Documentation of python libraries-
  https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html?highlight=local%20binary%20pattern%20histogram#face-database

- Video Lectures-
  https://www.youtube.com/watch?v=wpAwdsubl1w
  https://www.youtube.com/watch?v=uEJ71VlUmMQ