

**IDENTITY AUTHENTICATION SYSTEM USING FACE RECOGNITION
TECHNIQUES IN HUMAN-COMPUTER INTERACTION**

BY

Md. Khasrur Rahman

ID: 161-15-7214

This Report Presented in Partial Fulfillment of the Requirements for the Degree of
Bachelor of Science in Computer Science and Engineering

Supervised By

Dr. Md. Ismail Jabiullah

Professor

Department of CSE

Daffodil International University

Co-Supervised By

Md. Zahid Hasan

Assistant Professor & Associate Head

Department of CSE

Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

NOVEMBER 2019

APPROVAL

This Project/internship titled “**Identity Authentication System Using Face Recognition Techniques in Human-Computer Interaction**”, submitted by Md. Khasrur Rahman, ID No: 161-15-7214 to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 05/12/2019.

BOARD OF EXAMINERS

Dr. Syed Akhter Hossain
Professor and Head

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Chairman



Abdus Sattar
Assistant Professor

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Farah Sharmin
Senior Lecturer

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Dr. Md. Saddam Hossain
Assistant Professor

Department of Computer Science and Engineering
United International University

External Examiner

DECLARATION

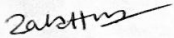
We hereby declare that, this project has been done by us under the supervision of **Dr. Md. Ismail Jabiullah, Professor, Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:



Dr. Md. Ismail Jabiullah
Professor
Department of CSE
Daffodil International University

Co-Supervised by:



Md. Zahid Hasan (ZH)
Assistant Professor & Associate Head
Department of CSE
Daffodil International University

Submitted by:



(Md. Khasrur Rahman)
ID: -161-15-7214
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First, we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the final year project successfully.

We really grateful and wish our profound our indebtedness to **Dr. Md. Ismail Jabiullah, Professor**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of “*Image Processing*” to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude to **Md. Zahid Hasan**, and Head, Department of CSE, for his kind help to finish our project and also to other faculty member and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

Automatic and separately identifying human faces is one of the most challenging issues at present. Although in a little light through the video monitoring cameras, recognizing people's faces is a bit default. Our proposed system handles thirty-five pixels better at minimum low resolution to detect human faces in different angles and different face detection. In our system, we use the LBPH algorithm to recognize people's faces in low light and at low resolutions. Here we will take 20 to 30 pictures of a human and create his dataset from different angles. Next, using that data set, we will train the data in the OpenCV library of Python. In the case of each data set of people will be taken from different angles, then all related information will be stored in a specific database. Each human's data set will be stored in a specific database if each input is inputted. Our system can display all its information in just 2 seconds by detecting a human face. To recognize people's faces, they will show all the data from the database and using all this information we will develop a security system or create a report system.

TABLE OF CONTENTS

CONTENTS	PAGE
Board of examiners	ii
Declaration	iii
Acknowledgments	iv
Abstract	v
CHAPTERS	
Chapter 1: INTRODUCTION	1-2
1.1 Introduction	1
1.2 Motivation	1
1.3 Expected Output	1
1.4 Objectives	2
Chapter 2: FACE RECOGNITION	3-11
2.1 Introduction	3
2.2 Principal Component Analysis	3
2.2.1 The Eigenface Approach	5
2.2.2 Testing	5
2.3 Advantages of Principal Component Analysis	5
2.4 Limitations of Principal Component Analysis	6
2.5 Linear Discriminant Analysis	6
2.5.1 Procedure	6
2.5.2 Testing:	7
2.6 Recognition	7
2.7 Captured Video for Face Detection	7
2.8 Read The video Captured window for Face Detection	8

2.9 Dataset Generator	9
2.10 Database Use for Save Person Information	10
2.11 Import Captured Person's Value in Database	10
2.12 Data Insert or Existing Information Update	11
2.13 Stored Information in Database	11
Chapter 3: DATA TRAINING	13-14
3.1 OS AND PIL Library for Training Data	13
3.2 Load The Training Data	13
3.3 Train The all Image by Id	14
3.4 All Trained Data	14
Chapter 4: DETECTOR	15-14
4.1 OS AND PIL Library for Training Data	15
4.2 Loading Recognizer	15
4.3 Recognize the Unknown face	16
4.4 The identified faces are round color and font color around the face	17
4.5 Get The face information From The database	17
Chapter 5: METHODOLOGY	18-20
5.1 Flowchart	18
5.2 Fundamental Steps in Processing Image	18
5.3 Local Binary Patterns Histograms Algorithm	19
5.4 Working Process	19
5.5 Internal Software Data Structure	19
5.6 False Acceptance Rate	20
Chapter 6: SUMMARY, AND CONCLUSION	21-22
6.1 Summary	21
6.2 Conclusions	21
6.3 Implication for Further Study	22
REFERENCES	23

LIST OF FIGURES

FIGURES	PAGE NO
Figure1: Eigenfaces	4
Figure 2: Diagram of MPCALDA	6
Figure 3: Recognition Using MPCALDA	7
Figure 4: Open Camera for Human Face Recognition	8
Figure 5: Read Video from Camera for Human Face Recognition	8
Figure 6: video capturing window	9
Figure 7: Gray Scale Image	10
Figure 8: Database Design	10
Figure 9: Data Input	11
Figure 10: For Inserting Data	12
Figure 11: For Updating Data	12
Figure 12: Stored Information Which Is Stored in Database	12
Figure 13: Import OS and PIL	13
Figure 14: Create the paths of the image	13
Figure 15: Training data	14
Figure 16: Stored Information Which Is stored in database	14
Figure 17: Identify face by ID	15
Figure 18: Identify face by ID	16
Figure 19: Identify unknown face	16
Figure 20: All Trained Data	17
Figure 21: Identify unknown face	17

LIST OF TABLES

TABLES	PAGE NO
Table 1: Flowchart	18
Table 2: Internal Software Data Structure	19
Table 3: False Rejection Rate	19
Table 4: Face Recognition Rate	20

CHAPTER 1

INTRODUCTION

1.1 Introduction

We have divided our system into three specific areas for developing human face recognition. Before developing a security system that recycles human faces, first of all we need to develop a system of recognizing human faces. Before developing a face recognition program, we need to first create a data set from each different face capture. Each individual captured image is assigned a specific ID and all the information of the person captured by that ID will be stored in the database later. Suppose we have to create two human data sets; the first man's ID will be the ID 1 and the second man's ID will be ID 2. And according to each ID, all of those people's information will be stored in the database. Then we will train those Tata set pictures for the next recognition or predict face from the live video frame.

These are the three major things we will do for human face recognition:

1. Dataset Creator
2. Trainer
3. Detector

1.2 Motivation

In today's networked world the need to maintain security of information or physical property is becoming both increasingly important and increasingly difficult. In countries like Nepal the rate of crimes are increasing day by day. No automatic systems are there that can track person's activity. If we will be able to track Facial expressions of persons automatically then we can find the criminal easily since facial expressions changes doing different activities. So, we decided to make a Facial Expression Recognition System. We are interested in this project after we went through few papers in this area. The papers were published as per their system creation and way of creating the system for accurate and reliable facial expression recognition system. As a result, we are highly motivated to develop a system that recognizes facial expression and track one person's activity

1.3 Expected Output

Our aim, which we believe we have reached, was to develop a method of face recognition that is fast, robust, reasonably simple and accurate with a relatively simple and easy to understand algorithms and techniques. The examples provided in this thesis are real-time and taken from our own surroundings.

Face detection is used in many places now a days especially the websites hosting images like Picasa, photo bucket and Facebook. The automatically tagging feature adds a new dimension to sharing pictures among the people who are in the picture and also gives the idea to other people about who the person is in the image. In our project, we have studied and implemented a pretty simple but very effective face detection algorithm which takes human skin color into account.

1.4 Objectives

The project mainly aims to come up with a solution to the recognized faces problem by Subproblems dividing it into sub-problems of classifications of some specific Action Units. The projects scope includes not only the two class problems which tell about whether an Action Unit is on or off, but also the multi-class problems that will inform the user about multi occurrences of more than one Action Unit at the same time. For this, different methodologies and techniques for feature extraction, nor-malaxation, selection and classification. Solutions to these problems as well as taking the computational complexity and timing issues into consideration. The project objective is to implement face recognition in an optimum way in terms of run time onto the embedded system. Various algorithms and method-ology are studied and hardware resources planning will be done to achieve the goal. This kind of face recognition embedded system can be widely used in our daily life in different sectors. We hope that human life can be greatly helped with this technology.

CHAPTER 2

FACE RECOGNITION

2.1 Introduction

The face recognition algorithms used here are Principal Component Analysis (PCA), Multilinear Principal Component Analysis (MPCA) and Linear Discriminant Analysis (LDA). Every algorithm has its own advantage. While PCA is the most simple and fast algorithm, MPCA and LDA which have been applied together as a single algorithm named MPCALDA provide better results under complex circumstances like face position, luminance variation etc. Each of them has been discussed one by one below.

2.2 Principal Component Analysis

Principal component analysis (PCA) was invented in 1901 by Karl Pearson. PCA involves a mathematical procedure that transforms a number of possibly correlated variables into a number of uncorrelated variables called principal components, related to the original variables by an orthogonal transformation. This transformation is defined in such a way that the first principal component has as high a variance as possible (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it be orthogonal to the preceding components. PCA is sensitive to the relative scaling of the original variables. Depending on the field of application, it is also named the discrete Korhonen–Louver transform (KLT), the Hotelling transform or proper orthogonal decomposition (POD). The major advantage of PCA is that the eigenface approach helps reducing the size of the database required for recognition of a test image. The trained images are not stored as raw images rather they are stored as their weights which are found out projecting each and every trained image to the set of eigenfaces obtained

2.1.1 The Eigenface Approach

In the language of information theory, the relevant information in a face needs to be extracted, encoded efficiently and one face encoding is compared with the similarly encoded database. The trick behind extracting such kind of information is to capture as many variations as possible from the set of training images. Mathematically, the principal components of the distribution of faces are found out using the eigenface approach. First the eigenvectors of the covariance matrix of the set of face images is found out and then they are sorted according to their corresponding eigenvalues. Then a threshold eigenvalue is taken into account and eigenvectors with eigenvalues less than that threshold values are discarded. So ultimately the eigenvectors having the most significant eigenvalues are selected. Then the set of face images are projected into the significant eigenvectors to obtain a set called eigenfaces. Every face has a contribution to the eigenfaces obtained. The best M eigenfaces from a M dimensional subspace is called “face space” Each individual face can be represented exactly as the linear combination of “eigenfaces” or each face can also be approximated using those significant eigenfaces obtained using the most significant eigen values.



Figure1: Eigenfaces

The approach to face recognition involves the following initialization operations:

1. Acquire an initial set of N face images (training images).
2. Calculate the eigenface from the training set keeping only the M images that correspond to the highest eigenvalues. These images define the “face space”. As new faces are encountered, the “eigenfaces” can be updated or recalculated accordingly.
3. Calculate the corresponding distribution in M dimensional weight space for each known individual by projecting their face images onto the “face space”.
4. Calculate a set of weights projecting the input image to the M “eigenfaces”.

2.2.2 Testing

According to the procedure given above “eigen faces” of the stored database is found out. Then feature vector of each individual is calculated by projecting it onto the set of eigenface. When a test image comes feature, vector is calculated exactly in the same way. It compares with the stored database by calculating the distance between two vectors. The training image who has minimum distance with the test image vector is our desired result.

2.3 Advantages Of PCA

1. It's the simplest approach which can be used for data compression and face recognition.
2. Operates at a faster rate.

2.4 Limitations Of PCA

1. Requires full frontal display of faces
2. Not sensitive to lighting conditions, position of faces.

3. Considers every face in the database as a different image. Faces of the same person are not classified in classes.

A better approach was studied and used to compensate these limitations which are called MPCALDA. While MPCA considers the different variations in images, LDA classifies the images according to same or different person

2.5 LDA

LDA which is known as Linear Discriminant Analysis is a computational scheme for evaluating the significance of different facial attributes in terms of their discrimination power. The database is divided into a number of classes each class contains a set of images of the same person in different viewing conditions like different frontal views, facial expression, different lighting and background conditions and images with or without glasses etc. It is also assumed that all images consist of only the face regions and are of same size. By defining all the face images of the same person in one class and faces of other people in different classes we can establish a model for performing cluster separation analysis. We have achieved this objective by defining two terms named “between class scatter matrix” and “within class scatter matrix”. The database used here is a FERET database which is a reference database for the testing of our studied algorithm

2.5.1 Procedure

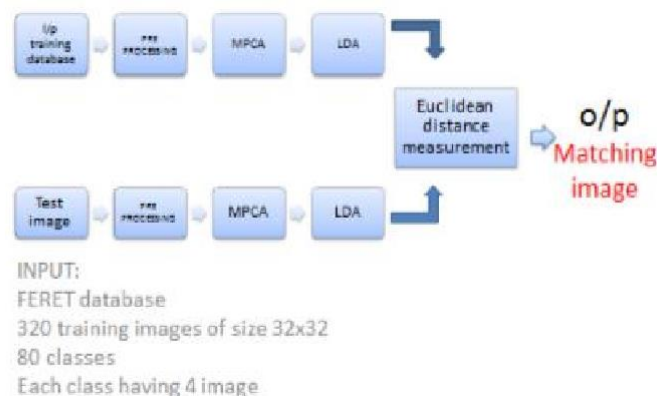


Figure 2: Diagram of MPCALDA

are not classified in classes.

A better approach was studied and used to compensate these limitations which are called MPCALDA. While MPCA considers the different variations in images, LDA classifies the images according to same or different person

2.6 LDA

LDA which is known as Linear Discriminant Analysis is a computational scheme for evaluating the significance of different facial attributes in terms of their discrimination power. The database is divided into a number of classes each class contains a set of images of the same person in different viewing conditions like different frontal views, facial expression, different lighting and background conditions and images with or without glasses etc. It is also assumed that all images consist of only the face regions and are of same size. By defining all the face images of the same person in one class and faces of other people in different classes we can establish a model for performing cluster separation analysis. We have achieved this objective by defining two terms named “between class scatter matrix” and “within class scatter matrix”. The database used here is a FERET database which is a reference database for the testing of our studied algorithm

2.6.1 Procedure

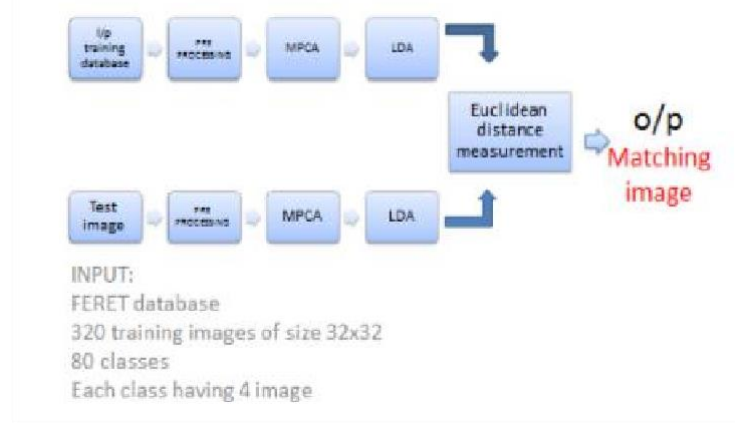


Figure 2: Diagram of MPCALDA

2.6.2 Testing

Now the TSB and TSW values are compared with each other to find a relation between them and the “fisher ratio” is found out which is the ratio of TSB & TSW.

$$\text{Fisher ratio} = \text{TSB}/\text{TSW}$$

Then the fisher ratio is sorted in the descending order. Then it is truncated to discard the features having a low fisher ratio value. Thereby we can minimize the dimensions of the database considering both within class and between class variations. In this method we are emphasizing the between class (different people) variation suppressing the within class variations (same person) which adds a new dimension to faster face recognition reducing the computational complexities.

2.7 Recognition

The features of test image were also extracted in the same fashion and were compared with the trained database. The recognition was successful on 95% of the occasion.



Figure 3: Recognition Using MPCALDA

The recognition algorithm was same as PCA and it involved the minimum Euclidian distance approach. Hence either way it detected a face either true or false. The quantitative success rates are provided below. Also, a comparison between PCA and MPCALDA algorithm is done to highlight the advantages of this algorithm over PCA.

2.8 Captured Video for Face Detection

In general, we're using Python here to identify any single face. After creating the first Python inversion, we used different libraries for Face Recognition. In detecting any face initially, we used two Python libraries that are NUMPY and OPENCV and these two lives are completely free and everyone can use it. After setting up everything, we use Cascade Classifier function of the OpenCV library, which is taken directly from the OpenCV folder. This file will be the root file of the system as this file will call all types of face detection functions.

After loading the classified, we will now focus on how to capture the video. After capturing the video, the video will be written to our program and converted to binary and after observing the pixel in pixel and see if there are any human faces.



Figure 4: Open Camera for Human Face Recognition

2.8 Read The video Captured window for Face Detection

After capturing the video from the web cam, we now need to know how to read that video.

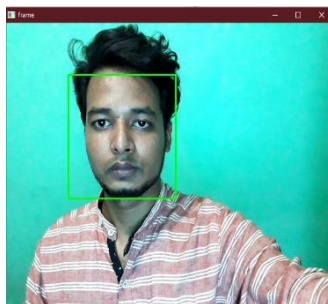


Figure 5: Read Video from Camera for Human Face Recognition

After reading the video, if there is any face, it will detect and it will show as a detector and we have to set how it will show. We used rectangles to indicate which face was captured

and identified them with the rectangles green color. Our entire project is identified by the green color quadrilateral on each face so It can be marked with any shape and any color. After we have finished capturing the image, we close the window with Cypress which can be customized by any key.

2.9 Dataset Generator

After all the video capturing process is done, we will see how to capture different images of the pages that will be captured from that video. We'll use an object called the **HarCascadeClassifier** to capture the faces of people who get their faces from every video.

The file name is: *haarcascade_frontalface_default.xml* file

When capturing each individual image, we will give each image a different name or ID so that it can later store the information of each individual human with that ID.

The faces of each human being captured from the camera will be stored in a specific folder and 20 to 30 pictures of each human being will be taken.

Next, we will create separate data sets with these 20 to 30 images that will be useful during our relationship. When an image captures by our program captures, it converts the image into a grayscale image and then puts it in the folder. The images that will be captured will be kept in each JPG format.

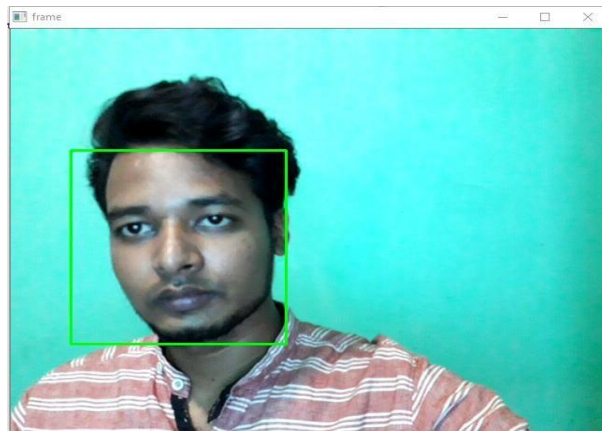


Figure 6: video capturing window

Here we will work with face detection. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. The good news is that OpenCV comes with a trainer as well as a detector. If you want to train your own classifier for any object like car, planes etc. you can use OpenCV to create one. Its full details are given here: Cascade Classifier Training.

If you do not want to create your own classifier, OpenCV already contains many pre-trained classifiers for face, eyes, smile, etc. Those XML files can be download from haarcascades directory.

After capturing all the images and converting them to grayscale images:

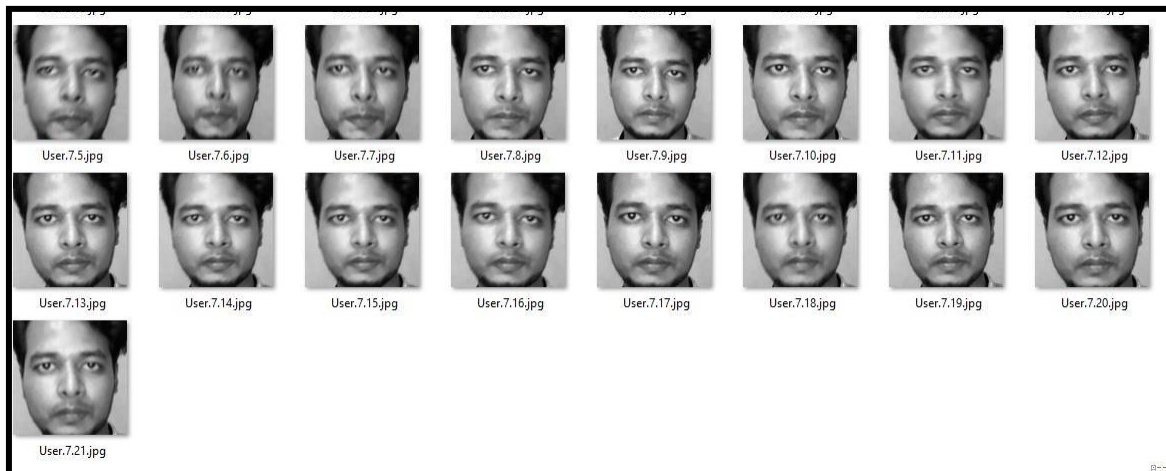


Figure 7: Gray Scale Image

2.10 Database Use for Save Person Information

As a database, we have used SQLite in our project. SQLite as a database is a very popular database interface for easy use. Now we have our own SQLite library built in Python for connection to our system with the database that we have used in our project. Now, to store each parson's data in SQLite database, we first have to create a table in the database that can be by any name.

The Library Which We are Imported: **#import sqlite3**




Table name: people		<input type="checkbox"/> WITHOUT ROWID							
	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Default value
1	ID	INT							NULL
2	Name	STRING							NULL
3	Age	INTEGER							NULL
4	Gender	INTEGER							NULL

Figure 8: Database Design

2.11 Import Captured Person's Value in Database

Now let's look at how to capture the value of captured people in our database. Database and Table are all created by us Now we will see how to save a parson's information in database when it is captured. Whenever his image is captured for insertion of a that Person in the database, he has to input his name and ID and he will not be able to capture his picture.

```
enter your id=1
enter your name="Khasrur Rahman"
```

Figure 9: Data Input

2.12 Data Insert or Existing Information Update

When a new person presses his ID while giving his information input, the database will check if his ID matches any previous ID. Then our system will be ready to update its information. And if its ID doesn't match any of the previous IDs, then the system will be ready to insert my information directly.

```
Insert Query: "INSERT INTO people (ID, Name) values (" +str (Id)+", "+str (Name)+")"
print 'insert'
```

```
enter your id=40
enter your name="new"
insert
```

Figure 10: Inserting Data

Update Query: UPDATE people SET Name =" +str (Name)+ " WHERE ID=" +str (Id)
print 'update'

```
enter your id=1
enter your name="Ratin"
update
```

Figure 11: Updating Data

2.13 Stored Information in Database

When captioning the image of each parson in the database, he gives his information input along with his ID. The import is all in the database.

Here is a sample of the information saved:

ID	Name	Age	Gender
1	ashaf	23	male
2	jubair	34	male
3	sajal	23	male
4	shuvo	23	male
5	rakib	13	male
6	panna	42	male
7	Ratin	NULL	NULL

Figure 12: Stored Information which is Stored in Database

According to the procedure given above “eigen faces” of the stored database is found out. Then feature vector of each individual is calculated by projecting it onto the set of eigenface. When a test image comes feature, vector is calculated exactly in the same way.

CHAPTER 3

DATATRaining

3.1 OS AND PIL Library for Training Data

We used two popular Python libraries here to train the data. The two libraries are:

1. OS
2. PIL

```
1 | import cv2,os
2 | import numpy as np
3 | from PIL import Image
```

Figure 13: Import OS and PIL

3.2 Load The Training Data

Now we will create a function to capture our captured images in a folder and hold their corresponding ID. We will hand over our functions to the folder in which all the images are held by their ID, to Train the data. Here we have named our function `Get_Images_and_Levels`. To load the image, we need to create the paths of the image:

```
1 | imagePaths=[os.path.join(path,f) for f in os.listdir(path)]
```

Figure 14: Create the Paths of the Image

We put the Face and ID into array. So that we can recognize multiple faces and many IDs together. One by one, it plots the image on the console simultaneously, then plots it into a data set.

We need to tell which folder the data set will be stored in and what name the data city will be in. We are storing a file named `trainer.yml` here in the specified folder. We are storing all the translation in a folder called `Data Set` on our system.

On this second phase, we must take all user data from our dataset and “trainer” the OpenCV Recognizer. This is done directly by a specific OpenCV function. The result will be a .yml file that will be saved on a “trainer/” directory.

3.3 Train The all Image by Id

We are storing all the translation in a folder called Data Set on our system.

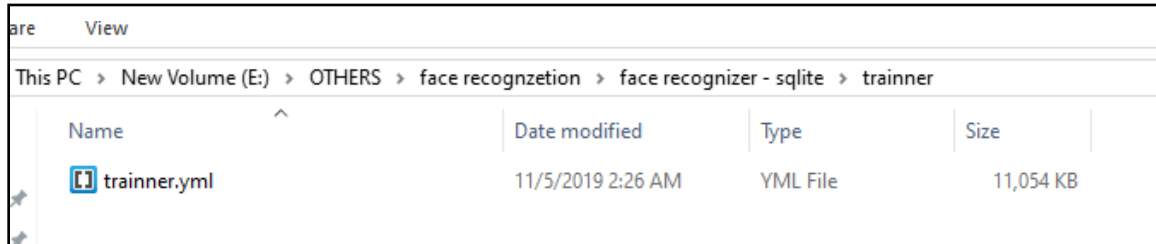


Figure 15: Training data

3.4 All Trained Data

Here is a sample of the faces we identified in the capturing window and the folders we identified with those pictures.



Figure 16: All Trained Data

CHAPTER 4

DETECTOR

4.1 Detect a face

Now we will see how the face detects from the data we have been trained. First, we have to try to figure out who's face is detecting face from video capture window. Our system will be able to see if its information is in our database, which can be seen in the capture window. For Detection, we will make three objects, the first is the Recognize object.

The second is the Camera object. The third is the Casket Classified object. In the capture window, the face of the capture will compile our training data to see if it actually falls inside data. We need to tell the system location to compress the face being detected.

4.2 Loading Recognizer

We have again used 2 libraries to recognize and load. The Library Are NumPy and OpenCV. Initially, the information we see in the window is shown only by its stored ID.



Figure 17: Identify Face by ID

On the picture, I show some tests done with this project, where I also have used photos to verify if the recognizer works.



Figure 18: Identify Face by ID

4.3 Recognize the Unknown face

If the face that can be found in the video capturing window does not match or cannot be found in our database, then how can our system find it?

When an unidentified face is found in the video capturing window, it will appear as a square green shape, indicating that this person's information is not in our database.

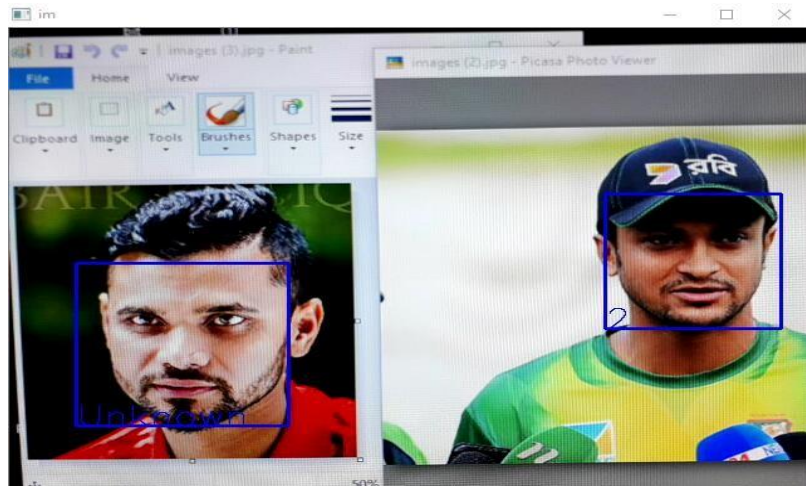


Figure 19: Identify Unknown Face

4.4 The Identified Faces Are Round Color and Font Color Around the Face

We identified the face that was detected in the capturing window with a green color and then used OpenCV as a default font. We showed all the information in our identified picture inside the box below the picture and the most color was green. The color and size and color of the account can all be changed as you wish

```
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY);  
cv2.rectangle(img, (x,y), (x+w,y+h), (255,0,0), 2)
```

Figure 20: Identify Face Font and Color

4.4 Get the Face Information from The Database

First, we identify different people's pages in a specific way and place them in a specific folder. And while doing all of the identified faces, we spoil the information data with a

specific ID of each face. Later, they traced the identified images and put a training file in a specific folder. Temporarily we tested whether our training kit information could properly match a human-identified face. We can see that our system is able to detect recognizable faces. Since our system is able to detect faces, we will show those identified faces with information from our database. Here is a sample of how our system can properly display all information from a known face database.

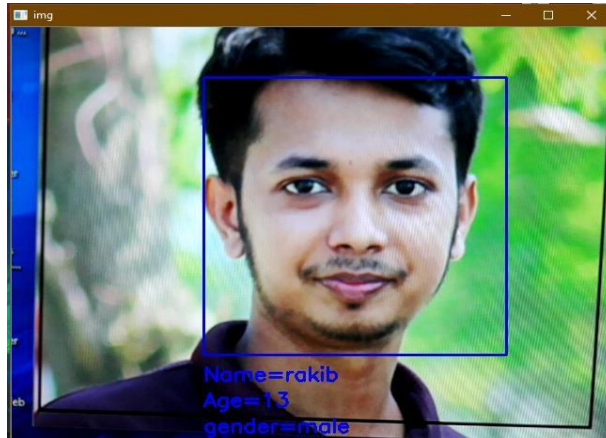


Figure 21: Final Result 1

CHAPTER 5

METHODOLOGY

5.1 Flowchart

working flow chart is given below:

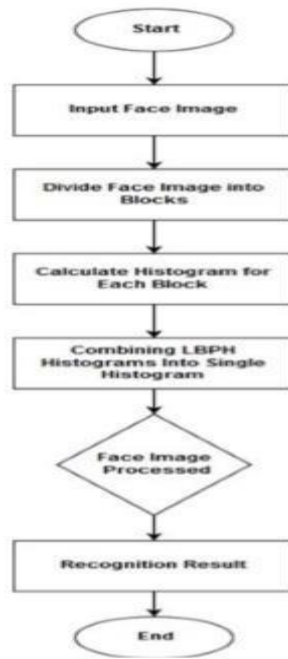


Table 1: Flowchart

5.2 Fundamental Steps in Processing Image

1. Image acquisition: to accumulate a digital image
2. Image pre-processing: to boost the image in ways in which will increase the probabilities for achievement of the opposite processes.
3. Image segmentation: to partitions associate input image into its constituent elements of objects.
4. Image segmentation: to convert the input file to a from appropriate for pc process.
5. Image description: to extract the options that end in some quantitative info of interest of options that area unit basic for differentiating one category of objects from another.

6. Image recognition: to assign a label to associate object supported the data provided by its description

5.3 Local Binary Patterns Histograms Algorithm

It is an easy resolution for the face recognition drawback, which might be recognizing each the front face and aspect face. LBP could be a straightforward and economical text description operator that labels the picture elements of a picture by thresholds the neighborhood of every pixel and that produces the result as a binary variety. Then the LBP combined with bar graph. we can represent the face pictures with an easy information vector.

5.4 Working Process

1. Divide the examined window into cells.
2. For each constituent in an exceedingly cell, (3X3 constituents) compare the pixel to every of its eight neighbors (left-top, left-middle, left-bottom and right-top, etc.) follow the pixels on a circle, i.e. right-handed or counterclockwise.
3. If the middle pixel's worth is bigger than the neighbor's worth, then write "1". otherwise, write "0". (example :250>90 = one, 90<50 =0 likewise establish the worth for every pixels).

5.5 Internal software data structure

passed back to XML mistreatment the intrinsic knowledge structures.

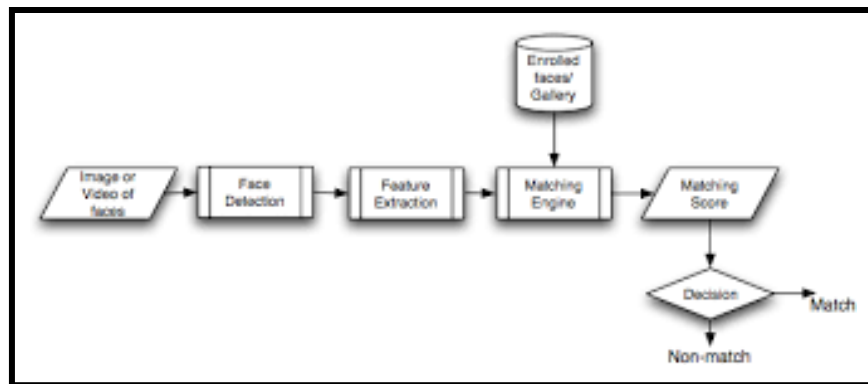


Table 2: Internal Software Data Structure

5.6 False Acceptance Rate

FAR = Number of False Acceptances / Number of testing faces

Number of testing faces	Number of training faces	False Rejected faces	FRR
7	10	3	42.8%
4	60	1	25%
2	100	0	0%

Table 3: False Rejection Rate

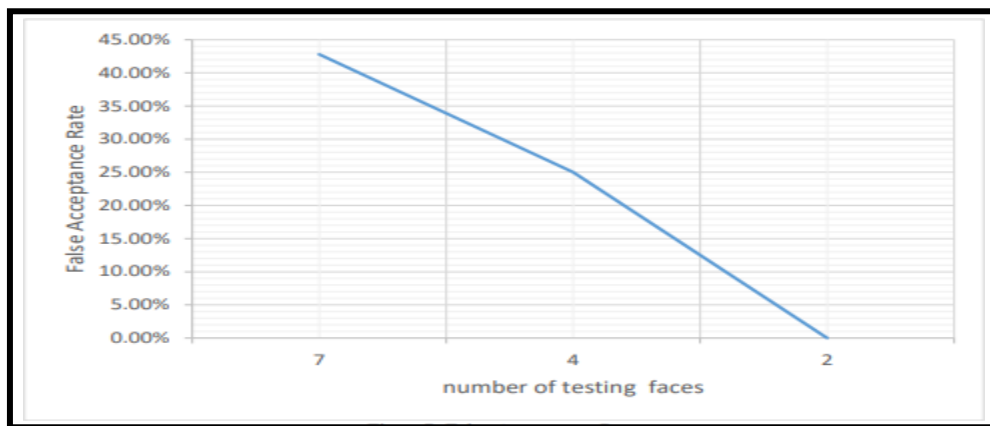


Table 4: False Rejection Rate

By analyzing each table, the man of science all over that the system includes a crossover rate of twenty fifth. Cross Over Rate (CER) is the purpose at False Rejection Rate is up to False Acceptance Rate. The emotion is comparatively low thus, the system is correct.

Test Case	Total Number of images	Total number of recognized images	Face Recognition Rate
1	20	18	90%
2	15	13	86.66%

Table 4: Face Recognition Rate

CHAPTER 6

SUMMARY, AND CONCLUSION

6.1 Summary

I am fully satisfied to see my final result. My system recognizes any faces within .2s. After recognize any faces its automatically get the data from the database and compare with them. By comparing the data, we know that the data is valid or not.

6.2 Conclusion

The computational models, which were implemented in this project, were chosen after extensive research, and the successful testing results confirm that the choices made by the researcher were reliable. The system with manual face detection and automatic face recognition did not have a recognition accuracy over 90%, due to the limited number of eigenfaces that were used for the PCA transform. This system was tested under very robust conditions in this experimental study and it is envisaged that real-world performance will be far more accurate. The fully automated frontal view face detection system displayed virtually perfect accuracy and in the researcher's opinion further work need not be conducted in this area.

6.3 Implication for Further Study

The implemented fully automated face detection and recognition system (with an eye detection system) could be used for simple surveillance applications such as ATM user security, while the implemented manual face detection and automated recognition system is ideal of mugshot matching. Since controlled conditions are present when mugshots are gathered, the frontal view face recognition scheme should display a recognition accuracy far better than the results, which were obtained in this study, which was conducted under adverse conditions. Department of ECE Page 48 Furthermore, many of the test subjects did not present an expressionless, frontal view to the system. They would probably be more compliant when a 6'5" policeman is taking their mugshot! In mugshot matching

applications, perfect recognition accuracy or an exact match is not a requirement. If a face recognition system can reduce the number of images that a human operator has to search through for a match from 10000 to even a 100, it would be of incredible practical use in law enforcement. The automated vision systems implemented in this thesis did not even approach the performance, nor were they as robust as a human's innate face recognition system. However, they give an insight into what the future may hold in computer vision.

REFERANCE

- [1] 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
- [2] J. CHAO W L, DING J J, LIU J Z. Facial expression recognition based on improved local binary pattern and class-regularized locality preserving projection. *Signal Processing*, 2015, 117:1-10.
- [3] J. HU Liqiao, QIU Runhe. Face recognition based on adaptive weighted HOG. *Computer Enigeering and Applications*, 2017, 53(3): 164-168.
- [4] J. Yu yan JIANG, Ping LI, Qing WANG. Labeled LDA model based on shared background topic. *Acta Electronica Sinica*, 2015, 2013, (9): 1794-1799.
- [5] J. WU Qi, WANG Tang-hong, LI Zhan-li. Imporved face recognition algorithm based on Gabor feature and collaborative representation. *Computer Engineering and Design*, 2016, 37(10): 2769-2774.
- [6] Aruni Singh, Sanjay Kumar Singh, Shrikant Tiwari, Comparison of face Recognition Algorithms on Dummy Faces, *The International Journal of Multimedia & Its Applications (IJMA)* Vol.4, No.4, August 2012.
- [7] XueMei Zhao, ChengBing Wei, A Real-time Face Recognition System Based on the Improved LBPH Algorithm, 2017 IEEE 2nd International Conference on Signal and Image Processing.
- [8] Varun Garg, Kritika Garg, Face Recognition Using Haar Cascade Classifier, *Journal of Emerging Technologies and Innovative Research (JETIR)* , December 2016, Volume 3, Issue 12.
- [9] Hongshuai Zhang, Zhiyi Qu Liping, Yuan GangLi, A Face Recognition Method Based on LBP Feature for CNN, 2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC).

[10] T. Chen, Y. Wotao, S. Z. Xiang, D. Comaniciu, and T. S. Huang, “Total variation models for variable lighting face recognition” IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(9):1519{1524, 2006

[11] Zhao and R. Chellappa “Robust face recognition using symmetric shape from-shading” Technical Report, Center for Automation Research, University of Maryland, 1999

[12] Zheng Xiang, Hengliang Tan, Wienling Ye. The excellent properties of dense gird-based HOG features on face recognition compare to gabor and LBP, 2018 volume issue 99.

Human Face recognition

ORIGINALITY REPORT

11 %	5 %	3 %	10 %
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	3 %
2	Submitted to RMIT University Student Paper	2 %
3	Aditya K. Saxena, Vijay K. Chaurasiya. "Multi-resolution texture analysis for fingerprint based age-group estimation", Multimedia Tools and Applications, 2017 Publication	2 %
4	Submitted to Savitribai Phule Pune University Student Paper	1 %
5	www.ijetcse.com Internet Source	1 %
6	www.haletestingsolutions.com Internet Source	1 %
7	Submitted to Higher Education Commission Pakistan Student Paper	1 %

Submitted to Federal University of Technology

8

Student Paper

<1%

9

"Proceedings of the International Conference on ISMAC in Computational Vision and Bio-Engineering 2018 (ISMAC-CVB)", Springer Science and Business Media LLC, 2019

Publication

<1%

10

Submitted to North West University

Student Paper

<1%

11

Submitted to University of Greenwich

Student Paper

<1%

Exclude quotes Off

Exclude matches Off

Exclude bibliography Off