AUTOMATIC STUDENT ATTENDANCE SYSTEM BASED ON FACE DETECTION AND RECOGNITION

A PROJECT REPORT

Submitted by

SRI VASANTHI.B [REGISTERNO:211417104270]

SUVATHI.T [REGISTER NO:211414104280]

UMAMAHESWARI.S [REGISTER NO:211417104284]

in partial fulfillment for the award of the

degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

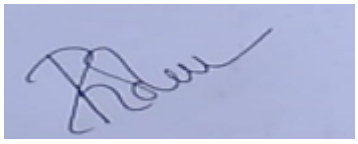
PANIMALAR ENGINEERING COLLEGE, CHENNAI-600123.

ANNA UNIVERSITY: CHENNAI 600025

AUGUST 2021

BONAFIDE CERTIFICATE

Certified that this project report **"AUTOMATIC ATTENDANCE SYSTEM BASED ON FACE RECOGNITION AND FACE DETECTION"** is the bonafide work of "**SRI VASANTHI.B(211417104270), SUVATHI.T(211417104280), UMAMAHESWARI.S(211417104284)**" who carried out the project work under my supervision.



SIGNATURE

Dr.S.MURUGAVALLI,M.E.,Ph.D.,
HEAD OF THE DEPARTMENT

DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,
COLLEGE, NAZARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

SIGNATURE

R.DEVI M.E.
SUPERVISOR

DEPARTMENT OF CSE,
PANIMALAR ENGINEERING
NAZARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

Certified that the above candidate(s) was/ were examined in the Anna University Project Viva-Voce Examination held on.....

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We would like express our heartfelt and sincere thanks to our Directors **Tmt.C.VIJAYARAJESWARI**, **Dr.C.SAKTHIKUMAR, M.E., Ph.D.,** and **Tmt. SARANYASREE SAKTHIKUMAR B.E.,M.B.A.,** for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.** for his timely concern and encouragement provided to us throughout the course.

We thank the HOD of CSE Department, **Dr. S.MURUGAVALLI , M.E.,Ph.D.,** for the support extended throughout the project.

We would like to thank our **Project Guide Mrs.R.DEVI M.E,Tech.,M.E.,**our class in charge **Dr.D.Anuradha,M.E.,Ph.D.,** and all the faculty members of the Department of CSE for their advice and suggestions for the successful completion of the project.

SUVATHI.T
UMAMAHESWARI.S
SRI VASANTHI.B

ABSTRACT

Face recognition can be considered one of the most successful biometric identification methods among several types of biometric identification including fingerprints, DNA, palm print, hand geometry, iris recognition, retina and odor/scent. Face recognition provides biometric identification that utilizes the uniqueness of faces for security purposes. The problem with face recognition using biometric identification is its lengthy process and the accuracy of the results. This paper proposes solutions for a faster face recognition process with accurate results. The proposed face recognition process was done using a hybrid process of Haar Cascades and Eigen face methods, which can detect multiple faces (55 faces) in a single detection process. This improved face recognition approach was able to recognize multiple faces with 91.67% accuracy level.

TABLE OF CONTENTS

LIST FIGURES

LIST OF SYMBOLS, ABBREVIATION

LIST OF SYMBOLS

MATLAB - multi-paradigm

F – Fail

D – detected

Q - quit

ABBREVIATION

LBPH-- LOCAL BINARY PATTERN HISTOGRAM

ER-- ENTITY RELATIONSHIP

UML-- Unified Modeling Language

DFD-- Data flow diagram

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Face recognition is a combination of machine learning and biometric techniques which holds the qualities of not only high precision but also reliability. For automatically detecting the human's face from the databases this system can be used. In recent years open computer vision has been widely used in different kinds of applications such as surveillance cameras, robotics etc. This technology is used for authentication, validation, authorization, and identification.

A face recognition system may be a technology capable of identifying or verifying an individual from a digital image or a video frame from a video source. There are multiple methods during which face recognition systems work, but generally, they work by comparing selected countenance from a given image with faces within a database. It's also described as a Biometric AI based application which will uniquely identify an individual by analyzing patterns supporting the person's facial texture and shape.

While initially a sort of computer application, it's seen wider uses in recent times on mobile platforms and in other sorts of technology, like robotics. It's typically used as access control in security systems and may be compared to other biometrics like fingerprint or eye iris recognition systems. Although the accuracy of face recognition system as a biometric technology is less than iris recognition and fingerprint recognition, it's widely adopted due to its contactless and non- invasive process. Recently, it's also become popular as a billboard identification and marketing tool. Other applications include advanced human-computer interaction, video surveillance, automatic indexing of images, video database, among others.

face like the attention centers, mouth, etc., and these were mathematically rotated by computer to catch up on pose variation. The distances between landmarks were also automatically computed and compared between images to work out identity.

1.1 PROBLEM DEFINITION

Attendance Management System is software developed for daily student attendance in schools, colleges and institutes. It facilitates access to the attendance information of a particular student in a particular class. For automatically detecting the human's face from the databases this system can be used. In recent years open computer vision has been widely used in different kinds of applications such as surveillance camera, robotics etc. This technology is used for authentication, validation, authorization, and identification. A study has been conducted by Budi (2009) about considering several factors of migrating manual attendance system. Firstly, the author mentioned that the most common way used to monitor one's presence at a place is by using signature on a paper while in a company, the clock machine is use to check the presence of employees. Secondly, he highlighted that the presence of students in education institution are checked according to the signature of students on the attendance sheet. Sometimes, lecturers need to call or check their actual presence in the class and this process is time-consuming and less effective and efficient.

CHAPTER 2

2.1 LITERATURE SURVEY

Face detection may be a technology that determines the situation and size of human face in arbitrary image. The countenance is detected and the other objects like trees, buildings and bodies etc are ignored from the digital image. A study has been conducted by Budi (2009) about considering several factors of migrating manual attendance system. Firstly, the author mentioned that the most common way used to monitor one's presence at a place is by using signature on a paper while in a company, the clock machine is use to check the presence of employees. Secondly, he highlighted that the presence of students in education institution are checked according to the signature of students on the attendance sheet. Sometimes, lecturers need to call or check their actual presence in the class and this process is time-consuming and less effective and efficient. Why attendance of the students is so important? Yao and Chiang (2011) mentioned that there was strong correlation between absence from the first day of class and overall result. Usually, for those students who did not come for the first class of a particular subject, he/she might lose many important details and information for entire semester

REVIEWS ON LITERATURE WORK:

Jireh Robert Jam proposed a system of Face Detection and Recognition Student Attendance System. He implements algorithms for face detection and recognition in image processing to create a system which will detect and recognize frontal faces of scholars during a classroom. In human interactions, the face is the most important factor as it contains important information about a person. So he developed a working prototype of a system that will facilitate class control during a classroom by detecting the frontal faces of scholar. But the main drawback is that the evaluation showed that recognition part can achieve approximately 60% recognition rate.

Cheng, et al. developed the system to manage the context of the scholars for the classroom lecture by using note PCs are all the students. Because this system uses the more of every student, the attendance and therefore the position of the students are obtained. However, it is difficult to understand the detailed situation of the lecture. His system takes images of faces. In recent decade, variety of algorithms for face recognition has been proposed, but most of these works affect only single image of a face at a time. By continuously observing of face information, his approach can solve the matter of the face detection and improve the accuracy of face recognition. The drawback was detection time was taking more to detect more faces and it can't able to detect some faces on the screen.

YoheiKawaguchi developed the system in order to obtain the attendance, positions and face images in classroom lecture; they proposed the attendance management system based on face recognition in the classroom lecture. The system estimates the attendance and the position of each student by continuous observation and recording.

The result of their preliminary experiment shows continuous observation improved the performance for estimation of the attendance. Their current work is focused on the method to obtain the different weights of each focused seat according to its location. Their drawback was to improve face detection effectiveness by using the interaction among our systems.

Rajath S Bharadwaj proposed the system that allots attendance to the recognized faces in the database. The basis of developing an automatic attendance management system is to computerize the standard method of taking attendance. His proposed system strives to outgrow the constraints of the existing systems and provides features such as detection of faces, extraction of features, and detection of extracted features and analysis of student's attendance. The system's

correctness in detecting and recognizing faces will be more due to use of large number of features(shape, color, LBP, wavelet, auto correlation, etc) of the face. The main drawback this system was attendance is updated only for a single face thou it scans many faces, and more than 5 faces detection was failed.

The[4] Authors had developed a system that capturing the images from camera or cc camera and applying techniques face detection and recognition can decrease the manual work from human and increase the security safety, taking the decision from this recognition result. Based in this face detection and recognition can used in implement so many application like automatic attendances system based on face recognition, worker attendances, security, safety, police application like finding thief in image that help to catching thief. In this system they have implemented an attendance system for a lecture section or laboratory by which lecturer in teaching assistant a record student's attendance. It saves time and effort, especially if it's a lecture with huge number of scholars. The complete system is implemented in MATLAB.

Sunil M P developed the system to implement an attendance system for a lectures or laboratory in which lecturer or teaching assistant can record student's attendance. It saves time and effort. The complete system is implemented with OpenCV and Rasbperry pi. This attendance system shows the use of facial recognition techniques for the purpose of student attendance. The result of the experiment shows the detection and recognition part. This method canal so detect multiple faces and can be Easily used in a classroom. The precision of face recognition is almost more than 90%, only the limitation of the system is if the image is out of the database it produces the faulty result.

Hari Prasanth proposed the solution for the present problem is thru automation of attendance system using face recognition. This project describes the tactic of

detection and recognition the face in real time. Here, the camera is fixed with in the classroom and which can capture the image. The faces with the image are detected then recognized with the database after which the attendance is marked. If the attendance is marked as absent, the message about the student's absence is shipped to their parents through Iot. This method requires only simple hardware for installation. The management of attendance in this technique is simpler and more accuracy. One difficult task in this system is face testing is considered to be unknown; otherwise, it is known and belongs to the person in the database.

CHAPTER 3

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The Existing System is single face recognition system and a new face detection approach using color base segmentation and morphological operations is presented. The algorithm uses color plane extraction, background subtraction, thresholding, morphological operations (such as erosion and dilation), filtering (to avoid false detection). Then particle analysis is done to detect only the face area in the image and not the other parts of the body. This method given result is poor performance and accuracy. So, we will move the proposed system.

Face is widely used for identification and resolution of humans. The face identification process refers to generating face identifiers and associating personal identity information with the identifier. The face resolution process refers to searching personal identity information or obtaining related personalized service by face identifier.Different from ID code which mainly consists of numbers and alphabets, face identifier needs to be generated by performing some facial image processing algorithms, including face detection, preprocessing, feature extraction and face identifier generation. In the face resolution process, face identifier matching algorithms need to be performed. In addition, the size of face identifier is larger and its structure is more complex than ID code. Therefore, the processes of face identification and resolution need more computation, communication and storage capabilities. Some researchers have adopted cloud computing to improve processing and storage capacity. Image processing deals with CNN and RNN algorithms with large dataset of trained images. Every image represents an Eigen vector, the data set helps produce variety for the system. A representation of these eigenvectors are called as Eigen face.

Figure 3.1.1

3.2 PROPOSED SYSTEM

The proposed system consists of 4 steps, including

(1) training of real time images

(2) multiple face detection using Haar-classifier

(3) comparison of trained real time images with images from the surveillance

(4) camera

(5) result based on the comparison.

We implement this application in an official environment, when an employee enters into the zone, the application automatically scales the current location with the static company's zone. Then the employee needs to do face verification. When the face is matched then the attendance will be added into the database. Thus, the attendance count will be added to the server. This project was created to detect multi-faces at a time. The accuracy level of detection was increased from the existing systems.

3.3 REQUIREMENT ANALYSIS AND SPECIFICATION

3.3.1 INPUT REQUIREMENT

The input requirements are Web cam, processor-I3 core and hard disk consist of 500 GB then the RAM should be 4 GB and higher versions of the above mentioned requirements are needed to be used.

3.3.2 OUTPUT REQUIREMENT

A computer pc/laptop and output of the attendance system is viewed in excel sheet.

3.3.3 FUNCTIONAL REQUIREMENT

- User/client should have a android phone with latest version of android .To get notification message in the android app.
- The webcam installed for capturing the video have to be working in good condition

3.4 TECHNOLOGY STACK

3.4.1 HARDWARE REQUIREMENTS

- Hard Disk : 500GB and Above
- RAM : 4GB and Above
- Processor : I3 and Above
- Webcam

3.4.2 SOFTWARE REQUIREMENTS

- Operating System : Windows 7 , 8, 10 (64 bit)
- Software : Python
- Tools : Python 3.7 IDLE, Spyder

Introduction to Python

Python is a widely-used general-purpose, high-level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

Python is a programming language that lets you work quickly and integrate systems more efficiently.

It is used for:

- web development (server-side),
- software development,
- mathematics,
- System scripting.

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-orientated way or a

functional way

Python Syntax compared to other programming languages

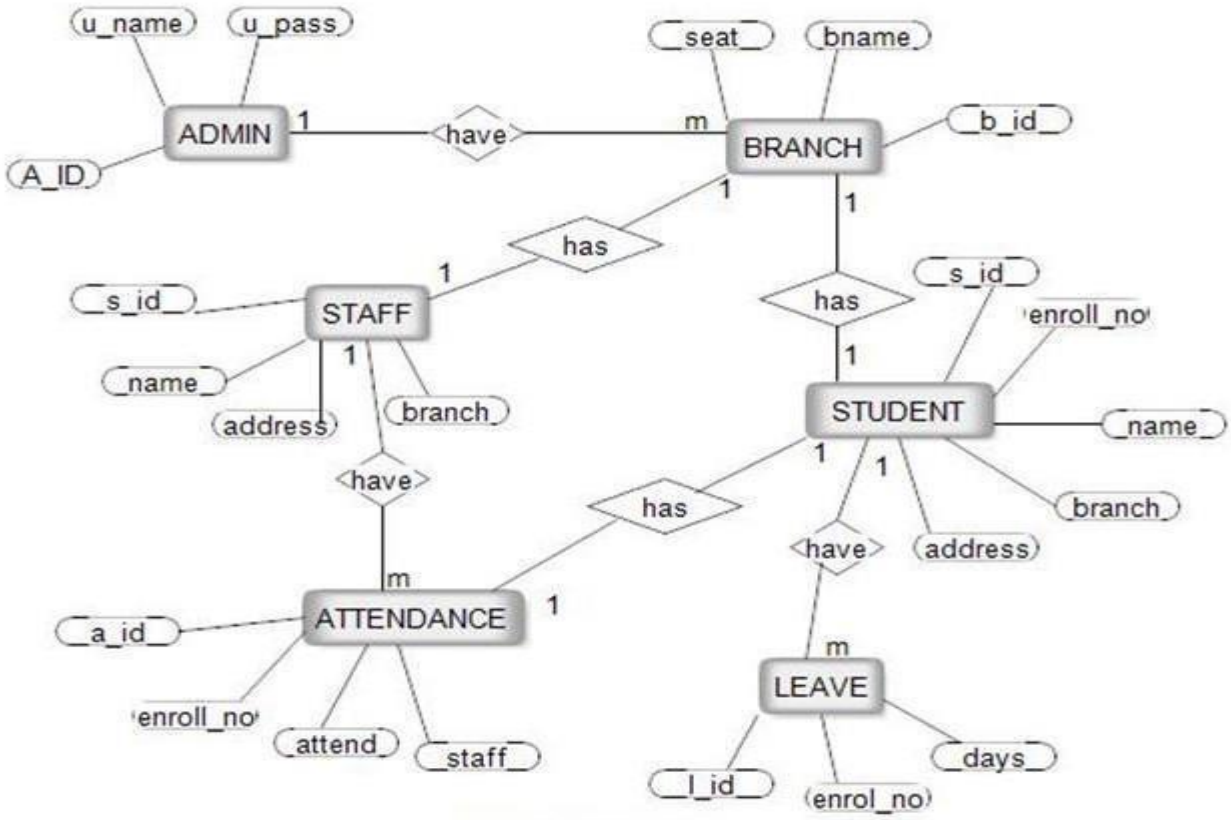
- Python was designed to for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

CHAPTER 4

4. SYSTEM DESIGN

4.1 ER DIAGRAM

An entity-relationship model describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types and specifies relationships that can exist between entities. Entity Relationship Diagram (ERDs) illustrates the logical structure of databases. An entity-relationship (ER) diagram is a specialized graphic that illustrates the interrelationships between entities in a database.



E- R Diagram
Student Attendance Management System

Figure number 4.1.1

4.2 DATA DICTIONARY

Field Name	Data Type	Length	Constraint	Description
Roll_no	Int	3	Primary key	Student roll no
Name	Varchar	20	Not null	Name of student
Date	Date	10	Not null	Date of the attendance
Time	Time	10	Not null	Time of the attendance
Attendance	Varchar	7	Present or Absent	Attendance of a student
Images	.pgm	100	Size must be of 11KB	Images of students

Figure number 4.2.1

4.3 TABLE NORMALIZATIONS:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1			ATTENDANCE FOR THE MONTH OF JANUARY																
2	SL.NO	STUDENT NAME	01/01/2015								02/01/2015								
3			HOUR1	HOUR2	HOUR3	HOUR4	HOUR5	HOUR6	HOUR7	HOUR8	HOUR1	HOUR2	HOUR3	HOUR4	HOUR5	HOUR6	HOUR7	HOUR8	
4	1	MIKE	A	P	P	P	P	P	P	P	A	A	A	A	A	P	P	P	
5	2	JOHN	A	A	A	A	A	A	A	A	P	P	P	P	P	P	P	P	
6	3	SAM	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	
7	4	BILL	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
8	5	SMITH	P	P	P	P	A	A	A	P	A	P	P	P	P	P	P	P	
9																			

Figure 4.3.1

4.4 UML DIAGRAMS

Use Case Diagram:

Unified Modeling Language (UML) is a standardized general-purpose modeling language in the field of software engineering. The standard is managed and was created by the Object Management Group. UML includes a set of graphic notation techniques to create visual models of software intensive systems. This language is used to specify, visualize, modify, construct and document the artifacts of an object oriented software intensive system under development.

A Use case Diagram is used to present a graphical overview of the functionality provided by a system in terms of actors, their goals and any dependencies between those use cases.

Use case diagram consists of two parts:

Use case: A use case describes a sequence of actions that provided something of measurable value to an actor and is drawn as a horizontal ellipse.

Actor: An actor is a person, organization or external system that plays a role in one or system.

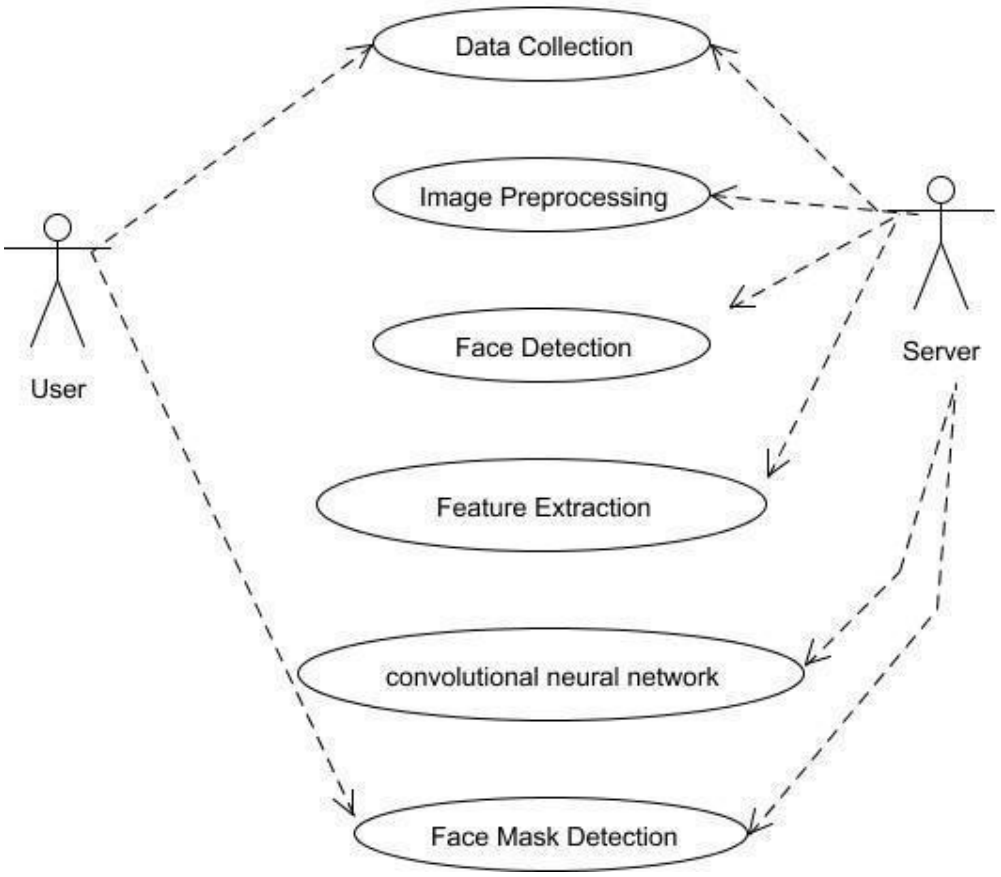


Figure number 4.4.1

Sequence Diagram:

A Sequence diagram is a kind of interaction diagram that shows how processes operate

with one another and in what order. It is a construct of Message Sequence diagrams are sometimes called event diagrams, event sceneries and timing diagram.

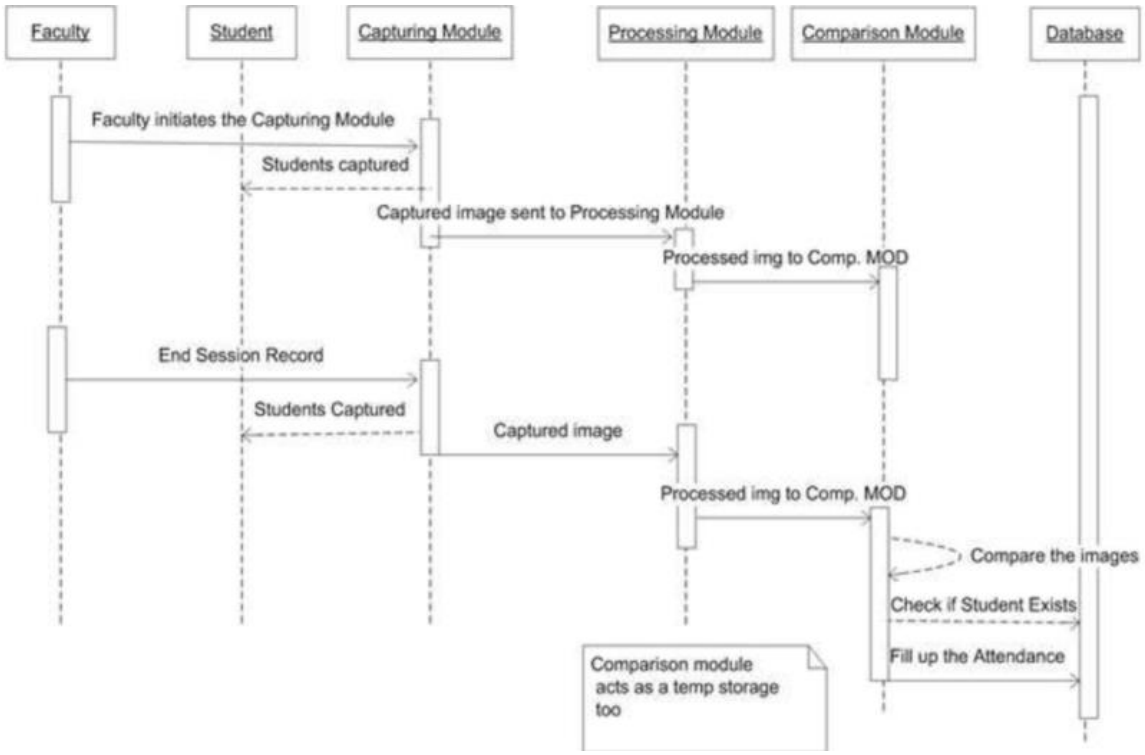


Figure number 4.4.2

Activity Diagram:

Activity diagram is a graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. An activity diagram shows the overall flow of control.

The most important shape types:

- Rounded rectangles represent activities.
- Diamonds represent decisions.
- Bars represent the start or end of concurrent activities.
- A black circle represents the start of the workflow.

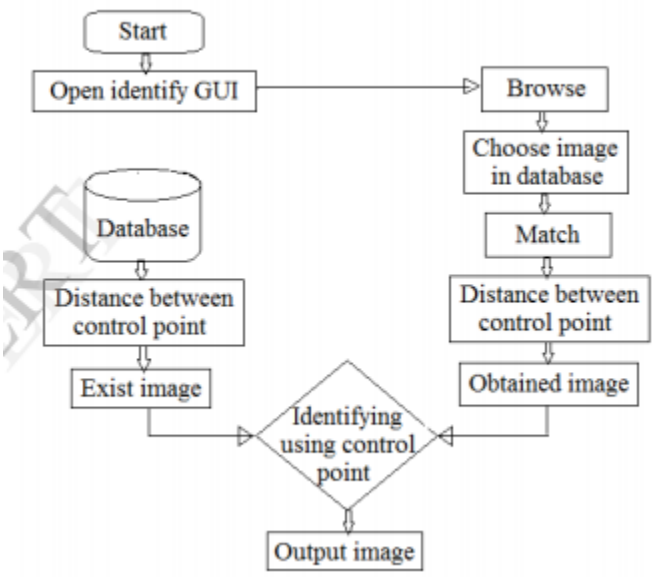


Figure number 4.4.3

Collaboration Diagram:

UML Collaboration Diagrams illustrate the relationship and interaction between software objects. They require use cases, system operation contracts and domain model to already exist. The collaboration diagram illustrates messages being sent between classes and objects

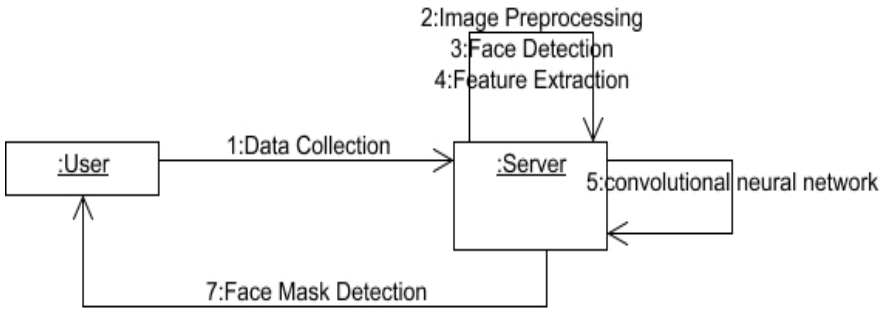


Figure number 4.4.4

Data flow diagram:

A data-flow diagram is a way of representing a flow of data through a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart

LEVEL 0:

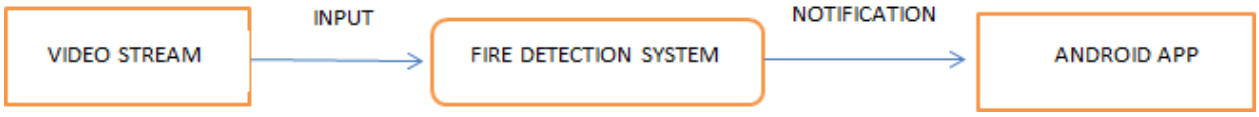


Figure number 4.4.5

CHAPTER 5

5. SYSTEM ARCHITECTURE

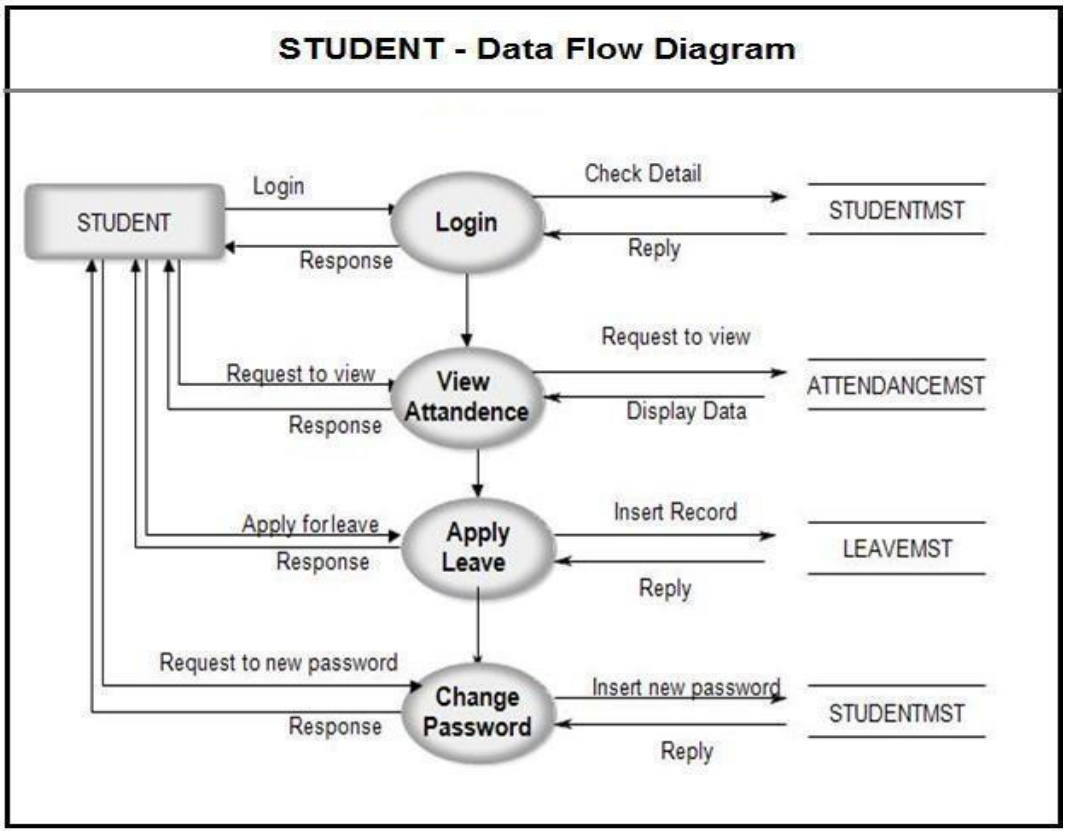


Figure 5.1

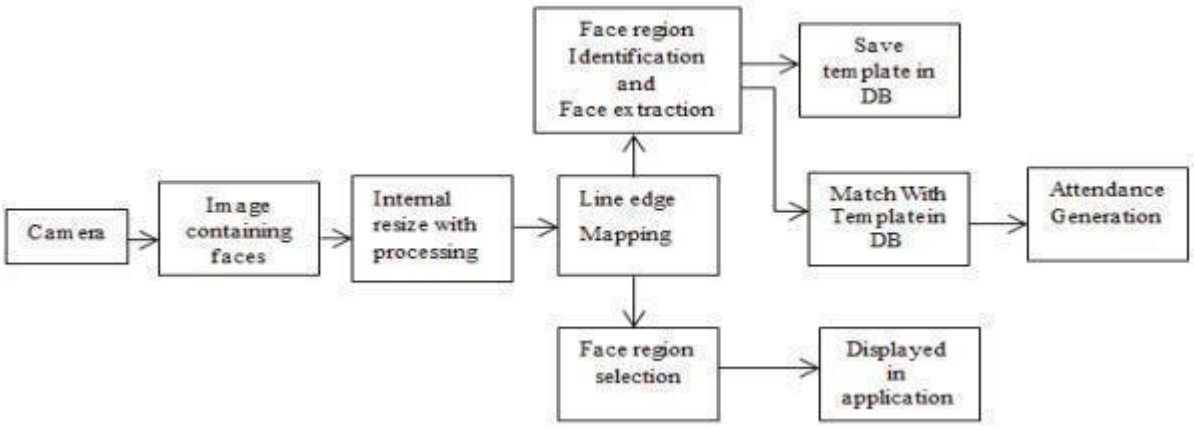


Figure 5.2

5.1 Architecture Overview:

Details of the Proposed Architecture for automatic attendance system

In the initiative image is captured from the camera. There are illumination effects in the captured image due to different lighting conditions and a few noise which is to be removed before going to the next steps. Median filter is employed for removal of noise within the image. There are other techniques like FFT and low pass filter for noise removal and smoothing of the images but median filter gives good results.

Let's see about the modules work clearly.

5.2 Module Design Specification:

In the initiative image is captured from the camera. There are illumination effects in the captured image due to different lighting conditions and a few noise which is to be removed before going to the next steps. Median filter is employed for removal of noise within the image. There are other techniques like FFT and low pass filter for noise removal and smoothing of the images but median filter gives good results.

* **IMAGE ACQUISITION**
* **PRE-PROCESSING IMAGE**
* **FACE DETECTION**
* **FACE RECOGNITION**

These steps of process are takes place during the face recognition. Let's see about the modules work clearly.

IMAGE ACQUISITION:

Image acquisition are often accomplished by digitally scanning an existing photograph or by using an electro-optical camera to accumulate a live picture of a topic . Video also can be used as a source of facial images. The most existing face

recognition systems contain one camera.

The recognition rate is comparatively low when face images are of varied pose and expression and different illumination. With increasing of the pose angle, the popularity rate decreases. The recognition rate decreases greatly when the pose angle is larger than 30 degrees. Different illumination isn't a drag for a few algorithms like LDA which will still recognize faces with different illumination, but this is often not true for PCA. To overcome this problem, we will generate the face images with frontal view (or little rotation), moderate countenance, and same illumination if PCA algorithm is getting used.

PRE- PROCESSING IMAGE:

The system captures around 50 images of every individuals face. The images are converted into grey scale as LBPH operates using images in grey scale and the images are stored in a folder. The stored images will be saved with a name and ID unique to that person. In Image pre-processing HAAR CASCADE algorithm was used to train the image. HAAR CASCADE features are more useful in image processing.

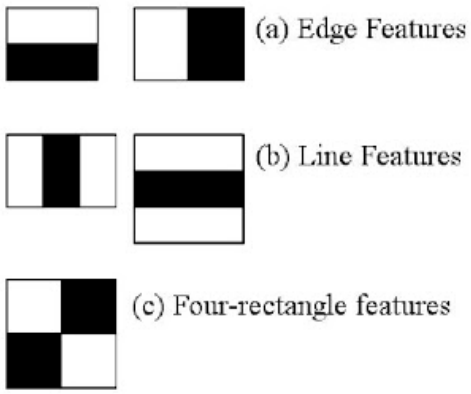


Figure 5.2.1

FACE RECOGNITION:

Facial recognition may be a way of recognizing a person's face through technology. A face recognition system uses biometrics to map countenance from a photograph or video. It compares the knowledge with a database of known faces to seek out a match. Facial recognition can help verify identity, but it also raises privacy issues.

Biometrics can add safety and convenience, but what about the question of knowledge security? But that's where it gets complicated. certainly want some control over how your personal information your data is employed. And here's the thing: your "face print" is data.

You might be good at recognizing faces. You probably find it a cinch to spot the face of a friends and classmates. You're conversant in their countenance — their eyes, nose, mouth — and the way they are available together.

That's how a face recognition system works, but on a grand, algorithmic scale. Where you see a face, recognition technology sees data. That data can be stored and accessed.

FACE DETECTION:

Face detection may be a technology getting used during a sort of applications that identifies human faces in digital images. Face detection also refers to the psychological process by which humans locate and attend to faces during a visual scene.

Face detection are often considered a selected case of object-class detection. In

object-class detection, the task is to seek out the locations and sizes of all objects in a picture that belong to a given class. Examples include upper torsos, pedestrians, and cars.

5.3 Program Design

Algorithm used:

LOCAL BINARY PATTERN HISTOGRAM (LBPH)

Introduction:

LBPH stands for Local Binary Pattern Histogram, a basic algorithm that's used to detect faces from the front side. Local Binary Patterns is a type of visual descriptor used for classification in computer vision. The LBP operator helps to get local features by Local Binary Pattern acts. The LBP operator will divide the face which was shown in the camera into pixels. It will divide the examined window into cells. Concatenate (normalized) histograms of all cells. This gives a feature vector for the entire window.

Applying the LBP operation:

The first computational step of the LBPH is to make an intermediate image that describes the first image during a better way, by highlighting the facial characteristics. To do so, the algorithm uses an idea of a window, supported the parameters radius and neighbors.

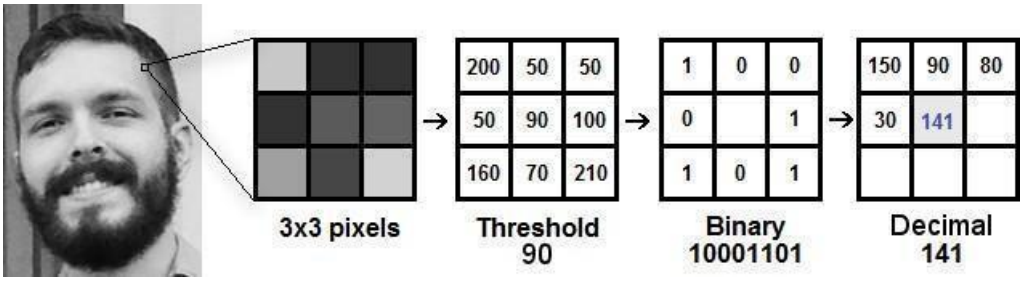


Figure 5.3.1

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

- Suppose we have a facial image in gray scale.
- We can get part of this image as a window of 3x3pixels.
- It also can be represented as a 3x3 matrix containing the intensity of every pixel(0~255).
- Then, we'd like to require the central value of the matrix to be used because the threshold.
- This value will be used to define the new values from the 8 neighbors.
- For each neighbor of the central value (threshold), we set a replacement binary value.

We set 1 for values equal or above the edge and 0 for values less than the edge.

- Now, the matrix will contain only binary values (ignoring the central value). We need to concatenate each binary value from each position from the matrix line by line into a replacement.
- Then,we convert this binary value to a decimal value and set it to the central value of the matrix, which is really a pixel from the first image.
- At the top of this procedure (LBP procedure), we've a replacement image which represents better the characteristics of the first image.

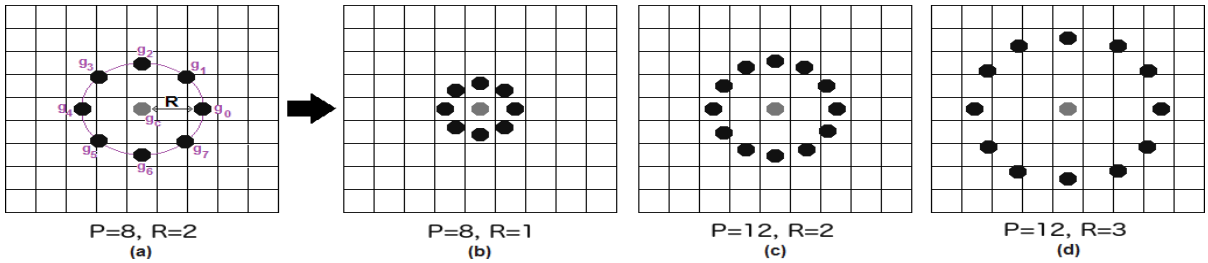


Figure 5.3.2

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

Extracting the Histograms:

Now, using the image generated within the last step, we will use the Grid X and Grid Y parameters to divide the image into multiple grids, as are often seen in th

following image:

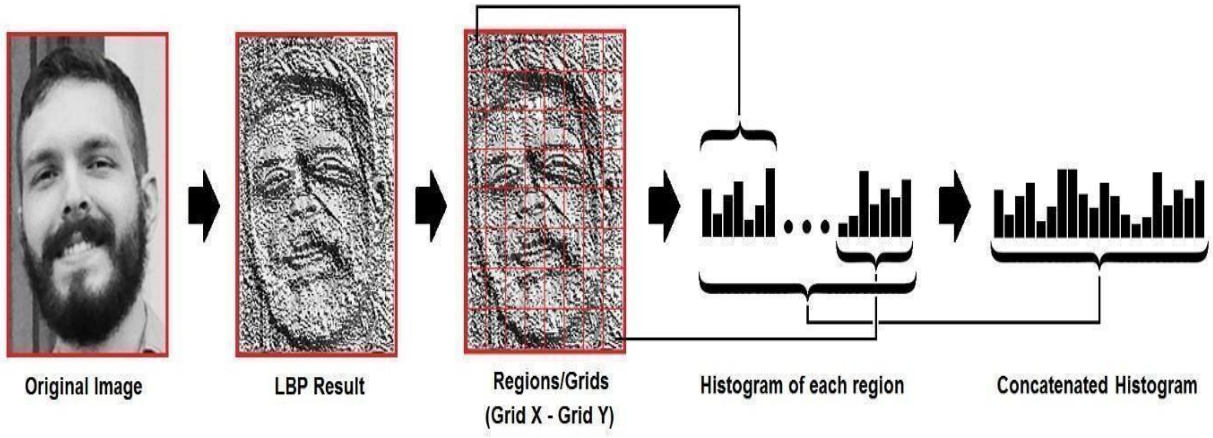


Figure 5.3.3

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

- As we've a picture in gray scale, each histogram (from each grid) will contain only 256 positions (0~255) representing the occurrences of every pixelintensity.
- Then,we'd like to concatenate each histogram to make are placement and larger histogram. Supposing we have 8x8 grids, we will have 8x8x256=16.384 positions in the final histogram. The final histogram represents the characteristics of the original image.

CHAPTER 6

SYSTEM IMPLEMENTATION

6.1 CLIENT-SIDE CODING:

<?xml version="1.0"?>
<opencv_storage>

<cascadetype_id="**opencv-cascade classifier**">

<stageType>BOOST</stageType><featureType>HAAR</featureTy

pe><height>24</height><width>24</width><stageParams><maxWe

akCount>211</maxWeakCount>

</stageParams><featureParams>

<maxCatCount>0</maxCatCount>

</featureParams>

<stageNum>25</stageNum>

<stages><_><maxWeakCount>9</maxWeakCount>

<stageThreshold>- 5.0425500869750977e+00</stageThreshold><weakClassifiers><_><

internalNodes> 0 -1 0 -3.1511999666690826e-02</internalNodes>

<leafValues> 2.0875380039215088e+00 -

2.2172100543975830e+00</leafValues>

</_><_><internalNodes> 0 -1 1 1.2396000325679779e-

02</internalNodes>

<leafValues>-1.8633940219879150e+00

1.3272049427032471e+00</leafValues>

</_><_><internalNodes> 0 -1 2 2.1927999332547188e-

02</internalNodes>

<leafValues>- 1.5105249881744385e+001.0625729560852051e+00</leafValues>

akCount>211</maxWeakCount>
</stageParams><featureParams>
<maxCatCount>0</maxCatCount>
</featureParams>
<stageNum>25</stageNum>
<stages><_><maxWeakCount>9</maxWeakCount>
<stageThreshold>- 5.0425500869750977e+00</stageThreshold><weakClassifiers><_><
internalNodes> 0 -1 0 -3.1511999666690826e-02</internalNodes>
<leafValues> 2.0875380039215088e+00 -
2.2172100543975830e+00</leafValues>
</_><_><internalNodes> 0 -1 1 1.2396000325679779e-
02</internalNodes>
<leafValues>-1.8633940219879150e+00
1.3272049427032471e+00</leafValues>
</_><_><internalNodes> 0 -1 2 2.1927999332547188e-
02</internalNodes>
<leafValues>- 1.5105249881744385e+001.0625729560852051e+00</leafValues>
</_><_><internalNodes> 0 -1 3 5.7529998011887074e-
03</internalNodes>
<leafValues>- 8.7463897466659546e011.1760339736938477e+00</leafValues>
</_><_><internalNodes> 0 -1 4 1.5014000236988068e-
02</internalNodes>
<leafValues>-7.7945697307586670e-
1.2608419656753540e+00</leafValues>
</_><_><internalNodes> 0 -1 5 9.9371001124382019e-
02</internalNodes>
<leafValues>5.5751299858093262e-

011.8743000030517578e+00</leafValues>
</_><_>

<internalNodes> 0 -1 6 2.7340000960975885e- 03</internalNodes><leafValues> -
1.6911929845809937e+00 4.4009700417518616e-01</leafValues>
</_><_>

<internalNodes> 0 -1 7 -1.8859000876545906e-02</internalNodes>
<leafValues> -1.4769539833068848e+00 4.4350099563598633e-
01</leafValues>
</_><_>
<internalNodes> 0 -1 8 5.9739998541772366e
8.5255599021911621e-
01</leafValues></_></weakClassifiers></_><_><maxWeakCount>1
6</maxWeakCount><stageThreshold>-
4.9842400550842285e+00</stageThreshold><weakClassifiers><_>< internalNodes> 0
-1 9 -2.1110000088810921e-
02</internalNodes><leafValues> 1.2435649633407593e+00 -
1.5713009834289551e+00</leafValues></_><_><internalNodes> 0
-1 10 2.0355999469757080e-02</internalNodes><leafValues> -
1.6204780340194702e+00
1.1817760467529297e+00</leafValues></_><_><internalNodes> 0
-1 11 2.1308999508619308e-02</internalNodes><leafValues> -
1.9415930509567261e+00 7.0069098472595215e-
01</leafValues></_><_><internalNodes> 0 -1 12
9.1660000383853912e-02</internalNodes><leafValues> -
5.5670100450515747e-01
1.7284419536590576e+00</leafValues></_><_><internalNodes> 0
-1 13 3.6288000643253326e-02</internalNodes><leafValues>

2.6763799786567688e-01 -
2.1831810474395752e+00</leafValues></_><_><internalNodes> 0
-1 14 -1.9109999760985374e-02</internalNodes><leafValues> -
2.6730210781097412e+00 4.5670801401138306e-
01</leafValues></_><_><internalNodes> 0 -1 15
8.2539999857544899e-03</internalNodes><leafValues> - 1.0852910280227661e+00
5.3564202785491943e-
01</leafValues></_><_><internalNodes> 0 -1 16
1.8355000764131546e-02</internalNodes><leafValues> - 3.5200199484825134e-01
9.3339198827743530e-
01</leafValues></_><_><internalNodes> 0 -1 17 -
7.0569999516010284e-

6.2 SERVER-SIDE CODING:

FACE RECOGNITION WITH DEEP LEARNING:

```
{
 "cells": [
  {
   "cell_type": "markdown",
   "metadata": {
    "_uuid": "050307096c25e11ef3fd75df86a9477903882246"
   },
   "source": [

    "# Quick data visualization"

   ]
  },
  {
   "cell_type": "code",
   "execution_count": 18,
   "metadata":
  "_uuid": "ada86668a25cba36e2676a20bc76f96581057d26"
   },
   "outputs": [
    {
     "data": {
```

```
        "image/png":
"text/plain": [
         "<Figure size 864x1440 with 16 Axes>"
        ]
       },
       "metadata": { "needs_background":
        "light"
       },
       "output_type": "display_data"
      }
     ],
    "source": [
      "# display some images for every different expression\n",
      "\n",
      "import numpy as np\n", "import
      seaborn as sns\n",
      "from      keras.preprocessing.image           import
load_img, img_to_array\n",
      "import matplotlib.pyplot as plt\n", "import
      os\n",
      "\n",
      "# size of the image: 48*48 pixels\n", "pic_size
      = 48\n",
      "\n"
      "text/plain": [
            "<Figure size 864x1440 with 16 Axes>"
            ]
        },
        "metadata": {
            "needs_background": "light"
        },
        "output_type": "display_data"
        }
         ],
         "source": [
        "# display some images for every different
      expression\n",
        "\n",
        "import numpy as np\n",
        "import seaborn as sns\n",
        "from keras.preprocessing.image import load_img,
      img_to_array\n",
        "import matplotlib.pyplot as plt\n",
```

```
  "import os\n",
  "\n",
  "# size of the image: 48*48 pixels\n",
  "pic_size = 48\n",
  "\n",
  "# input path for the images\n",
  "base_path =
\"C:/Users/user/Downloads/Automated_Attendance_System/ima
ges/\"\n",
  "\n",
  "plt.figure(0, figsize=(12,20))\n",
  "cpt = 0\n",
  "\n",
  "for expression in os.listdir(base_path +
\"train/\"):\n",
  "    for i in range(1,5):\n",
  "        cpt = cpt + 1\n",
  "        plt.subplot(7,4,cpt)\n",
  "        img = load_img(base_path + \"train/\" +
expression + \"/\" +os.listdir(base_path + \"train/\" +
expression)[i], target_size=(pic_size, pic_size))\n",
  "        plt.imshow(img, cmap=\"gray\")\n",
  "\n",
  "plt.tight_layout()\n",
  "plt.show()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 19,
   "metadata": {
  "_uuid": "2fc206564d516020e9c3001c8a1448766aace848"
   },
   "outputs": [
  {
  "name": "stdout",
  "output_type": "stream",
  "text": [
      "55 Jinping images\n",
      "61 Martin images\n",
    "61 modi images\n",
      "61 Trumph images\n"
  ]
  }
```

```
   ],
   "source": [
  "# count number of train images for each expression\n",
  "\n",
  "for expression in os.listdir(base_path +
\"train\"):\n",
  "    print(str(len(os.listdir(base_path + \"train/\" +
expression))) + \" \" + expression + \" images\")"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
  "_uuid": "74f292f501e5a60adb5463f412f42e5584f3a6a5"
   },
   "source": [
  "# Setup the data generators"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 20,
   "metadata": {
  "_uuid": "0041128a27f4da936ad63e90959797391736fc8b"
   },
   "outputs": [
  {
  "name": "stdout",
  "output_type": "stream",
  "text": [
      "Found 238 images belonging to 4 classes.\n",
      "Found 238 images belonging to 4 classes.\n"
  ]
  }
   ],
   "source": [
  "from keras.preprocessing.image import
ImageDataGenerator\n",
  "\n",
  "# number of images to feed into the NN for every
batch\n",
  "batch_size = 128\n",
  "\n",
  "datagen_train = ImageDataGenerator()\n",
```

```
  "datagen_validation = ImageDataGenerator()\n",
  "\n",
  "train_generator =
datagen_train.flow_from_directory(base_path +
\"train\",\n",
  "
target_size=(pic_size,pic_size),\n",
  "
  color_mode=\"grayscale\",\n",
  "
  batch_size=batch_size,\n",
  "
class_mode='categorical',\n",
  "
  shuffle=True)\n",
  "\n",
  "validation_generator =
datagen_validation.flow_from_directory(base_path +
\"validation\",\n",
  "
target_size=(pic_size,pic_size),\n",
  "
  color_mode=\"grayscale\",\n",
  "
  batch_size=batch_size,\n",
  "
class_mode='categorical',\n",
  "
  shuffle=False)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 34,
   "metadata": {},
   "outputs": [
  {
  "data": {
      "text/plain": [
      "{'Jinping': 0, 'Martin': 1, 'Trumph': 2, 'modi':
3}"
      ]
  },
  "execution_count": 34,
```

```
  "metadata": {},
  "output_type": "execute_result"
  }
   ],
   "source": [
  "train_generator.class_indices"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 22,
   "metadata": {
  "_uuid": "4914afe556e0db3c03614ed1ac2138fd68d03815"
   },
   "outputs": [],
   "source": [
  "from keras.layers import Dense, Input, Dropout,
GlobalAveragePooling2D, Flatten, Conv2D,
BatchNormalization, Activation, MaxPooling2D\n",
  "from keras.models import Model, Sequential\n",
  "from keras.optimizers import Adam\n",
  "\n",
  "# number of possible label values\n",
  "nb_classes = 4\n",
  "\n",
  "# Initialising the CNN\n",
  "model = Sequential()\n",
  "\n",
  "# 1 - Convolution\n",
  "model.add(Conv2D(64,(3,3), padding='same',
input_shape=(48, 48,1)))\n",
  "model.add(BatchNormalization())\n",
  "model.add(Activation('relu'))\n",
  "model.add(MaxPooling2D(pool_size=(2, 2)))\n",
  "model.add(Dropout(0.25))\n",
  "\n",
  "# 2nd Convolution layer\n",
  "model.add(Conv2D(128,(5,5), padding='same'))\n",
  "model.add(BatchNormalization())\n",
  "model.add(Activation('relu'))\n",
  "model.add(MaxPooling2D(pool_size=(2, 2)))\n",
  "model.add(Dropout(0.25))\n",
  "\n",
  "# 3rd Convolution layer\n",
```

```
   "model.add(Conv2D(512,(3,3), padding='same'))\n",
   "model.add(BatchNormalization())\n",
   "model.add(Activation('relu'))\n",
   "model.add(MaxPooling2D(pool_size=(2, 2)))\n",
   "model.add(Dropout(0.25))\n",
   "\n",
   "# 4th Convolution layer\n",
   "model.add(Conv2D(512,(3,3), padding='same'))\n",
   "model.add(BatchNormalization())\n",
   "model.add(Activation('relu'))\n",
   "model.add(MaxPooling2D(pool_size=(2, 2)))\n",
   "model.add(Dropout(0.25))\n",
   "\n",
   "# Flattening\n",
   "model.add(Flatten())\n",
   "\n",
   "# Fully connected layer 1st layer\n",
   "model.add(Dense(256))\n",
   "model.add(BatchNormalization())\n",
   "model.add(Activation('relu'))\n",
   "model.add(Dropout(0.25))\n",
   "\n",
   "# Fully connected layer 2nd layer\n",
   "model.add(Dense(512))\n",
   "model.add(BatchNormalization())\n",
   "model.add(Activation('relu'))\n",
   "model.add(Dropout(0.25))\n",
   "\n",
   "model.add(Dense(nb_classes, activation='softmax'))\n",
   "\n",
   "opt = Adam(lr=0.0001)\n",
   "model.compile(optimizer=opt,
loss='categorical_crossentropy', metrics=['accuracy'])"
    ]
   },
   {
    "cell_type": "markdown",
    "metadata": {
   "_uuid": "883ca4ab4d9c00ac08434916e7713c8f9b4353ba"
    },
    "source": [
   "# Train the model"
    ]
   },
```

```
  {
   "cell_type": "code",
   "execution_count": 23,
   "metadata": {
  "_uuid": "8d3db222ae6b74ab55df0d840d372b4306af6db0",
  "scrolled": true
   },
   "outputs": [
  {
  "name": "stdout",
  "output_type": "stream",
  "text": [
      "WARNING:tensorflow:From
C:\\Users\\user\\Anaconda3\\lib\\site-
packages\\tensorflow\\python\\ops\\math_grad.py:1250:
add_dispatch_support.<locals>.wrapper (from
tensorflow.python.ops.array_ops) is deprecated and will
be removed in a future version.\n",
      "Instructions for updating:\n",
      "Use tf.where in 2.0, which has the same broadcast
rule as np.where\n",
      "Epoch 1/50\n",
      "1/1 [==============================] - 20s
20s/step - loss: 1.8261 - acc: 0.2273 - val_loss: 1.6503
- val_acc: 0.3750\n",
      "\n",
      "Epoch 00001: val_acc improved from -inf to
0.37500, saving model to
C:/Users/user/Downloads/Automated_Attendance_System/model
_weights.h5\n",
      "Epoch 2/50\n",
      "1/1 [==============================] - 10s
10s/step - loss: 1.4742 - acc: 0.4297 - val_loss: 1.1339
- val_acc: 0.5455\n",
      "\n",
      "Epoch 00002: val_acc improved from 0.37500 to
0.54545, saving model to
C:/Users/user/Downloads/Automated_Attendance_System/model
_weights.h5\n",
      "Epoch 3/50\n",
      "1/1 [==============================] - 9s 9s/step
- loss: 1.1998 - acc: 0.4727 - val_loss: 1.1915 -
val_acc: 0.4688\n",
      "\n",
```

```
      "Epoch 00003: val_acc did not improve from
0.54545\n",
      "Epoch 4/50\n",
      "1/1 [==============================] - 10s
10s/step - loss: 0.9966 - acc: 0.5781 - val_loss: 0.7633
- val_acc: 0.8182\n",
      "\n",
      "Epoch 00004: val_acc improved from 0.54545 to
0.81818, saving model to
C:/Users/user/Downloads/Automated_Attendance_System/model
_weights.h5\n",
      "Epoch 5/50\n",
      "1/1 [==============================] - 9s 9s/step
- loss: 0.8544 - acc: 0.6909 - val_loss: 0.7948 -
val_acc: 0.8438\n",
      "\n",
      "Epoch 00005: val_acc improved from 0.81818 to
0.84375, saving model to
C:/Users/user/Downloads/Automated_Attendance_System/model
_weights.h5\n",
      "Epoch 6/50\n",
      "1/1 [==============================] - 9s 9s/step
- loss: 0.6221 - acc: 0.7656 - val_loss: 0.4888 -
val_acc: 1.0000\n",
      "\n",
      "Epoch 00006: val_acc improved from 0.84375 to
1.00000, saving model to
C:/Users/user/Downloads/Automated_Attendance_System/model
_weights.h5\n",
      "Epoch 7/50\n",
      "1/1 [==============================] - 10s
10s/step - loss: 0.4663 - acc: 0.8672 - val_loss: 0.5009
- val_acc: 0.9766\n",
      "\n",
      "Epoch 00007: val_acc did not improve from
1.00000\n",
      "Epoch 8/50\n",
      "1/1 [==============================] - 8s 8s/step
- loss: 0.4090 - acc: 0.9091 - val_loss: 0.2992 -
val_acc: 1.0000\n",
      "\n",
      "Epoch 00008: val_acc did not improve from
1.00000\n",
      "Epoch 9/50\n",
```

```
      "1/1 [==============================] - 9s 9s/step
- loss: 0.2736 - acc: 0.9455 - val_loss: 0.3057 -
val_acc: 1.0000\n",
      "\n",
      "Epoch 00009: val_acc did not improve from
1.00000\n",
      "Epoch 10/50\n",
      "1/1 [==============================] - 9s 9s/step
- loss: 0.3546 - acc: 0.8984 - val_loss: 0.1756 -
val_acc: 1.0000\n",
      "\n",
      "Epoch 00010: val_acc did not improve from
1.00000\n",
      "Epoch 11/50\n",
      "1/1 [==============================] - 9s 9s/step
- loss: 0.2407 - acc: 0.9636 - val_loss: 0.1912 -
val_acc: 1.0000\n",
      "\n",
      "Epoch 00011: val_acc did not improve from
1.00000\n",
      "Epoch 12/50\n",
      "1/1 [==============================] - 10s
10s/step - loss: 0.1755 - acc: 0.9844 - val_loss: 0.1086
- val_acc: 1.0000\n",
      "\n",
      "Epoch 00012: val_acc did not improve from
1.00000\n",
      "Epoch 13/50\n",
      "1/1 [==============================] - 9s 9s/step
- loss: 0.1702 - acc: 0.9727 - val_loss: 0.1178 -
val_acc: 1.0000\n",
      "\n",
      "Epoch 00013: val_acc did not improve from
1.00000\n",
      "Epoch 14/50\n",
      "1/1 [==============================] - 9s 9s/step
- loss: 0.1329 - acc: 0.9844 - val_loss: 0.0708 -
val_acc: 1.0000\n",
      "\n",
      "Epoch 00014: val_acc did not improve from
1.00000\n",
      "Epoch 15/50\n",
      "1/1 [==============================] - 10s
10s/step - loss: 0.1165 - acc: 0.9922 - val_loss: 0.0771
```

CHAPTER 7

SYSTEM TESTING

Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. The goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself. For example the design must not have any logic faults in the design is detected before coding commences, otherwise the cost of fixing the faults will be considerably higher as reflected. Detection of design faults can be achieved by means of inspection as well as walkthrough.

Testing is one of the important steps in the software development phase. Testing checks for the errors, as a whole of the project testing involves the following test cases:

- Static analysis is used to investigate the structural properties of the Source code.
- Dynamic testing is used to investigate the behavior of the source code by executing the program on the test data.

7.1 UNIT TESTING

Unit testing is conducted to verify the functional performance of each modular component of the software. Unit testing focuses on the smallest unit of the software design (i.e.), the module. The white-box testing techniques were heavily employed for unit testing.

7.2 INTEGRATION TESTING

Integration testing is a systematic technique for construction the program structure while at the same time conducting tests to uncover errors associated with interfacing. i.e., integration testing is the complete testing of the set of modules which makes up the product. The objective is to take untested modules and build a program structure tester should identify critical modules. Critical modules should be tested as early as possible. One

approach is to wait until all the units have passed testing, and then combine them and then tested. This approach is evolved from unstructured testing of small programs. Another strategy is to construct the product in increments of tested units. A small set of modules are integrated together and tested, to which another module is added and tested in combination. And so on. The advantages of this approach are that, interface dispenses can be easily found and corrected.

The major error that was faced during the project is linking error. When all the modules are combined the link is not set properly with all support files. Then we checked out for interconnection and the links. Errors are localized to the new module and its intercommunications. The product development can be staged, and modules integrated in as they complete unit testing. Testing is completed when the last module is integrated and tested.

7.3 TEST CASES REPORTS/PERFORMANCE ANALYSIS

S.NO	ACTION	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	TEST RESULT	TEST COMMENTS
1.	Login	username,password	homepage	homepage	pass	authorized user is logged in system

TEST CASE 2

S.NO	ACTION	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	TEST RESULT	TEST COMMENTS

2.	Registration	student roll no,name	New student details are added	New student details are added	Pass	Details of new student is added in database

TEST CASE 3

S.NO	ACTION	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	TEST RESULT	TEST COMMENTS
3.	Face recognition	Student face	Display student name,id	Display student name,id	pass	It will detect the face with trained images

TEST CASE 4

S.NO	ACTION	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	TEST RESULT	TEST COMMENTS
4.	Marking absent or present in excel sheet	Student face	Present	Absent	Fails	The student is absent in excel sheet.If student did not attend class or may be a new student who did not register his details.

CHAPTER 8

8.1 CONCLUSION AND FUTURE ENHANCEMENT

We have implemented an attendance management system for student's attendance. It helps to reduce time and effort, especially in the case of a large number of students marked attendance. The whole system is implemented in the Python programming language. Facial recognition techniques used for the purpose of student attendance. And also, this record of student attendance can further be used mainly in exam related issues like who are attending the exams and who are not attending. On this project, there is some further work remaining to do like installing the system in the classrooms. It can be constructed using a camera and computer.

APPENDICES

A.1 SAMPLE SCREENS

APPENDICES:

DATABASE STUDENT DETAILS:

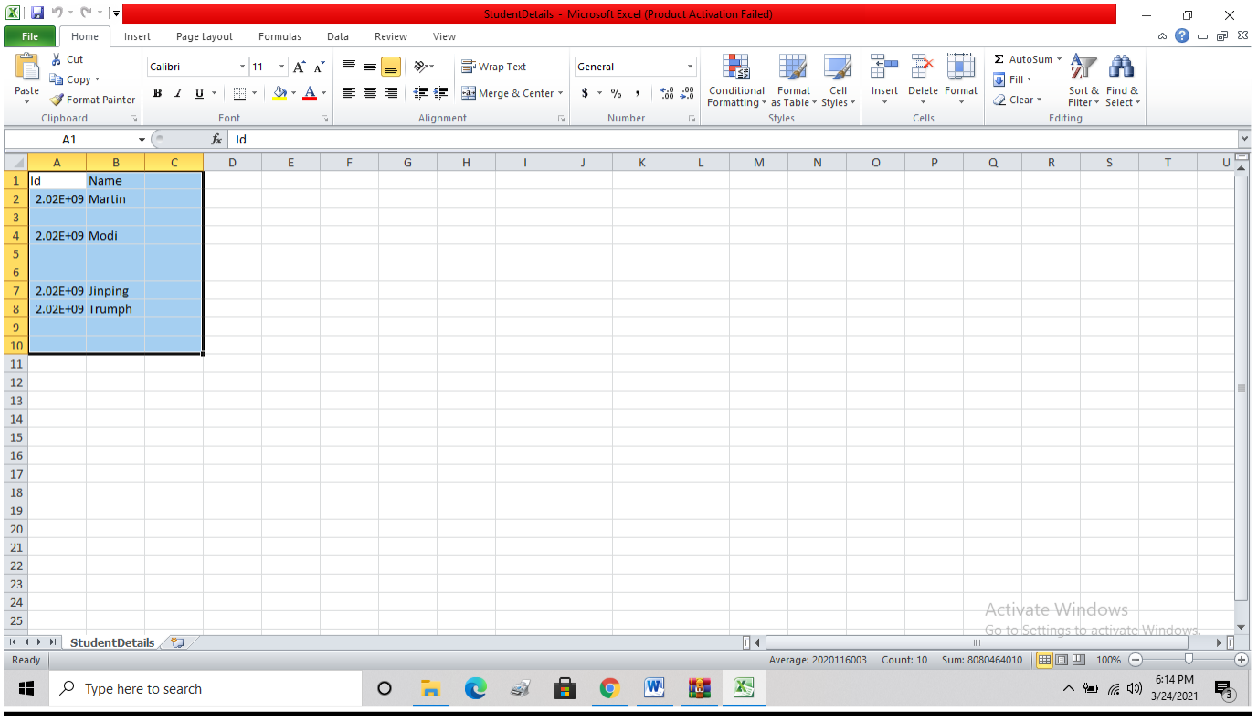


Figure A.1.1

OUTPUT SCREENSHOT:

STUDENT ATTENDANCE

SYSTEM:

APPLICATION:

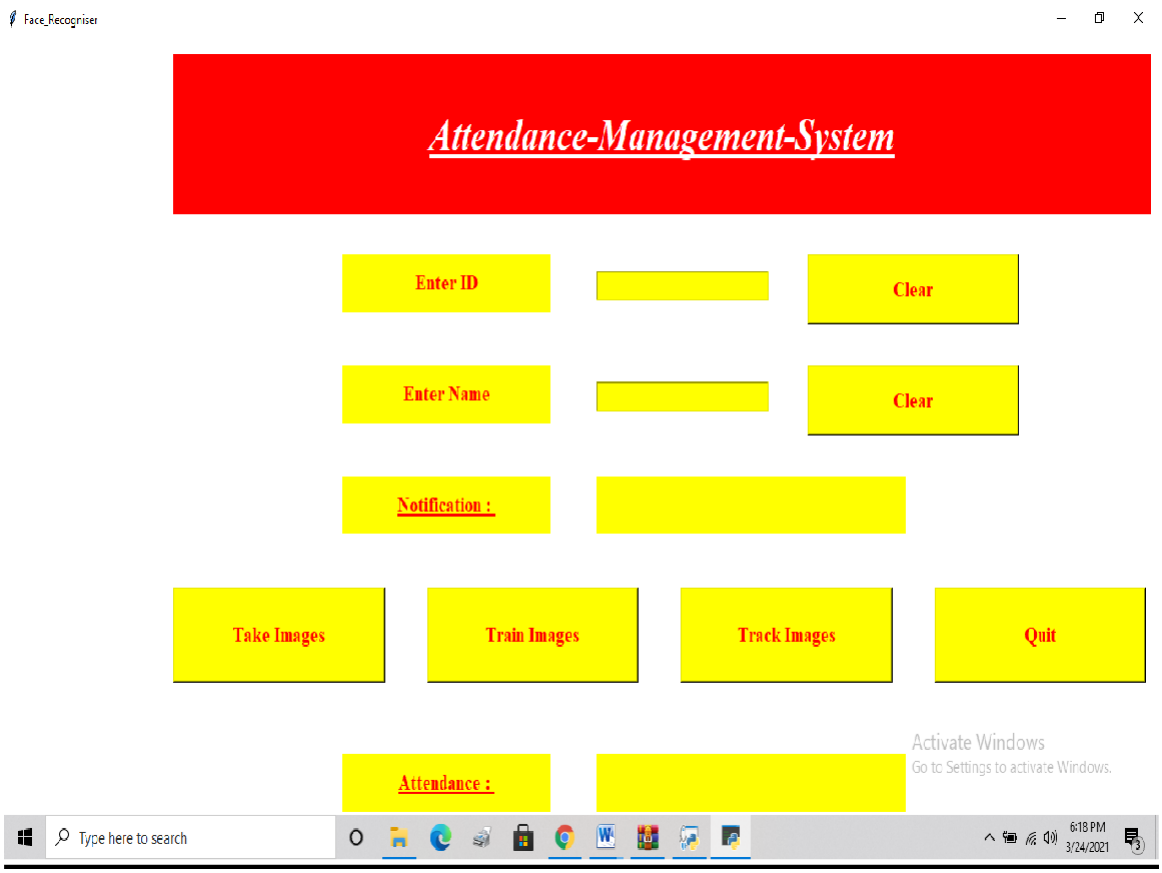


Figure A.1.2

TRAINING IMAGE:

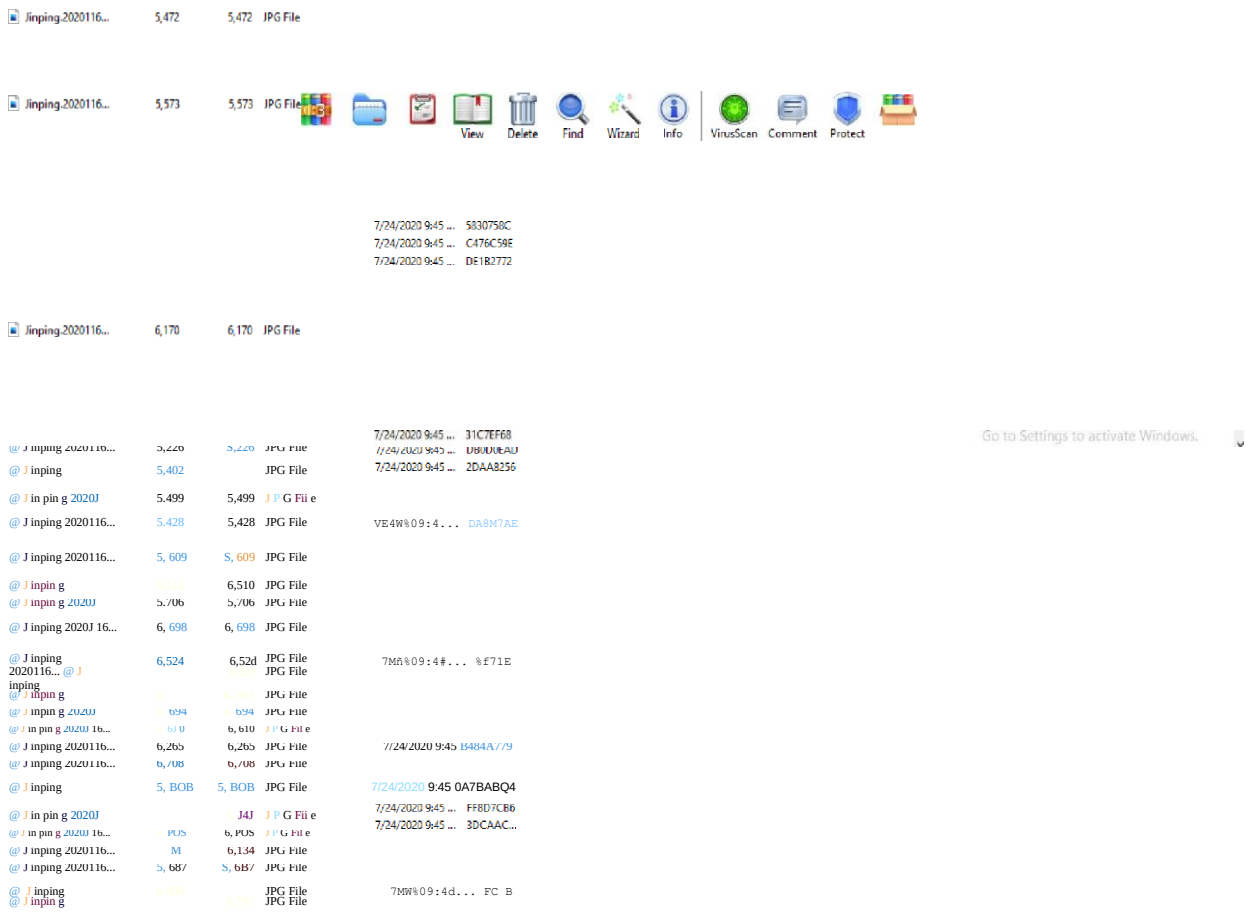


Figure A.1.3

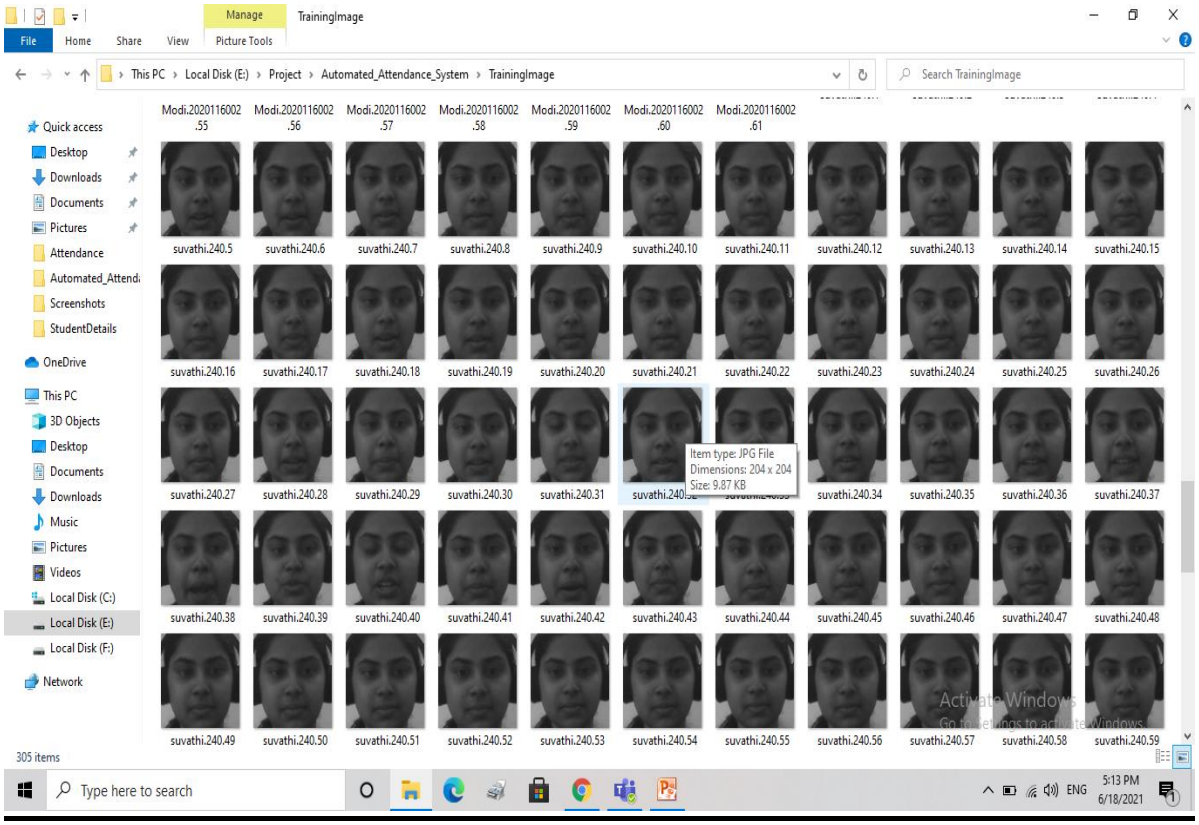

TrainingImage
Manage
File
Home
Share
View
Picture Tools
This PC › Local Disk (E:) › Project › Automated_Attendance_System › TrainingImage
Search TrainingImage
Quick access
Desktop
Downloads
Documents
Pictures
Attendance
Automated_Attenda
Screenshots
StudentDetails
OneDrive
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
Local Disk (E:)
Local Disk (F:)
Network
Modi.2020116002.55
Modi.2020116002.56
Modi.2020116002.57
Modi.2020116002.58
Modi.2020116002.59
Modi.2020116002.60
Modi.2020116002.61
suvathi.240.5
suvathi.240.6
suvathi.240.7
suvathi.240.8
suvathi.240.9
suvathi.240.10
suvathi.240.11
suvathi.240.12
suvathi.240.13
suvathi.240.14
suvathi.240.15
suvathi.240.16
suvathi.240.17
suvathi.240.18
suvathi.240.19
suvathi.240.20
suvathi.240.21
suvathi.240.22
suvathi.240.23
suvathi.240.24
suvathi.240.25
suvathi.240.26
suvathi.240.27
suvathi.240.28
suvathi.240.29
suvathi.240.30
suvathi.240.31
Item type: JPG File
Dimensions: 204 x 204
Size: 9.87 KB
suvathi.240.34
suvathi.240.35
suvathi.240.36
suvathi.240.37
suvathi.240.38
suvathi.240.39
suvathi.240.40
suvathi.240.41
suvathi.240.42
suvathi.240.43
suvathi.240.44
suvathi.240.45
suvathi.240.46
suvathi.240.47
suvathi.240.48
suvathi.240.49
suvathi.240.50
suvathi.240.51
suvathi.240.52
suvathi.240.53
suvathi.240.54
suvathi.240.55
suvathi.240.56
suvathi.240.57
suvathi.240.58
suvathi.240.59
Activate Windows
305 items
Type here to search
ENG
5:13 PM
6/18/2021

Figure A.1.4

TRAINED IMAGE:

Figure A.1.5

ATTENDANCE VERIFICATION:

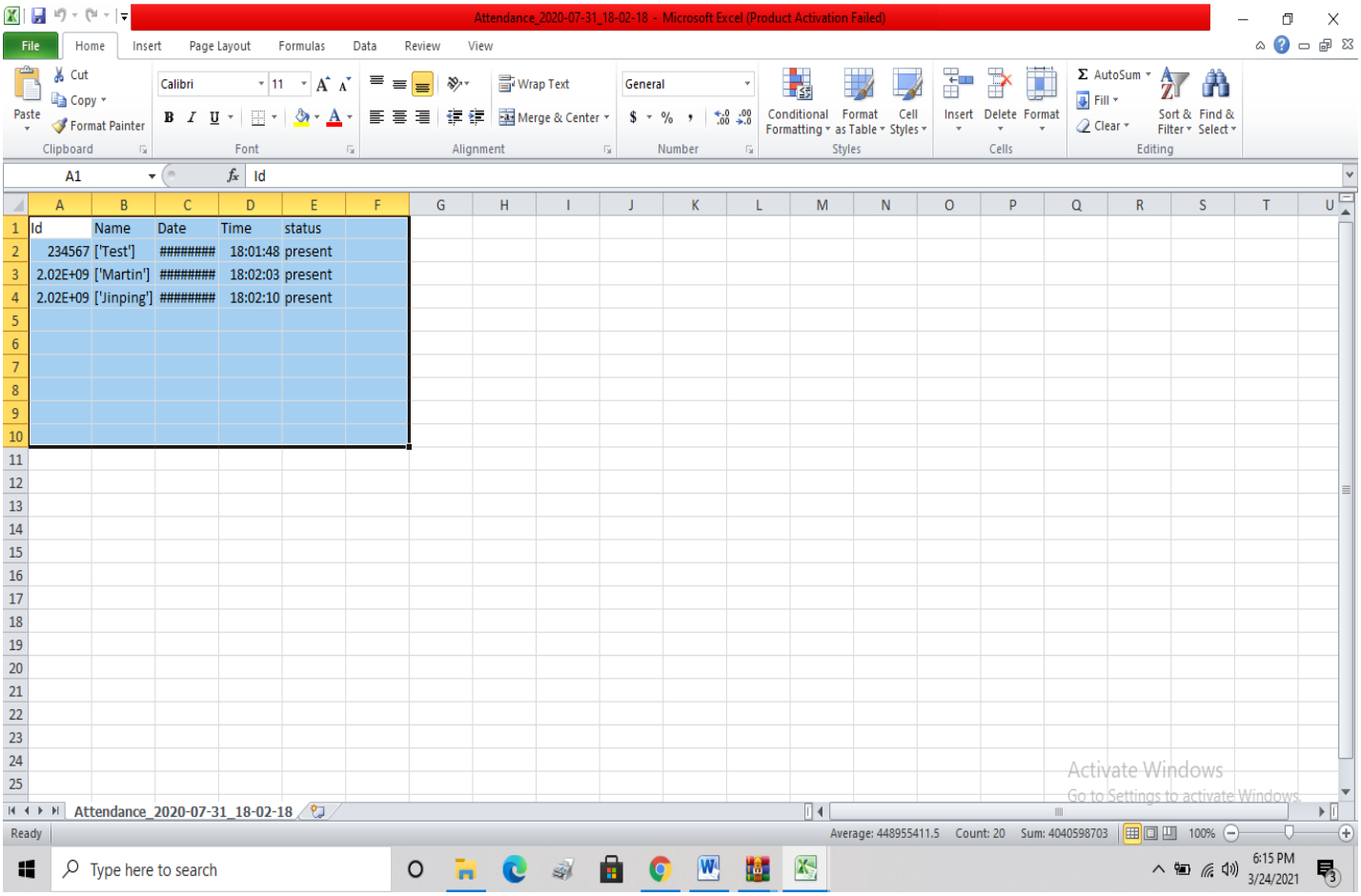


Figure A.1.6

PUBLICATION CERTIFICATE:

Certificate of Publication

IJRES Journal

This is to confirm that

S.Umamaheswari

Published following article

Automatic Attendance System Based On Face Recognition and Face Detection

Volume 9, Issue 5, pp: 20-23

www.ijres.org

A Peer Reviewed referred Journal

International Journal of Research in Engineering and Science (IJRES)

ISSN: 2320-9364

Editor-In-Chief

Automatic Attendance System Based On Face Recognition and Face Detection

R.Devi[1], B.Srivasanthi[2], T.Suvathi[3] ,S.Umamaheswari[4]
Assistant Professor (grade 1)[1], UG Scholar[2,3,4]
[1,2,3,4]Dept. of CSE, Panimalar Engineering College, Chennai

ABSTRACT:
In the traditional system, it is hard to be handle the attendance of huge students in a classroom. As it is time-consuming and has a high probability of error during the process of inputting data into the computer. Real-Time Face Recognition is a real-world solution which comes with day to day activities of handling a bulk of student's attendance. Face Recognition is a process of recognizing the students face for taking attendance by using face biometrics. In this project, a computer system will be able to find and recognize human faces fast that are being captured through a surveillance camera. Numerous algorithms and techniques have been developed for improving the performance of face recognition but our proposed system uses Haar cascade classifier to find the positive and negative of the face and LBPH (Local binary pattern histogram) algorithm for face recognition by using python programming and OpenCV library. Here we use the tkinter GUI interface for user interface purpose.

KEYWORDS: - *Haar cascade classifier, LBPH algoritham*

Date of Submission: 27-04-2021 Date of acceptance: 11-05-2021

I. INTRODUCTION :

The technology aims in imparting tremendous knowledge oriented technical innovations these days. Machine Learning is one among the interesting domain that enables the machine to train itself by providing some datasets as input and provides an appropriate output during testing by applying different learning algorithms. Nowadays Attendance is considered as an important factor for both the student and the teacher of an educational organization. With the advancement of the Machine learning technology the machine automatically detects the attendance performance of the students and maintains a record of those collected data. In general, the attendance system of the student can be maintained in two, different forms namely, Manual Attendance System (MAS) Automated Attendance System (AAS). Manual Student Attendance Management system is a process where a teacher concerned with the particular subject need to call the students name and mark the attendance manually. Manual attendance may be considered as a time-consuming process or sometimes it happens for the teacher to miss someone, or students may answer multiple times on the absence of their friends. So, the problem arises when we think about the traditional process of taking attendance in the classroom. To solve all these issues, we go with Automatic Attendance System (AAS). There are so many advantages using this technology. Some of them are as follows - Automation simplifies time tracking, and there is no need to have personnel to monitor the system 24 hours a day. With automated systems, human error is eliminated. - A time and attendance system using facial recognition technology can accurately report attendance, absence, and overtime with an identification process that is fast as well as accurate. - Facial recognition software can accurately track time and attendance without any human error -Facial biometric time tracking allows you to not only track employees but also add visitors to the system so they can be tracked throughout the worksite.

II. LITERATURE SURVEY:

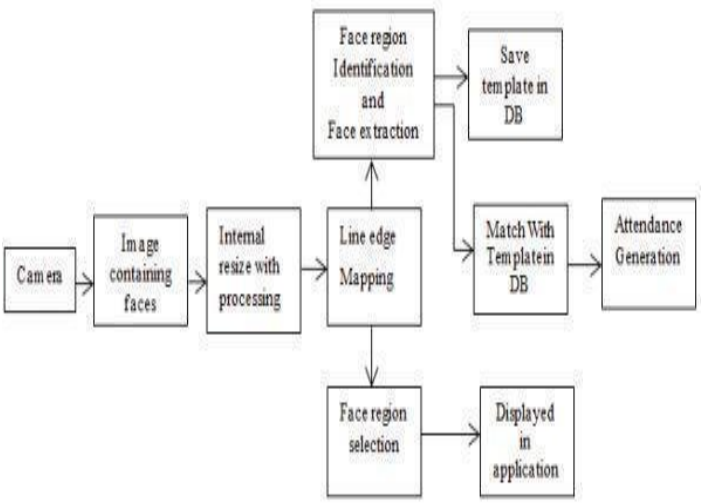
There were many approaches used for dealing with disparity in images subject to illumination changes and these approaches were implemented in object recognition systems and also by systems that were specific to faces. Some of the approaches as follows: - A method for coping with such variations was using gray-level information to extract a face or an object from shading approach [1]. The more reason gray scale representations are used for extracting descriptors instead of operating on color images directly and also gray scale simplifies the algorithm and reduces computational requirements. Here in our case, color is of limited benefit and introducing unnecessary information could increase the number of coaching data required to attain good performance [2]. Being an ill-posed problem, these proposed solutions assumed either the item shape and reluctance properties or the illumination conditions [3]. These assumptions made are too strict for general beholding, and so, it didn't persuade be sufficient for face recognition. The second approach is the edge map [4]

of the image which could be a useful object representation feature that's insensitive to illumination changes to certain event. Edge images might be used for recognition and to realize similar accuracy as gray level pictures. The edge map information approach owns the advantage of feature-based approaches, like invariance to illumination and low memory requirement. It integrates the structural information with spatial information of a face image which can be done by grouping pixels of face edge map to line segments. After thinning the edge map, a polygonal line fitting process is applied to come back up with the edge map of a face [5] [6] [7] There is one another approach through which the image disparities because of illumination differences are handled; it's by employing a model of several images [8] of the identical face which is taken under various illumination conditions. During this kind of approach, the pictures captured may be used as independent models or as a combined model-based recognition system [9] [10].

PROPOSED SYSTEM:

When we run the program, a window is opened and asks for Enter Id and Enter Name. After entering respective name and id fields then we have to click Take Images button. By clicking the Take Images button, a camera of running computer is opened and it starts taking image samples of person. This Id and Name is stored in Student Details folder and file name is saved as Student Details.csv. It takes 60 images as sample and stores them in Training Image folder. After completion it notifies that images saved. After taking image samples in order to train the image samples we have to click Train Image button. Now it takes few seconds to train the machine for the images and creates a Trainner.yml file and stores them in TrainingImageLabel folder. Now all initial setups are done. After completion of take images and Train images we have to click Track images button which is used to track the faces. If the face of particular student is recognized by the camera then Id and Name of person is shown on Image. Press Q (or q) for quit this window. After coming out of it, attendance of particular person will be stored in Attendance folder as csv file with name, id, date and time and it is also available in window.

SYSTEM ARCHITECTURE:



MODULE DESCRIPTION:

* **IMAGE ACQUISITION**
* **PRE-PROCESSING IMAGE**
* **FACE DETECTION**

FACE RECOGNITION

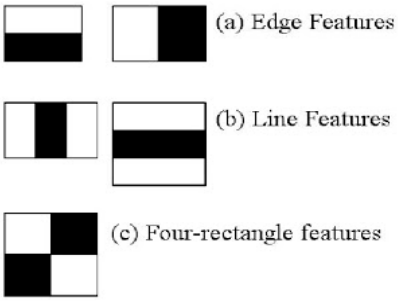
IMAGE ACQUISITION:

Image acquisition are often accomplished by digitally scanning an existing photograph or by using an electro-optical camera to accumulate a live picture of a topic . Video also can be used as a source of facial images. The most existing face recognition systems contain one camera.

PRE- PROCESSING IMAGE:

The system captures around 50 images of every individuals face. The images are converted into grey scale as LBPH operates using images in grey scale and the images are stored in a folder. The stored images will be saved with a name and ID unique to that person.

In Image pre-processing HAAR CASCADE algorithm was used to train the image. HAAR CASCADE features are more useful in image processing.



FACE RECOGNITION:
Facial recognition may be a way of recognizing a person's face through technology. A face recognition system uses biometrics to map countenance from a photograph or video. It compares the knowledge with a database of known faces to seek out a match. Facial recognition can help verify identity, but it also raises privacy issues.

FACE DETECTION:
Face detection may be a technology getting used during a sort of applications that identifies human faces in digital images. Face detection also refers to the psychological process by which humans locate and attend to faces during a visual scene.
IMPLEMENTATION:

1 HARDWARE REQUIREMENTS

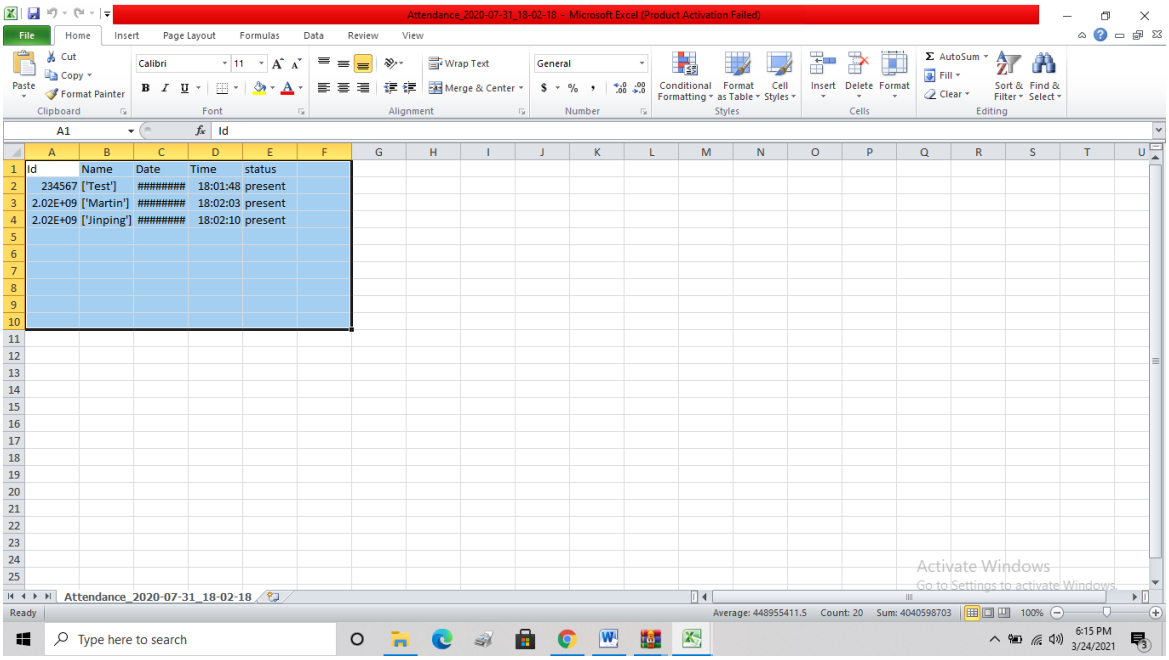
- Hard Disk :500GB and Above
- RAM :4GB and Above
- Processor :I3 and Above
- Webcam 1

3.4.2 SOFTWARE REQUIREMENTS

- Operating System: Windows 7 , 8, 10 (64 bit)
- Software : Python
- Tools : Python3.7 IDLE, Spyder

SCREENSHOTS:
STUDENT ATTENDANCE SYSTEM

ATTENDANCE VERIFICATION:



I. CONCLUSION:

We have implemented an attendance management system for student's attendance. It helps to reduce time and effort, especially in the case of large number of students marked attendance. The whole system is implemented in the Python programming language. Facial recognition techniques used for the purpose of the student attendance. And also, this record of student attendance can further be used mainly in exam related issues like who are attending the exams and who are not attending.

REFERENCES

[1]. V. Blanz and T. Vetter, "Face recognition based on fitting a 3D morphable model," IEEE Transactions On Pattern Analysis and Machine Intelligence, vol. 25, no. 9, September2003.

[2]. Y. Gao and K.H. Leung, "Face recognition using line edge map," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 6, June2002.

[3]. Michal Dolecki, Pawel Karczmarek, Adam Kiersztyn, Witold Pedrycz , "Face recognition by humans performed on basis of linguistic and neural network", Neural Networks (IJCNN)2016 International Joint Conference on , pp.5135-5140,2016,ISSN 2161-4407.

[4]. Ming Du, Aswin C.Sankaranarayanan, and Rama Chellappa , Robust Face Recognition from Multi-View Videos, IEEE "Transactions on Image Processing",Vol.23, No. 3,March 2014.Kyungnam Kim "Face Recognition using Principle Component Analysis", Department of Computer Science, University of Maryland, College Park, MD 20742, USA.

[5]. Javier Ruiz Del Solar, Rodrigo Verschae, and Mauricio Correa. Face recognition in unconstrained environments: A comparative study. In ECCV Workshop on Faces in RealLife Images: Detection, Alignment, and Recognition, Marseille, France, October 2008

[6]. H.K.Ekeneland R.Stiefelhagen,Analysis of local appearance based face recognition: Effects of feature selection and feature normalization. In CVPR Biometrics Workshop, New York, USA,2006.

REFERENCES

1.V. Blanz and T. Vetter, “Face recognition based on fitting a 3D morphable model,” IEEE Transactions On Pattern Analysis and Machine Intelligence, vol. 25, no. 9, September2003.

2.Y. Gao and K.H. Leung, “Face recognition using line edge map,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 6, June2002.

3. Michal Dolecki, Pawel Karczmarek, Adam Kiersztyn, Witold Pedrycz , “Face recognition by humans performed on basis of linguistic and neural network”, Neural Networks (IJCNN)2016 International Joint Conference on , pp.5135-5140,2016,ISSN 2161-4407.

4.Ming Du, Aswin C.Sankaranarayanan, and Rama Chellappa , Robust Face Recognition from Multi-View Videos, IEEE “Transactions on Image Processing”,Vol.23, No. 3,March 2014.Kyungnam Kim “Face Recognition using Principle Component Analysis”, Department of Computer Science, University of Maryland, College Park, MD 20742, USA.

5. Javier Ruiz Del Solar, Rodrigo Verschae, and Mauricio Correa. Face recognition in unconstrained environments: A comparative study. In ECCV Workshop on Faces in RealLife Images: Detection, Alignment, and Recognition, Marseille, France, October 2008

6.H.K.Ekenel and R.Stiefelhagen,Analysis of local appearance based face recognition: Effects of feature selection and feature normalization. In CVPR Biometrics Workshop, New York, USA,2006.

7.Goldstein, A.J., Harmon, L.D., and Lesk,A.B.(1971). Identification of human faces.

In Proc. IEEE, Vol.59, page 748.

8.A.F. Abate, M. Nappi, D. Riccio, and G. Sabatino, "2D and 3D face recognition: A survey", Pattern Recognition Letters, vol.28, issue 15, pp.1885-1906, Oct2007.

9.K. Wong, H. Law, and P. Tsang, “A System for Recognising Human Faces,” Proc. ICASSP, pp. 1,638-1,642,1989.

10.N. Dalal, B. Triggs “Histograms of oriented gradients for Human Detection”, IEEE Computer Society Conference on Computer Vision and Pattern Recognition , Vol. 1, 2005, pp. 886 –893.

11.S.Edelman, D.Reisfeld, and Y. Yeshurun, “A System for Face Recognition that Learns from Examples,” Proc. European Conf. Computer Vision, S. Sandini, ed., pp. 787-791. Springer- Verlag,1992.

12.P. Hallinan, “A Low-Dimensional Representation of Human Faces for Arbitrary Lighting Conditions,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 995-999,1994.

13.W.Zhao, R.Chellappa, P. J. Phillips, A. Rosenfeld, “Face Recognition: A Literature Survey.” ACM Computing Surveys, 2003, vol. 35, no. 4, pp.399-458.

14.L. Zhao and X. Xu, "Distance function for face recognition based on 2D PCA,"2011 Int. Conf. Multimed. Technol., pp. 5814-5817,2011.

15.Nasser, M. A. et al. (2011). "Automated facial recognition attendance system." Graduation Project, Petra University, January2011

16.RoshanTharanga, J. G., et al. "Smart Attendance Using Real Time Face

Recognition (SMAR-FR)." Department of Electronic and Computer Engineering, Sri Lanka Institute of Information Technology (SLIIT), Malabe, SriLanka.

17.D. Cahi and K.N. Ngan, “Face segmentation using skin color map”, IEEE Trans.On Circuit and Systems for Video Technology, vol 9, no 4, pp. 551-564, Jun.1999