IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 13, April 2026
Impact Factor: 8.2

Face Recognition Attendance System

Mayank Chaudhary, Neeraj Kumar, Archit Chaudhary, Anuj Mavi
CSE (AI&ML)
Raj Kumar Goel Institute of Technology, Ghaziabad, India
chaudharymayank8394@gmail.com, architchoudhary.1439gmail.com, mavianuj7166@gmail.com

Abstract: *In the era of modern technologies emerging at rapid pace there is no reason why a crucial event in educational sector such as attendance should be done in the old boring traditional way.*
Attendance monitoring system will save a lot of time and energy for the both parties students as well as the class teachers. Attendance will be monitored by the face recognition algorithm by recognizing only the face of the students from the rest of the objects and then marking them as present. The system will be pre feed with the images of all the students and with the help of this pre feed data the algorithm will detect them who are present and match the features with the already saved images of them present in the database.
With the increasing adoption of automation and artificial intelligence, educational institutions are moving toward smart systems that reduce manual effort and improve accuracy. Attendance management is a fundamental yet repetitive task that, when performed using traditional methods, often results in time consumption, errors, and proxy attendance. To address these challenges, this project proposes a Face Recognition–based Attendance System that automates the process of recording student attendance in a secure and efficient manner.
The proposed system utilizes computer vision and machine learning techniques to detect and recognize faces captured through a camera in real time. Facial features extracted from live images are compared with a previously created database of registered students. Upon successful identification, the system automatically marks attendance along with date and time, ensuring reliable record maintenance without human intervention. This approach minimizes manual workload for teachers and enhances the authenticity of attendance data..

Keywords: *artificial intelligence*.

I. INTRODUCTION

The purpose of the attendance monitoring system using face recognition is to ease the attendance process which consumes lot of time and efforts , it is a convenient and easy way for students and teacher. The system will capture the images of the students and using face recognition algorithm mark the attendance in the sheet. This way the class-teacher will get their attendance marked without actually spending time in traditional attendance marking.
The identification process to determine the presence of a person in a room or building is currently one of the routine security activities. Every person who will enter a room or building must go through several authentication processes first, that later these information's can be used to monitor every single activity in the room for a security purpose. Authentication process that is being used to identify the presence of a person in a room or building still vary. The process varies from writing a name and signatures in the attendance list, using an identity card, or using biometric methods authentication as fingerprint or face scanner.
In recent years, rapid advancements in computer vision and artificial intelligence have enabled the development of intelligent systems capable of automating routine tasks with high accuracy and efficiency. One such application is the Face Recognition Attendance System, which aims to modernize the traditional attendance management process by leveraging facial recognition technology. This system provides a reliable, contactless, and time- efficient solution for recording attendance in educational institutions and organizations.

Traditional attendance methods such as manual roll calls, signature-based registers, and identity card verification are often time-consuming, prone to errors, and susceptible to proxy attendance. These methods also require significant human effort and may lead to inaccuracies in record-keeping. To overcome these limitations, biometric-based attendance systems have gained popularity, among which face recognition stands out due to its non- intrusive nature and ease of implementation.

II. LITERATURE SURVEY

Face recognition technology has been widely studied and applied in various domains such as security systems, access control, surveillance, and attendance management. With the advancement of image processing and machine learning techniques, automated face recognition systems have gained significant attention due to their accuracy, efficiency, and non-intrusive nature.

Early face recognition systems were based on traditional image processing and statistical approaches such as Eigenfaces and Fisherfaces. These methods relied on dimensionality reduction techniques to extract facial features and were effective only under controlled lighting and pose conditions. However, their performance degraded significantly in real-world environments, limiting their practical use.

To overcome these limitations, texture-based methods such as Local Binary Patterns (LBP) and Local Binary Pattern Histogram (LBPH) were introduced. LBPH gained popularity due to its simplicity, low computational cost, and robustness to minor lighting variations. Many attendance management systems adopted LBPH combined with Haar Cascade classifiers for face detection because of their real-time performance and ease of implementation. Despite these advantages, LBPH-based systems struggle with variations in facial expressions, occlusion, and large datasets.

Recent research focuses on deep learning-based face recognition methods using Convolutional Neural Networks (CNNs). Models such as FaceNet, VGG-Face, and ArcFace learn highly discriminative facial embeddings and achieve superior accuracy compared to traditional techniques. These models perform well even under challenging conditions such as different poses, illumination changes, and partial occlusions. Several studies have demonstrated the successful deployment of deep learning-based face recognition systems for automated attendance in classrooms and workplaces.

In the context of attendance management, researchers have highlighted the benefits of face recognition systems in reducing manual effort, preventing proxy attendance, and maintaining accurate digital records. However, challenges such as dataset size, computational requirements, real-time performance, and privacy concerns are commonly discussed. Many systems address these challenges by using pretrained models, optimizing detection pipelines, and storing attendance data securely in electronic formats such as databases or spreadsheets.

III. SOFTWARE REQUIREMENTS AND SPECIFICATION

3.1 INTRODUCTION

3.1.1 Project Scope

The scope of the system is to reduce the time of the teacher as well as student which they wasted by doing traditional attendance.

3.1.2 User Classes and Characteristics

Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.

3.1.3 Assumptions and Dependencies

This document will provide a general description of our project, including user requirements, product perspective, and overview of requirements, general constraints. In addition, it will also provide the specific requirements and functionality needed for this project such as interface, functional requirements and performance requirement

FUNCTIONAL REQUIREMENTS

Functional user requirements may be high-level statements of what the system should do but functional system requirements should also describe clearly about the system services in detail.

3.2 EXTERNAL INTERFACE REQUIREMENTS

3.2.1 User Interfaces

The user interface for the software shall be compatible to any Android version by which user can access to the system. The user interface shall be implemented using any tool or software package like Android Studio, MYSQL etc.

3.2.2 Hardware Interfaces

Since the application must run over the internet, the hardware shall require to connect internet to the hardware which is android device for the system.

3.2.3 Software Interfaces

This system is a Single-user, multi-tasking environment. It enables the user to interact with the server and attain interact with the server to show the animal information also leaves a record in the inbuilt database. It uses Java and android as the front end programming tool and MySQL as the back end application tool.

3.2.4 Communication Interfaces

The e-store system shall use the HTTP protocol for communication over the internet and for the intranet communication will be through TCP/IP protocol suite.

NON FUNCTIONAL REQUIREMENTS:

3.2.5 Performance Requirements

- System can produce results faster on 2GB/4GB of RAM.
- It may take LESS time for peak loads at main node.
- The system will be available 100% of the time. Once there is a fatal error, the system will provide understandable feedback to the user.

3.2.6 Safety and Security Requirements

- The system is designed in modules where errors can be detected and fixed easily.

3.2.7 Software Quality Attributes

- Reliability: The Client machine will change the status of data indicating successful data transmission.
- Usability: The application should be easy to use through interactive interface.
- Maintainability: The system will be developed using the standard software development conventions to help in easy review and redesigning of the system.
- Support ability: The system will be able to support to transfer different types of SQL queries.

Portability: This software is portable to any system with the requirements specified. There must also be a server where the database can be set-up.

SYSTEM REQUIREMENT

3.2.8 Software Requirements Platform :

1. Operating System : Windows OS
2. Platform: Android Studio
3. Programming Language : PHYTON

3.2.9 Hardware Requirements

1. Processor: INTEL Pentium 4 Processor Core
2. 2. Hard Disk: 40 GB (min)
3. RAM: 256 MB or higher

3.3 BASIC FLOWCHART DIAGRAM

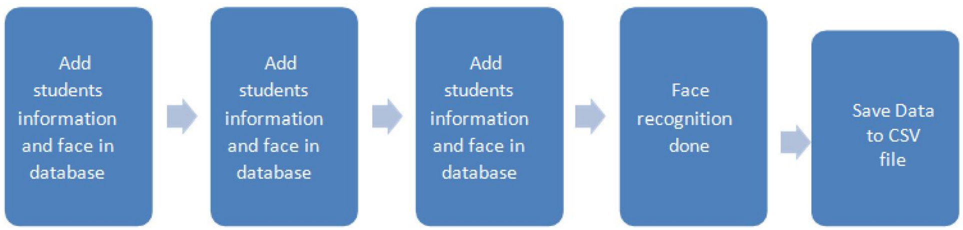
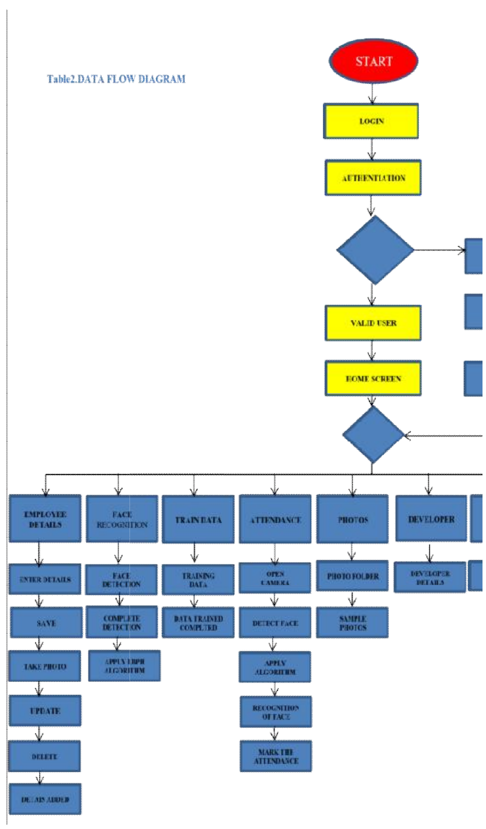


Fig 1. Flow chart



IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 13, April 2026
Impact Factor: 8.2

IV. SYSTEM DESIGN

HAAR CASCADE ALGORITHM

The core basis for Haar classifier object detection is the Haar-like features. These features, rather than using tShaevientDenastiaty values of a CSV pixel, use the change in contrast values between adjacentile rectangular groups of pixels. The contrast variances between the pixel groups are used to determine relative light and dark areas. Two or three adjacent groups with a relative contrast variance form a Haar-like feature. Haar-like features as shown in figure are used to detect an image. Haar features can easily be scaled by increasing or decreasing the size of the pixel group being examined. This allows features to be used to detect objects of various sizes. The cascading of the classifiers allows only the sub-images with the highest probability to be analyzed for all Haar-features that distinguish an object. It also allows one to vary the accuracy of a classifier. One can increase both the false alarm rate and positive hit rate by decreasing the number of stages. The inverse of this is also true. Viola and Jones were able to achieve a 90% accuracy rate for the detection of a human face using only 100 simple features. Detecting human facial features, such as the mouth, eyes, and nose require that Haar classifier cascades first are trained. In order to train the classifiers, this gentle AdaBoost algorithm and Haar feature algorithms must be implemented. Fortunately, Intel developed an open source library devoted to easing the implementation of computer vision related programs called Open Computer Vision Library (OpenCV). The OpenCV library is designed to be used in conjunction with applications that pertain to the field of HCI, robotics, biometrics, image processing, and other areas where visualization is important and includes an implementation of Haar classifier detection and training. Thus with help of this algorithm system will detect the person's face in the video. Face of the person gets Green Square as an indication of detection process. As soon as the face gets detected user can paused the video and enters the data of detected person such as person's name, address, profession, criminal record if any. If the detected person has criminal record then it can be defined as suspect. Check box option is given in the system where user can tick whether the person is suspect on not. This is the working of first module in which sample video is browsed and face is detected.

Fig 3.Haar Features

V. LOCAL BINARY PATTERN HISTOGRAM (LBPH)

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

The first computational step of the LBPH is to create an intermediate image that describes the original image in a better way, by highlighting the facial characteristics. To do so, the algorithm uses a concept of a sliding window, based on the parameters radius and neighbors.

5.1 Applying the LBP operation:

The first computational step of the LBPH is to create an intermediate image that describes the original image in a better way, by highlighting the facial characteristics. To do so, the algorithm uses a concept of a sliding window, based on the parameters radius and neighbors.

The image below shows this procedure:

DOI: 10.48175/IJARSCT-33974

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 13, April 2026
Impact Factor: 8.2

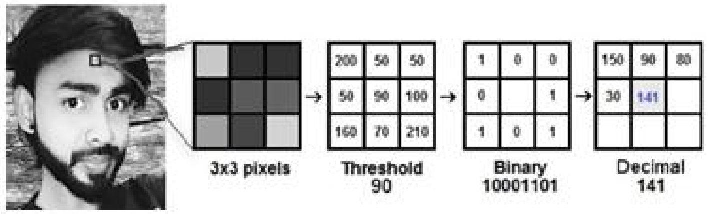


Fig 4.Applying the LBP operation

5.2. Performing the face recognition:

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

So to find the image that matches the input image we just need to compare two histograms and return the image with the closest histogram.

We can use various approaches to compare the histograms (calculate the distance between two histograms), for example: Euclideandistance, chi-square, absolutevalue, etc. In this example, we can use the Euclidean distance (which is quite known) based on the following formula:

So the algorithm output is the ID from the image with the closest histogram. The algorithm should also return the calculated distance, which can be used as a 'confidence' measurement. Note: don't be fooled about the 'confidence' name, as lower confidences are better because it means the distance between the two histograms is closer.

We can then use a threshold and the 'confidence' to automatically estimate if the algorithm has correctly recognized the image. We can assume that the algorithm has successfully recognized if the confidence is lower than the threshold defined

5.3 TRAINING THE ALGORITHM:

First, we need to train the algorithm. To do so, we need to use a dataset with the facial images of the people we want to recognize. We need to also set an ID (it may be a number or the name of the person) for each image, so the algorithm will use this information to recognize an input image and give you an output.

Images of the same person must have the same ID. With the training set already constructed, let's see the LBPH computational Steps.

5.4 APPLYING LBH OPERATIONS

The first computational step of the LBPH is to create an intermediate image that describes the original image in a better way, by highlighting the facial characteristics. To do so, the algorithm uses a concept of a sliding window, based on the parameters radius and neighbors.

IMPORTANT POINTS

- Suppose we have a facial image in grayscale.
- We can get part of this image as a window of 3x3 pixels.
- It can also be represented as a 3x3 matrix containing the intensity of each pixel (0~255).
- Then, we need to take the central value of the matrix to be used as the threshold.

5.4 EXTRACTING THE HISTOGRAM

Now, using the image generated in the last step, we can use the Grid X and Grid Y parameters to divide the image into multiple grids, as can be seen in the following image

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 13, April 2026
Impact Factor: 8.2

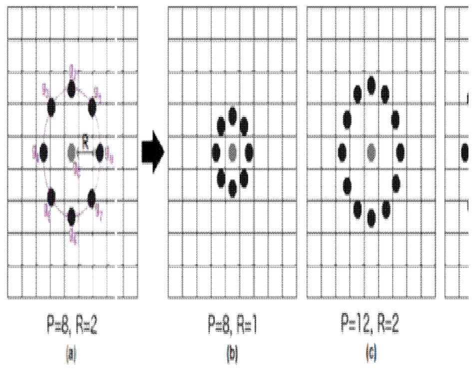


Fig.5 Radius of central pixel

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

• This value will be used to define the new values from the 8• As we have an image in grayscale, each histogram (from each grid) neighbors.

For each neighbor of the central value (threshold), we set a new binary value. We set 1 for values equal or higher than the threshold and 0 for values lower than the threshold. will contain only 256 positions (0~255) representing the occurrences of each pixel intensity.

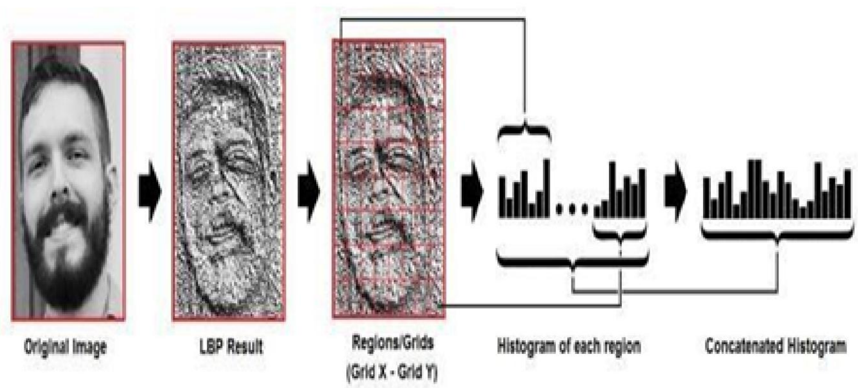


Fig 6. Histogram generation

• Then, we need to concatenate each histogram to create a new and bigger 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 image original image.

No table of figures entries found.

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

So to find the image that matches the input image we just need to compare two histograms and return the image with the closest histogram.

We can use various approaches to compare the histograms (calculate the distance between two histograms), for example: euclidean distance, chi-square, absolute value, etc. In this example, we can use the Euclidean distance (which is quite known) based on the following formula:

$$D = \sqrt{\sum_{i=1}^{n}(hist1_i - hist2_i)^2}$$

DOI: 10.48175/IJARSCT-33974

So the algorithm output is the ID from the image with the closest histogram. The algorithm should also return the calculated distance, which can be used as a 'confidence' measurement. Note: don't be fooled 'confidence' name, as lower confidences are better because it means the distance between the two histograms is closer.

• We can then use a threshold and the 'confidence' to automatically estimate if the algorithm has correctly recognized the image. We can assume that the algorithm has successfully recognized if the confidence is lower than the threshold defined.

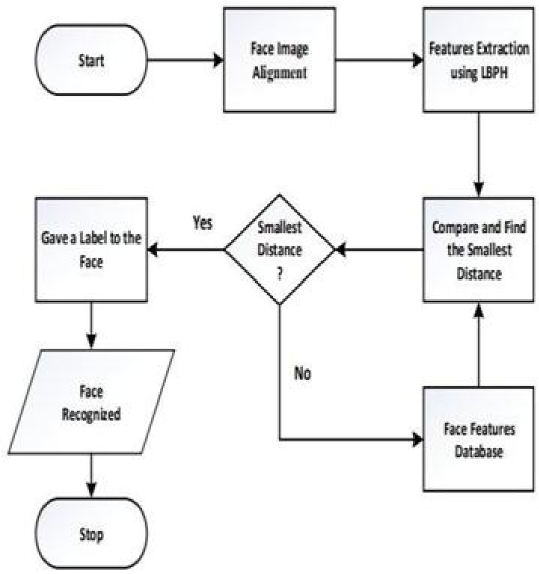


Fig 7. Face alignment and feature extraction

REFERENCES

1) G.Satyanarayana Reddy,Rallabandi Srinivasu,Srikanth Reddy Rikkula,Vuda Sreenivasa Rao," Management Information System To Help Managers For Providing Decision Making In An Organization", International Journal of Reviews in Computing.

2) Learning OpenCV –Computer Vision with the OpenCV Library O'Reilly Publication

3) Paul Viola, Michael Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features", IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Link: https://ieeexplore.ieee.org/document/990517

4) W. Zhao, R. Chellappa, A. Rosenfeld, P. Phillips, "Face Recognition: A Literature Survey", ACM Computing Surveys. Link: https://dl.acm.org/doi/10.1145/954339.954342

5) OpenCV Documentation – Face Recognition Link: https://docs.opencv.org/

6) Anil K. Jain, Arun Ross, Salil Prabhakar, "An Introduction to Biometric Recognition", IEEE Transactions on Circuits and Systems for Video Technology. Link: https://ieeexplore.ieee.org/document/1464434

7) Adrian Rosebrock, "Face Recognition with OpenCV, Python, and Deep Learning", PyImageSearch. Link: https://pyimagesearch.com

8) Kaggle – Face Recognition Datasets Link: https://www.kaggle.com/datasets

9) Ahonen, T., Hadid, A., Pietikäinen, M., "Face Description with Local Binary Patterns: Application to Face Recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI). Link: https://ieeexplore.ieee.org/document/4385780

10) Turk, M., Pentland, A., "Eigenfaces for Recognition", Journal of Cognitive Neuroscience, MIT Press. Link: https://ieeexplore.ieee.org/document/139758

11) Dalal, N., Triggs, B., "Histograms of Oriented Gradients for Human Detection", IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Link: https://ieeexplore.ieee.org/document/1467360
12) Zhang, K., Zhang, Z., Li, Z., Qiao, Y., "Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks (MTCNN)", IEEE Signal Processing Letters. Link: https://ieeexplore.ieee.org/document/7553523
13) Schroff, F., Kalenichenko, D., Philbin, J., "FaceNet: A Unified Embedding for Face Recognition and Clustering", IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Link: https://ieeexplore.ieee.org/document/7298682
14) King, D. E., "Dlib-ml: A Machine Learning Toolkit", Journal of Machine Learning Research (JMLR). Link: http://jmlr.org/papers/v10/king09a.html