**A Project Report**

On

# Face Recognition based Attendance Management System Using Haar cascade classifier and Pattern Histogram Algorithm .

***Submitted in partial fulfillment of the***

***requirement for the award of the degree of***

Bachelor of Technology in Computer Science and Engineering


**Under The Supervision of**
**DR.SPS Chauhan**
**Assistant Professor**

**Department of Computer Science and Engineering**

**Submitted By**

19SCSE1010437 - Ravi Ranjan Kumar

18SCSE1010718 - Rishav Kewat

**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**GALGOTIAS UNIVERSITY, GREATER NOIDA, INDIA**

**DECEMBER - 2021**

# SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

# GALGOTIAS UNIVERSITY, GREATER NOIDA

## CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the project, entitled **" Ur Info: A cross-platform Application "** in partial fulfillment of the requirements for the award of the **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING**

submitted in the **School of Computing Science and Engineering** of Galgotias University, Greater Noida, is an original work carried out during the period of **JULY-2021 to DECEMBER-2021**, under the supervision of **DR.SPS Chauhan , Assistant Professor**, **Department of Computer Science and Engineering** of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the project has not been submitted by me/us for the award of any other degree of this or any other places.

19SCSE1010437 – Ravi Ranjan Kumar

18SCSE1010718– Rishav Kewat

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Supervisor

(DR.SPS Chauhan, Assistant Professor)

**CERTIFICATE**

The Final Thesis/Project/ Dissertation Viva-Voce examination of **19SCSE1010437 – Ravi Ranjan Kumar, 18SCSE1010718- Rishav Kewat** has been held on GALGOTIAS UNIVERSITY and his/her work is recommended for the award of **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING**.

**Signature of Examiner(s)**
**(MR.C.RAMESH KUMAR)**
**(Assistant Professor)**

**Signature of Supervisor(s)**
**(DR.SPS CHAUHAN )**
**(Assistant Profesoor)**

**Signature of Project Coordinator**
**(MR.V.Arul)**
**(Assistant Professor)**

**Signature of Dean**

## Table of Contents


| Title | | Page No. |
|---|---|---|
| **Candidates Declaration** | | |
| **Acknowledgement Abstract** | | |
| **List of Figures** | | |
| **Chapter 1** | **Introduction** | **6** |
| | **1.1** Facial Recognition Attendance System | **6** |
| | **1.2** Algorithms used | **7** |
| **Chapter 2** | **Literature Survey/Project Design** | **8** |
| **Chapter 3** | **Functionality/Working of Project** | **15** |
| **Chapter 4** | **Results and Discussion** | **25** |
| **Chapter 5** | **Conclusion and Future Scope** | **26** |
| | **Reference** | **29** |
| | **Publication/Copyright/Product** | **31** |

**List of Figures**


| S.No. | Caption | Page No. |
|---|---|---|
| 1 | Eigen faces diagram | 11 |
| 2 | Local Binary Pattern Histogram | 13 |
| 3 | Data Flow Diagram | 14 |
| 4 | UML Diagram | 15 |
| 5 | Integral image ,<br>Face Recognition | 16<br>19 |


# ACKNOWLEDGMENT


We would like to express my gratitude towards **Dr.SPS Chauhan** for guiding us throughout the project. We also feel thankful and express my kind gratitude towards our guide **Dr.SPS Chauhan** for allowing us to conduct **Face Recognition based Attendance Management System Using Haar cascade classifier and Pattern Histogram Algorithm** project. The mentioned project was done under the supervision of **Mr.V.Arul, (Assistant Professor)** .

We feel thankful to the Galgotias University for giving us such a big opportunity. We believe we will enroll in more such events in the coming future. We ensure that this project was done by ourself and is not copied.

19SCSE1010437 – Ravi Ranjan Kumar

18SCSE1010718 – Rishav Kewat

**ABSTRACT :**

ABSTRACT In this digital era, face recognition system plays a vital role in almost every sector. Face recognition is one of the mostly used biometrics. It can used for security, authentication, identification, and has got many more advantages. Despite of having low accuracy when compared to iris recognition and fingerprint recognition, it is being widely used due to its contactless and non-invasive process. Furthermore, face recognition system can also be used for attendance marking in schools, colleges, offices, etc. This system aims to build a class attendance system which uses the concept of face recognition as existing manual attendance system is time consuming and cumbersome to maintain. And there may be chances of proxy attendance. Thus, the need for this system increases. This system consists of four phasesdatabase creation, face detection, face recognition, attendance updation. Database is created by the images of the students in class. Face detection and recognition is performed using HaarCascade classifier and Local Binary Pattern Histogram algorithm respectively. Faces are detected and recognized from live streaming video of the classroom. Attendance will be mailed to the respective faculty at the end of the session. Technology used : -open CV (Open source Computer Vision) -Python – tkinter GUI interface . The cropped images are then stored as a database with respective labels. The features are extracted using LBPH algorithm.

# Introduction:

**Facial Recognition Attendance System| OPEN-CV | ML**

An efficient module that comprises of face recognition using Open CV to manage the attendance records of employees or students.

**Things used in this project.**

Hardware components: Camera (generic).

Software apps and online services.

## Open CV :

Open CV (Open Source Computer Vision) is a popular computer vision library started by Intel in 1999. The cross-platform library sets its focus on real-time image processing and includes patent-free implementations of the latest computer vision algorithms. In 2008 Willow Garage took over support and Open CV 2.3.1 now comes with a programming interface to C, C++, Python and Android. Open CV is released under a BSD license so it is used in academic projects and commercial products alike.

Attendance maintenance is a significant function in all the institutions to monitor the performance of the students. A facial recognition system is a computerized biometric software which is suited for determining or validating a person by performing comparison on patterns based on their facial appearances. Face recognition systems have upgraded appreciably in their management over the recent years and this technology is now vastly used for various objectives like security and in commercial operations. Face recognition is a powerful field of research which is a computer based digital technology. Face recognition for the intent of marking attendance is a resourceful application of attendance system. It is widely used in security systems. As the number of students in an educational institute or employees at an organization increases, the needs for lecturers or to the organization also increase the complication of attendance control.

**The currently available algorithms are:**

- Eigen faces (see Eigen Face Recognizer::create)
- Fisher faces (see Fisher Face Recognizer::create)
- Local Binary Patterns Histograms (see LBPH Face Recognizer::create)

**The 3 Phases**

To Create A Complete Project On Face Recognition, We Must Work On 3 Very Distinct Phases:

- Face Detection And Data Gathering
- Train The Recognizer
- Face Recognition

1. **Face Detection** :
   Face detection is a technique that identifies or locates human faces in digital images. A typical example of face detection occurs when we take photographs through our smartphones, and it instantly detects faces in the picture. Face detection is different from Face recognition. Face 6 detection detects merely the presence of faces in an image while facial recognition involves identifying whose face it is.
2. **Trainer :**
   On this second phase, we must take all user data from our dataset and "trainer" the Open CV Recognizer. This is done directly by a specific Open CV function. The result will be a .yml file that will be saved on a "trainer/" directory.
3. **Recognizer :**
   Now, we reached the final phase of our project. Here, we will capture a fresh face on our camera and if this person had his face captured and trained before, our recognizer will make a "prediction" returning its id and an index, shown how confident the recognizer is with this match.

## Problem analysis/Literature review:

Face recognition has taken a dramatic change in today's world of, it has been widely spread throughout last few years in drastic way. There have been some drastic improvements in last few years which hasmade it so much popular that now it is being widely used for commercial purpose as well as security purpose also. Tracking a user presence is becoming one of the problems in today's world, so an attendance system based on facial recognition can act as a real world solution to this problem and add great heights of simplicity for tracking a user attendance .The manual entering of attendance in logbooks becomes difficult and takes a lot of time also, so we have designed an efficient module that comprises of face recognition using LBPH algorithm(Open

CV) to manage the attendance records of employee or students. During enrolling of a user, we take multiple images of a user along with his/her id/roll number and name also .The presence of each student/employee will be updated in database, and the user can check their attendance on the webpage also. The results showed improved performance over manual attendance system .This process can give us more accurate results in user interactive manner rather than the existing attendance systems .This also gives students/employees a more accurate result in user interactive manner rather than existing attendance management system.
This project may be helpful for the explanation of these types of problems. The number of students present in a lecture hall is observed, each person is identified and then the information about the number of students who are present.

## Technology/Tools Used :

Here, We will be using various python libraries and modules for face recognition, face identification, saving a user image and other information also . We use **OPEN-CV(Open Source Computer Vision)** library for face recognition, identification, we use **pandas** package to store student information in local database **, Numpy** is used to perform the appropriate task, **pymy sql** is used to connect to a **MySQL database**, **Tkinter** helps us to make **GUI** for better interaction with the program .In this project, we use **MySQL database** to store the students attendance .For Webpage, to implement our front-end, we have used **HTML, CSS/SCSS** and for better interaction we have used **JavaScript** and **JQuery**. As far as back-end technology is concerned we have used **PHP** for that.

## Existing System :

Traditional attendance marking techniques i.e, pen and paper or signing attendance sheets are easy to bypass and trick as giving proxies or false signatures is a common practice among students nowadays, students take an unfair advantage of this at most times. But a facial recognition

system is unassailable and cannot be fooled as each person has a set of unique and individual features common to that person and cannot be replicated or changed, it all comes down to one simple truth that is, unless you are physically present in the lecture your attendance will not get marked.

**Existing systems and their limitations:**

1. Pen and paper False signatures and proxies.
2. RFID tags Can be used by anybody, no guarantee.
3. Biometric, fingerprint is a costlier approach.

## Project Design :

The design part of the attendance monitoring system is divided into two sections which consist of the hardware and the software part. Before the software .The design part can be developed, the hardware part is first completed to provide a platform for the software to work. Before the software part we need to install some libraries for effective working of the application.

We install **Open CV** and **Numpy through** Python.

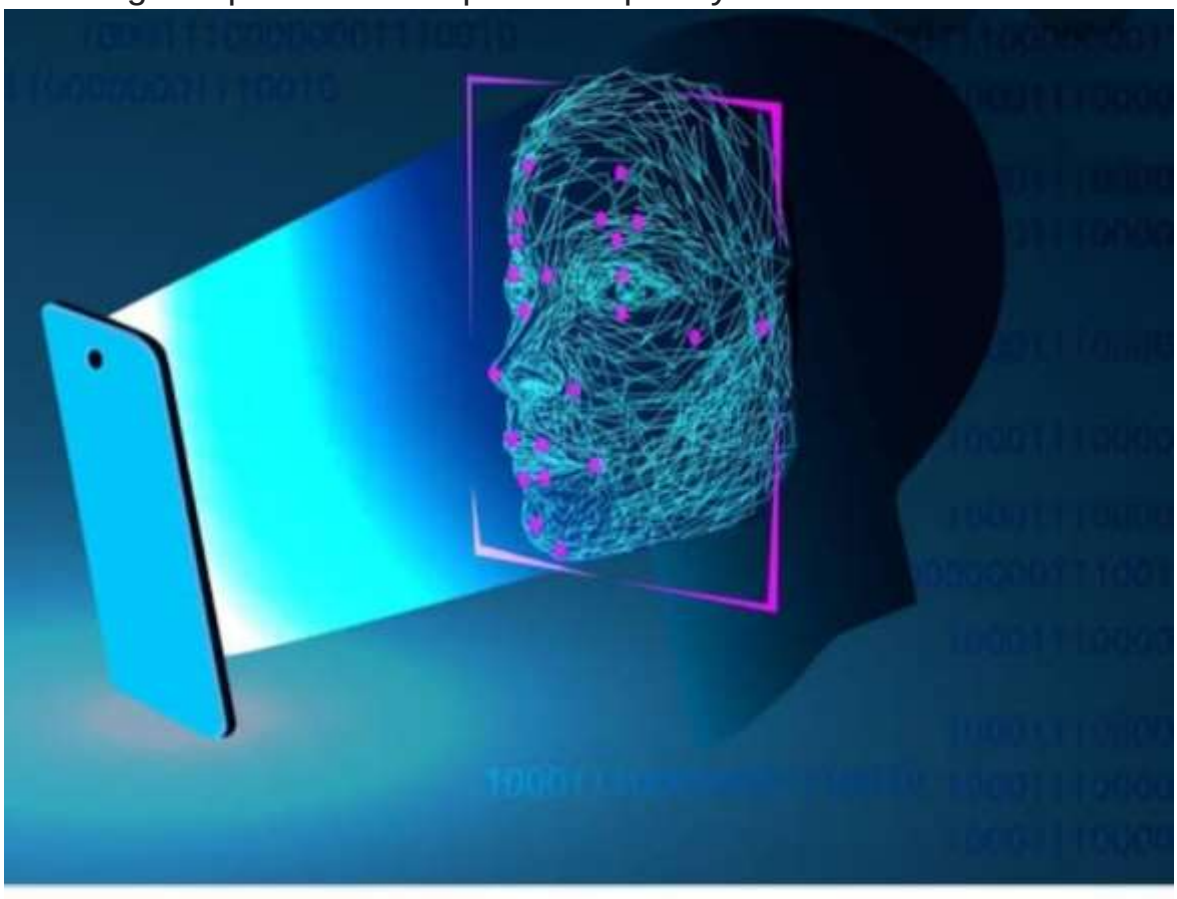
**Algorithms :**

The currently available Face Recognizer Algorithms in OPEN-CV are:

- Eigen Faces
- Fisher Faces
- Local Binary Patterns Histograms For our purpose, we would be using the last algorithm (Local Binary Patterns Histogram)

## • Eigen Faces :

Eigenfaces is a method that is useful for face recognition and detection by determining the variance of faces in a collection of face images and use those variances to encode and decode a face in a machine learning way without the full information reducing computation and space complexity.



- **Fisher Faces:**

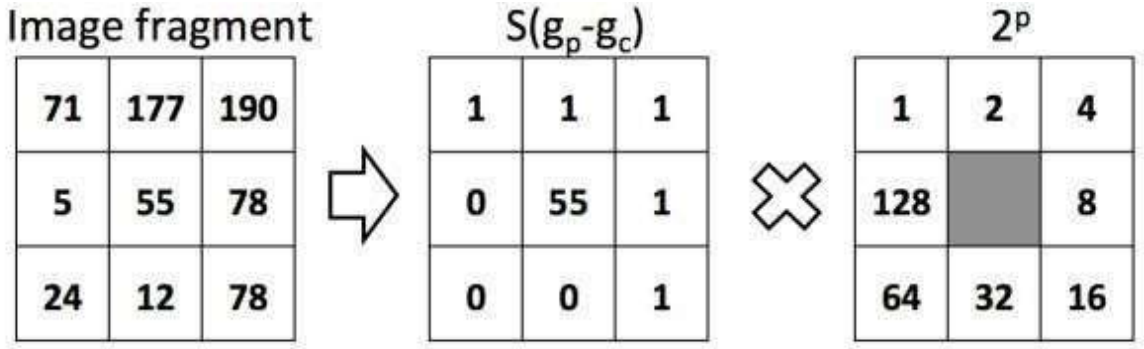
Fisherfaces algorithm extracts principle components that separates one individual from another. So, now an individual's features can't . Image recognition using this algorithm is based on reduction of face space domentions using PCA method and then applying LDA method also known as Fisher Linear Discriminant (FDL) method to obtain characteristic features of image.

- **Local Binary Pattern Histogram:**

Eigen Faces and Fisher Faces take a somewhat holistic approach to face-recognition. You treat your data as a vector somewhere in a high-dimensional image space. We all know highdimension is bad, so a lower-dimensional subspace is identified, where (probably) useful 13 information is preserved. The Eigen Faces approach maximizes the total scatter, which can lead to problems if the variance is generated by an external source, because components with a maximum variance over all classes aren't necessarily useful for classification. So to preserve some discriminative information we applied a Linear Discriminant Analysis and optimized as described in the Fisher Faces method. The Fisher Faces method worked great at least for the constrained scenario we've assumed in our mode.

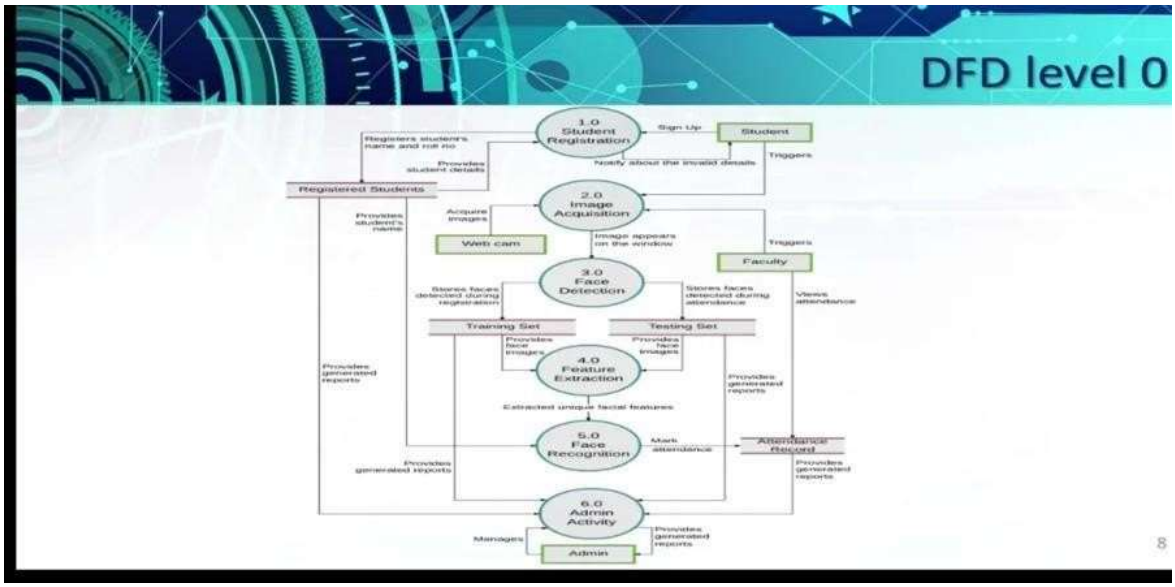
Now real life isn't perfect. You simply can't guarantee perfect light settings in your images or 10 different images of a person. So what if there's only one image for each person? Our co-variance estimates for the subspace may be horribly wrong, so will the recognition .So some research concentrated on extracting local features from images. The idea is to not look at the whole image as a high-dimensional vector. Now real life isn't perfect. You simply can't guarantee perfect light settings in your images or 10 different images of a person. So what if there's only one image for each person? Our co-variance estimates for the subspace may be horribly wrong, so will the recognition.So some research concentrated on extracting local features from images. The idea isto not look at the whole image as a high-dimensional vector, but describeonly

local features of an object. The features you extract this way will have a low-dimension implicitly. A fine idea! But you'll soon observe the image representation we are given doesn't only suffer from illumination variations. Think of things like scale, translation or rotation in images - your local description has to be at least a bit robust against those things. The Local Binary Patterns methodology has its roots in 2D texture analysis. The basic idea of Local Binary Patterns is to summarize the local structure in an image by comparing each pixel with its neighborhood. Take a pixel as center and threshold its neighbours against. If the intensity of the center pixel is greater-equal its neighbor, then denote it with 1 and 0 if not. You'll end up with a binary number for each pixel, just like 11001111. So with 8 surrounding pixels you'll end up with 2^8 possible combinations, called Local Binary Patterns or sometimes referred to as LBPcodes. The first LBP operator described in literature actually used a fixed 3 x 3neighborhood just like this:

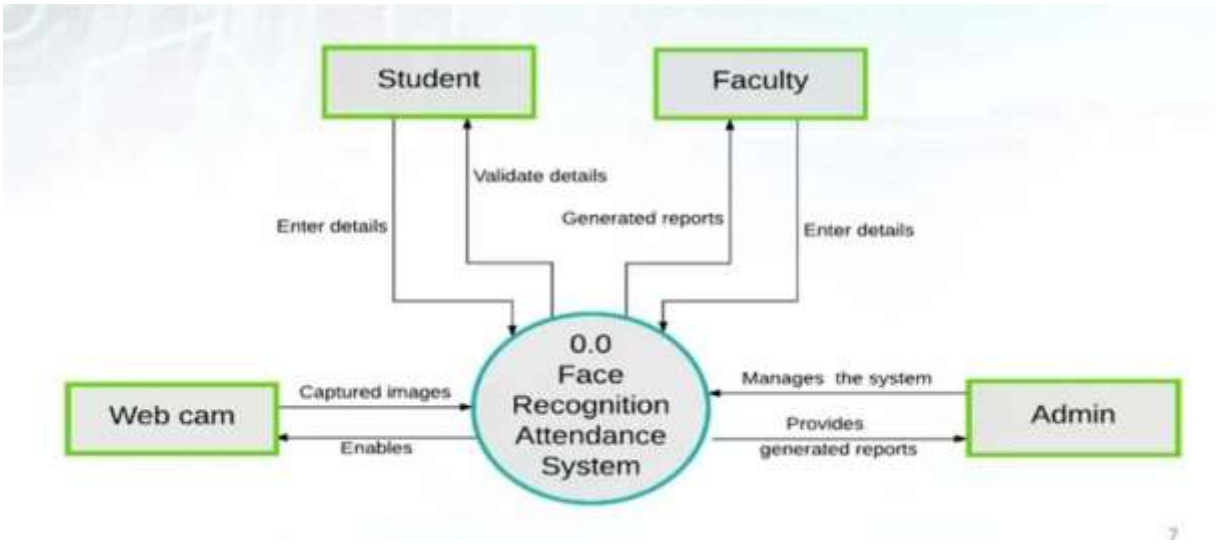



**LBP=1 x 1 + 1 x 2 +1 x 4 + 1 x 8 + 1 x 16 + 0 x 32 + 0 x 64 + 0 x 128= 4 + 8 + 16 = 31**

**Data Flow Diagram :**



**UML ( Unified Modeling Language) :**




## Working Of Project / Methodology :

The proposed methodology starts with the registration of students into the system. Following methodology has few main stages such as capturing images, pre-processing of the images, Haar Cascade classifier is used for face detection, developing a dataset of images, the further process of face recognition is done with the help of LBPH algorithm as shown in fig 1.

**Proposed Methodology for Attendance monitoring.**

1. Image Capture: The high-resolution camera which is used for capturing video is used to take frontal images of the students.

2. Pre-processing: The images are converted from RGB to Grayscale and are scaled down by a factor of 1.2.

3. Face Detection Face Detection is composed of four stages as shown in fig 2.

**Face Detection:**

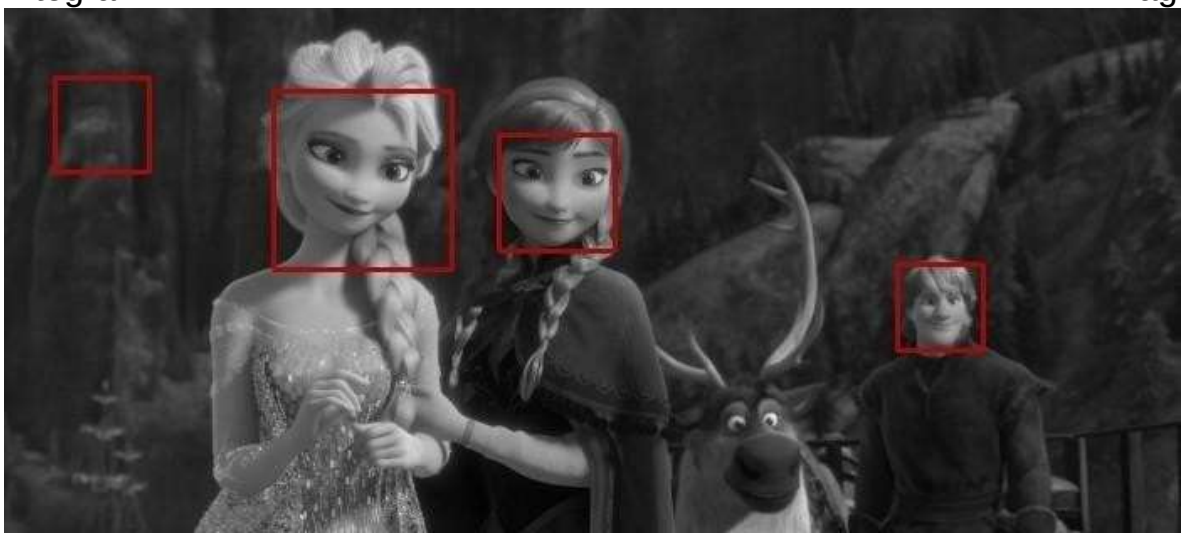
**Haar Features**

Haar features are the same as convolutional kernels and are used to detect the features in a given image. There are different kinds of Haar features such as line feature, edge feature, four – rectangle feature etc. A single value is used to represent each feature which i calculated by subtracting the sum of pixels under the white rectangle from the sum of pixels under the black rectangle as shown in fig 3. Haar cascade algorithm makes use of 24*24 windows which ends up calculating 160000+ features in a window. To simplify the work of calculating the feature values, an Integral image algorithm is introduced.

**Integral Image:**

To reduce the computation of the pixels to find the feature values, Viola Jones introduced a technique called Integral Image. As shown in fig 4, the value of the pixel at (x, y) in an Integral image is calculated by adding the values of pixels above and to the left of (x, y) pixel.

Calculation of the sum of all pixels inside any given rectangle can be done by using only four values at the corner of that rectangle with the help of Integral image.



## AdaBoost Algorithm

As Haar cascade us 24*24 windows there can be 160000+ features within a detector which needs to be calculated. AdaBoost is an algorithm based on machine learning. This algorithm finds the best features among all the possible features and eliminates the irrelevant features. To evaluate and decide whether a given window has a face or not, a weighted combination of all the found features are used. Features are included only if they can at least work better than random estimation. These features are known as weak features. AdaBoost

linearly combines these weak classifiers to design a strong classifier. The weak classifier gives 0 or 1 as an output depending on its performance in image. Output is 1 when it performs well and able to identify features applied on image and Output is 0 when no pattern of the feature is present in image.

**The steps involved in the AdaBoost algorithm are as follows:**

1. It distributes the uniform weights over training examples, positive weights for faces and negative weights for non-faces.

2. Selects the weakest classifier i.e., with lowest weighted error. For e.g., x=0.5 if x>0.5 then they are faces otherwise non-faces, but due to this some of the non-faces comes under faces category.

3. To overcome this drawback the algorithm increases the weights on the misclassified training examples because we need a new classifier which concentrates more on misclassified features.

4. These steps are repeated and at the end, all the weak classifiers obtained at all iterations are combined linearly and help to define a perfect boundary region.

5. A single rectangular feature is used to classify a single rectangular feature and with respect to the weighted neighbour, the positive and negative images are separated. Gaussian weak classifiers are also used for this purpose.

**a.C ascading:**

In Paul Viola and Michael Jones detection algorithm, one single image is scanned many times by the detector with a new size every time. When multiple faces appear in an image the algorithm concentrates on removal of non-faces and brings out the most feasible face area. Since the computation

cost is very high for each window when a particular strong classifier is a linear combination of all the best features and is not so appropriate for evaluation. Hence a cascade classifier is used.

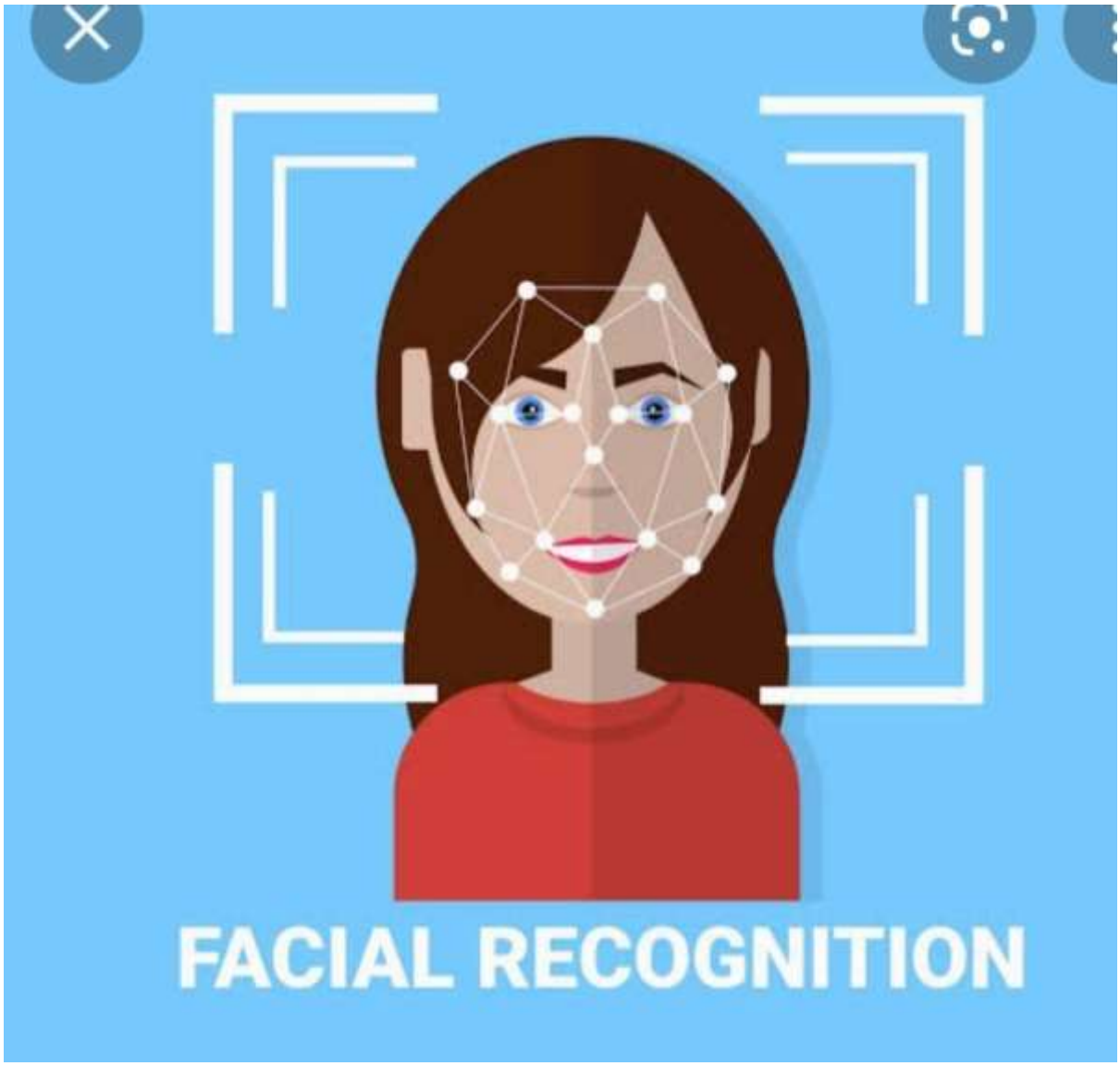
Cascade classifiers comprise various stages. All the features are grouped into different stages and each stage contains a strong classifier and a feature. To design these stages, AdaBoost is used. As shown in fig 7, each stage decides whether a sub- window is a face or not. The window is discarded if it does not contain a face. While training a classifier the numberof features, stages and thresholds are taken into account.

## Developing a dataset

The faces detected in images are stored in the database after pre-processing and detection. A minimum of 20 images are captured per individual student along with a unique ID. The dimensions of these stored images are 212×212 pixels. These images are later used to train the recognizer.

## Face Recognition

Local Binary Pattern (LBP) is a smooth & adequate operator, which operates by setting the pixels of an image by thresholding the neighborhood of each pixeland examines the outcome as a binary number. Histogram of Oriented Gradients (HOG) descriptor increases the detection performance when combined with LBP. Therefore, a combination of LBP & HOG which gives LBPH algorithm is used for face recognition.




Steps involved in LBPH are as follows.

1. LBPH considers four parameters for face recognition which are as follows Radius: To set up a circular local binary pattern radius is used. Generally, it is set to 1. Neighbors: To set the circular local binary pattern neighbors are used. Normally, set to 8. Grid X: Gives cells count which are in horizontal direction. Normally, it is set to 8. Grid Y: Gives cells count which are in vertical direction. Normally, it is set to 8.

2. Training the Algorithm: A database of the face images of students which are to be recognized is used to train the

algorithm. The unique ID which is set while developing a dataset is useful for recognizing the student.

3. Applying the LBP operation: By intensifying facial characteristics, create an intermediate image that describes the original image. Based on the parameters like radius and neighbors, the algorithm uses a sliding window concept..

    - Assume that we have a grayscale facial image, take a part of images as a 3 x 3 matrix containing pixel intensities in the range (0 255), as shown in step1 and 2 in fig 8.

    - In step 3, consider central pixel intensity as threshold and change the values of 8 neighbors with respect to the threshold value. (Set it to 1 if neighboring pixel intensity is greater than or equal to the threshold value, otherwise set to 0.)

    - In step 4, convert the binary value into decimal value. The central pixel value of the image matrix is replaced by a decimal value. This central pixel is actually a pixel of a primary image.

    - Applying these steps to all the parts of the image, we get a new image (result of LBP operation) that describes the features of the primary image.

**Extracting the Histograms:** Grid X and Grid Y parameters are used to divide images into multiple grids.

Each histogram holds only 256 positions (0-255) that shows the existence of each pixel intensity as the image is in grayscale. Histogram of each cell is to be concatenated to generate a bigger and new histogram. The final histogram shows the characteristics of the primary image.

Performing the face recognition: The algorithm for creating histogram is initially trained. Each image from the training dataset is represented by each histogram. To generate a histogram for the input image the above steps are performed again on that image. The histogram of input image is compared with the histograms of dataset images, selecting the closest histogram gives the matching

image from the dataset. Various methods like Absolute value, Euclidean distance, etc can be used to compare the histograms. The Euclidean

distance can be calculated using equation 1 to compare the histograms.

Where HistD – Histograms of dataset images HistR – Histogram of real time image

The algorithm returns a unique Id of the student with the minimum difference in the histograms of the student's image and dataset images. It also returns the calculated distnce, which can be used as a confidence measurement. Lower the confidence measurement, more is the precision of the recognizer.

**Implementation:**

```
import cv2
import numpy as np
import face_recognition
import os
import datetime

path='imgfld'
images=[]
classnames=[]
mylist=os.listdir(path)
print(mylist)


for cl in mylist:
    curimg=cv2.imread(f'{path}/{cl}')
    images.append(curimg)
    classnames.append(os.path.splitext(cl)[0])
print(classnames)
```

```
def findEncoding(images):
    encodelist=[]
    for img in images:
        img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
        encode=face_recognition.face_encodings(img)[0]
        encodelist.append(encode)
    return encodelist

def markattendance(name):
    with open('Attendancenew.csv','r+') as f:
        mydatalist=f.readlines()
        #print(mydatalist)
        namelist=[]
        for line in mydatalist:
            entry=line.split(',')
            namelist.append(entry[0])
        if name not in namelist:
            now=datetime.datetime.now()
            dtString=now.strftime('%H:%M:%S')
            f.writelines(f'\n{name},{dtString}')

markattendance(' sakshi')


encodelistknown=findEncoding(images)
print(len(encodelistknown))
print('Encoding complete')

cap=cv2.VideoCapture(0)

while True:
```

```
    success,img=cap.read()
    imgS=cv2.resize(img,(0,0),None,0.25,0.25)
    imgS=cv2.cvtColor(imgS,cv2.COLOR_BGR2RGB)

    facescurFrame=face_recognition.face_locations(imgS)

encodecurframe=face_recognition.face_encodings(imgS,facesc
urFrame)

    for encodeface,faceloc in
zip(encodecurframe,facescurFrame):

matches=face_recognition.compare_faces(encodelistknown,en
codeface)

facedis=face_recognition.face_distance(encodelistknown,enco
deface)
       # print(facedis)
        matchindex=np.argmin(facedis)

        if matches[matchindex]:
            name=classnames[matchindex].upper()
            #print(name)
            y1,x2,y2,x1=faceloc
            y1, x2, y2, x1=y1*4,x2*4,y2*4,x1*4
            cv2.rectangle(img,(x1,y1),(x2,y2),(0,255,0),2)
            cv2.rectangle(img,(x1,y2-
35),(x2,y2),(0,255,0),cv2.FILLED)
            cv2.putText(img,name,(x1+6,y2-
6),cv2.FONT_HERSHEY_COMPLEX,1,(255,255,255),4)
            markattendance(name)

    cv2.imshow('webcam',img)
```

```
    cv2.waitKey(1)
================================================================
==============



import cv2
import numpy as np
import face_recognition

pic2=face_recognition.load_image_file("formalpic.jpg")
pic2=cv2.cvtColor(pic2,cv2.COLOR_BGR2RGB)

pic1=face_recognition.load_image_file("guddi.JPG")
pic1=cv2.cvtColor(pic1,cv2.COLOR_BGR2RGB)

faceloc=face_recognition.face_locations(pic2)[0]
encodepic=face_recognition.face_encodings(pic2)[0]
cv2.rectangle(pic2,(faceloc[3],faceloc[0]),(faceloc[1],faceloc[2]),
(255,0,255,),2)

faceloctest=face_recognition.face_locations(pic1)[0]
encodepictest=face_recognition.face_encodings(pic1)[0]
cv2.rectangle(pic1,(faceloctest[3],faceloctest[0]),(faceloctest[1]
,faceloctest[2]),(255,0,255,),2)

results=face_recognition.compare_faces([encodepic],encodepi
ctest)
facedis=face_recognition.face_distance([encodepic],encodepict
est)
print(results,facedis)
cv2.putText(pic1,f'{results}{round(facedis[0],2)}',(50,50),cv2.FO
NT_HERSHEY_COMPLEX,1,(0,0,255),2)
```

```
cv2.imshow('formalpic',pic2)
cv2.imshow('guddi',pic1)
cv2.waitKey(0)

================================================
================================================
========

Make CSV file
=============
Output
======
Name, Time
gudiya,15:16:13
GUDDIYATEST,15:16:31
MUMMY,15:16:32
SAKSHI,15:54:06
sakshi,16:04:21
```

**Results and Discussion :**

**Results :**

Once the image is taken then the system automatically runs the algorithm and generates the data, the data is then logged in excel sheet for ease with maintain, the actual time of attendance is also shown in the excel file. The details are stored in .csv format and the results with good accuracy is then electronically mailed to the staff in charge automatically.

**Results and Discussion :** Fig. 7. Recognizing the faces

**Discussion:** In this project perform face recognition in both images and video streams using:

- OpenCV

- Python
- Deep learning

The deep learning-based facial embeddings we'll be using here today are both (1) highly accurate and (2) capable of being executed in real-time.

## Conclusions :

Face recognition is an emerging technology that can provide many benefits. Face recognition can save resources and time, and even generate new income streams, for companies that implement it right. If this prediction becomes a reality, any company that implemented the technology today might gain a competitive advantage in the future. Face recognition technology has come a long way in the last twenty years. Today, machines are able to automatically verify identity information for secure transactions, for surveillance and security tasks, and for access control to buildings etc. These applications usually work in controlled environments and recognition algorithms can take advantage of the environmental constraints to obtain high recognition accuracy. However, next generation face recognition systems are going to have widespread application in smart environments -- where computers and machines are more like helpful assistants.

## FUTURE SCOPE :

The future scope of the project can be integrated with the hardware components for example GSM through which a monthly list of the defaulter students can be sent to the mentor.

Additionally, an application can be developed to help students to maintain a track of their attendance. It can also be used in offices where a large group of employees sit in a hall and their attendance will be marked automatically by capturing a video but for this the accuracy of the recognition needs to be improved.

# REFERENCES


[1] Xin Geng, Zhi-Hua Zhou, & Smith-Miles, K. (2008). Individual Stable Space: An Approach to Face Recognition Under Uncontrolled Conditions. IEEE Transactions on Neural Networks.

[2] Winarno, Wiwien Hadikurniawati, Imam Husni Al Amin, Muji Sukur,"Anti-Cheating

[3] Presence System Based on 3WPCA Dual Vision Face Recognition, Faculty of Information Technology Universitas Stikubank Semarang Indonesia.

[4] Prototype model for an Intelligent Attendance System based on facial Identification by Raj Malik, Praveen Kumar, Amit Verma, Seema Rawat, Amity University Uttar Pradesh.

[5] Convolutional Neural Network Approach for Vision Based Student Recognition System, Nusrat Mubin Ara1, Dept. of CSE, SUST, Sylhet, Bangladesh.

[6] NFC Based Mobile Attendance System with Facial Authorization on Raspberry Pi and Cloud Server Siti Ummi Masruroh Andrew Fiade Imelda Ristanti Julia.

[7] Face recognition-based AttendanceSystem using Machine Learning Algorithms, Radhika C. Damale, Department of Electronics and Telecommunications, Cummins College of engineering for Women, Pune, Maharashtra, India.

[8] Class Attendance system based on Face Recognition" Priyanka Wagh.

[9] Design of Classroom Attendance System Based on Face Recognition,WenxianZeng.

[10] Automated Attendance System Using Face Recognition, Akshara Jadhav, Akshay Jadhav Tushar Ladhe, Krishna Yeolekar.

[11] An Attendance Marking Systembased on Face Recognition" written by Khem Puthea, Rudy Hartanto and Risanuri Hidayat.

[12] Class Attendance Management System Using Face Recognition ,Omar Abdul Rhman Salim Department of Electrical and ComputerEngineering, Faculty of Engineering International Islamic University Malaysia, Kuala Lumpur,Malaysia o.salem92@gmail.com

[13] Face Recognition Based Attendance System Nandhini R, Duraimurugan N.

[14] Student Attendance System in Classroom Using Face RecognitionTechnique, Samuel LukasAditya Rama Mitra, Ririn Ikana Desanti, Dion Krisnadi, Informatics Department,Computer System Department, Information System Department Universitas Pelita Harapan Karawaci, Indonesia.

[15] [14] Attendance System based on Face Recognition Venkata Kalyan Polamarasetty, Muralidhar Reddy Reddem, Dheeraj Ravi,Mahith Sai Madala.

Thanks for your contribution to the International Journal of Scientific and Engineering Research (IJSER) - (ISSN 2229-5518).

Dear Author,
We are pleased to inform you that your paper:

Paper Title: FACE RECOGNITION BASED ATTENDENCE SYSTEM USING HAAR CASCADE AND PATTERN ALGORTHIM
Author": RARI RANJAN KUMAR, RISHAV KEWAT

has been accepted for publication in USER, which will be published in USER Volume 12, Issue 12, December 2021Edition. Congratulations!
At least one author of each accepted paper should register as the following steps ( till Dec 11", 202a ):

&ips:ri|Ii we #=n rorm

Please complete the online registration checklist form before sending the registration completion mail to us. (mandatory for all the authors )
h -

Digital Signed Certificate
(Please fill the certificate request form after completion of registration process, we'll send you certificates )
h - iser.org

# ACCEPTED CERTIFICATE

ISSN 2229-5518

CERTIFICATE OF ACCEPTANCE

**International Journal of Scientific & Engineering Research (IJSER)**

THIS IS TO CERTIFY THAT OUR REVIEW BOARD HAS ACCEPTED RESEARCH PAPER OF

RAVI RANJAN KUMAR

FACE RECOGNITION BASED ATTENDENCE SYSTEM USING HAAR CASCADE AND PATTERN ALGORTHIM

May 10,2022

Visit us at: **www.ijser.org**

IJSER 2022

Editor in Chief

ISSN 2229-5518

# CERTIFICATE OF ACCEPTANCE

**International Journal of Scientific & Engineering Research (IJSER)**

THIS IS TO CERTIFY THAT OUR REVIEW BOARD HAS ACCEPTED RESEARCH PAPER OF

# RISHAV KEWAT

FACE RECOGNITION BASED ATTENDENCE SYSTEM USING HAAR CASCADE AND PATTERN ALGORTHIM

May 10,2022

Visit us at: **www.ijser.org**

IJSER 2022 PUBLICATION

Editor in Chief

Thank You !!!!!