International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

FACE RECOGNITION ATTENDANCE SYSTEM WITH MACHINE LEARNING

Skanda S[1]*, Darshan I S*[1]*, Ms. Keerthana H R*[2]

[1]Students,Department of Computer Applications,BMS College of Commerce and Management Bangalore, India
[2]Assistant professor, Department of Computer Applications,BMS College of Commerce and management,India

ABSTRACT :

Face recognition systems have gotten to be a significant portion of numerous businesses in our digital age. They are broadly utilized for security, confirmation, and character confirmation. Indeed, though face recognition isn't as exact as iris or unique finger impression recognizable proof, it's still exceptionally prevalent since it's contactless and non-invasive. One of the foremost commonsense employments of this innovation is in following attendance for schools, colleges, and companies. Conventional manual attendance frameworks are frequently moderate and difficult to oversee, driving to botches and wasteful aspects. Additionally, there's continuously the risk of somebody checking attendance for someone else. This has driven to a developing require for computerized attendance systems that use face recognition to create the method quicker and more precise. This proposed system points to form a class attendance system that relies on face recognition. The framework works in four primary steps: creating a database, recognizing faces, recognizing faces, and updating attendance. To begin with, we construct a database utilizing photographs of the students. These pictures will be used to distinguish students when marking attendance. Amid class, a camera captures live footage, and the system recognizes the faces of students within the classroom. It at that point compares these recognized faces with the stored pictures within the database to recognize each student. Once the system recognizes a face, it updates the attendance record consequently. At the conclusion of the class, an attendance report is produced and emailed to the pertinent staff member. This shrewd attendance system offers a few benefits. It spares time and exertion compared to manual attendance, anticipates proxy attendance, and gives a dependable record of who was present. Technologies like OpenCV and NumPy are fundamental for this system, as they empower productive image handling and face recognition. In brief, executing a face recognition-based attendance system may be a significant step forward for schools and organizations. It leverages present day innovation to form attendance following more effective and precise.

Keywords: Face Recognition Attendance System, OpenCV, Haar Classifier, Flask, KNN

1. Introduction :

This project is about an image-based attendance system for educational institutions. In this chapter, we will explain the project problem, research objectives, contribution to society, background in detail.

A) Problem Definition

The largest challenge with the current attendance management system is actually the accuracy of the data collected. This is because attendance may not be recorded personally by the person in question. In other words, the presence of a particular person may be recorded by a third party without the institution's knowledge, which may compromise the accuracy of the data. If the institution were to introduce enforcement, it may have to spend a lot of effort and time, which is not practical at all. As a result, the attendance recorded in the previous system is less reliable for analysis. The second drawback of the current system is that it takes too long to complete. Suppose a student record his/her attendance in a name list in about 1 minute. Only about 60 students can register their attendance in an hour, which is obviously inefficient and time-consuming. The third issue is whether legitimate stakeholders have access to this information. For example, most parents are very interested in tracking the exact whereabouts of their wards to ensure that they are attending their college or school classes regularly. But in the previous system, parents had little access to such information. Therefore, the current system needs to be further developed to maximize efficiency, data accuracy, and accessibility of information to authorized parties.

B) Research Objectives

In general, there are two main methods to recognize human faces that is feature-based and brightness-based. Feature-based methods use features of important points on the face, so-called landmarks, such as eyes, nose, mouth, edges or other distinctive features. This means that only parts of the image previously extracted are covered during the calculation process. On the other hand, brightness-based methods integrate and calculate all aspects of a given image, which are also known as holistic or image-based methods. The brightness-based method is more time consuming and also more complex since the entire image needs to be considered.

Various advancements are made during the process of this face recognition approach, but the two most important steps are face recognition and face recognition. First, an image of the student's face is needed to track attendance. This image can be taken using a camera installed in the classroom, in a position where the entire classroom can be seen. This image is used as input by the system. For accurate face recognition, the image needs to be enhanced using image processing techniques such as grayscale conversion and histogram equalization. After the image is enhanced, it is forwarded to face recognition. Face recognition takes place after the face identification process.

C) Significance and contribution

The main objective of this paper is to develop an intelligent attendance management system based on face recognition, which solves the problems of other automated systems currently in use. The main strategy is to compare the recent photos of students with some images that are purposely taken and stored in a database. If the images in the database match the real-time images, these images can be used to mark the attendance.

D) Objectives

Many of the current attendance management systems are inefficient and do not share information. Therefore, this project will address and further improve the following limitations:

- Students will be more conscientious in attending classes because they can only record their attendance directly and all absences will be recognized by the system. This not only trains students to be punctual but also avoids immoral behaviours such as signing their friends' attendance books.
- Institution can save a lot of resources as enforcement is now done by technology instead of human oversight, which means wasting a lot of human resources on non-essential processes.
- The application can run anywhere and, on any device, as long as there is a Wi-Fi or Ethernet connection, which makes the attendance system portable and can be placed anywhere else, for example, a device can be placed at the entrance of a classroom to record attendance.
- It saves a lot of money as all the paperwork is eliminated and it saves time as all the computation are automated.

2. METHODOLOGY :

We propose a cost-effective solution to detect student attendance using facial recognition technology. Our solution consists of four steps: image capture, facial recognition, attendance registration, and attendance monitoring. This method is designed to improve staff productivity, reduce their workload, and ultimately increase the accuracy of attendance records. Generally, fingerprints, retina scans, access cards, and other methods are used to record attendance. The strategy proposed in this paper uses a face recognition approach. The idea of Haar cascade and LBPH algorithm were used to identify faces in a given image. The goal is to record classroom attendance without teacher involvement. The advantage of Haar cascade over current techniques is its faster face recognition speed. A separate Haar cascade is created for each user. Furthermore, we collect positive photos or photos containing faces to train a classifier that creates a Haar cascade file. The same file can be used for face recognition. Thus, the Haar cascade offers the potential to be used for both general object identification and face recognition. Four implementation phases are proposed in this work. In the first step, we focus on face recognition images from a video collected for few seconds. Additionally, a grayscale image is created from each frame. The grayscale images are converted in step two and stored in the dataset to be trained in step 3. Finally, when an input image containing a face is provided, the trained model identifies both the face and the presence. A student ID is assigned to each individual face during the creation of the dataset.

A) Face Detection

The Primary Viola and Jones' main face recognition algorithm runs for over 150 frames. In order to adapt the original technique to various real-time applications, several developers and scientists have improved on it in recent years. We employ a technique to reduce the computational complexity by only applying the face recognition algorithm to the segmented regions with the background subtracted. Our face recognition approach is implemented using the wavelet transform. The wavelet coefficient subset serves as a representation of the object's shape. Integral images were used to compute Haar features. Rectangular features were created by computing the difference in variance between black and white regions. Integral images and squared integral image techniques are used to compute these features.

B) Database Creation

The first step in using an automatic attendance tracking system is to enroll each student in a class. It needs to be properly trained on the faces that need to be identified. So the first step in face recognition involved extracting the faces of all the required people from various photos and recording them as a dataset as 200x200 grayscale images. Assuming that x belongs to an entity, at this point a variety of photos containing x are provided as input. This is the first stage where the faces in the input photos are detected, identified and converted into grayscale images. After conversion, each file will be assigned a unique ID (student ID and USN) to better identify its identity. To improve the accuracy of face recognition, we need to train the faces of all members under different conditions.

C) Face Recognition

A single dataset is now available that contains all the members' faces under different circumstances. After training all these photos, a NumPy array was created. The trained classifier file was saved to label the test dataset taken from the class. The images representing each member of the class are used as input images. First, face detection techniques are used to find all the faces and then they are identified using Local Binary Pattern Histogram (LBPH). After the face is recognized, a grayscale image is created. The faces were also recognized using a trained classifier. Each identified face is tagged with a student ID tag, which further helps in tracking attendance.

3. ALGORITHMS :

A) LBPH:

Face recognition in computer science is essentially the task of identifying people based on their facial expressions. Its popularity has grown significantly over the past two decades, mainly due to new technologies and the superior quality of modern recordings and cameras. The Local Binary Pattern (LBP) texturing operator labels each pixel in an image by thresholding its surroundings and treating the result as a binary number. Furthermore, it has been shown that using LBP in combination with HOG descriptors can significantly improve detection performance on certain datasets. Using LBP in combination with histograms allows us to represent face images using simple data vectors. Since LBP is a visual descriptor, it can also be used for face recognition tasks, as explained in the step-by-step instructions below.

1. Parameters: LBPH uses four parameters: Radius: Radius is used to create the circular local binary pattern and represents the radius around the center pixel. Number of Neighbors: The number of sample points that create the circular local binary pattern. Grid X: The number of cells in the horizontal direction. Grid Y: The number of cells in the vertical direction.

2. Training the Algorithm: First we need to train the algorithm. To do this, we need to use a dataset that contains facial photos of people we want to identify. We also need to set a Student ID for each image so that the algorithm can identify the input image and return the result.

3. Application of LBP computation: The first computational stage of LBPH is to generate an intermediate image that more accurately describes the original image by emphasizing facial features. The algorithm does this using the concept of a sliding window based on parameters radius and neighborhood. Suppose we have a grayscale face image. We can take a portion of this image as a window of 3x3 pixels. It can also be represented as a 3x3 matrix containing the intensity (0 to 255) of each pixel. Now we need to use the central value of the matrix as the threshold. That's what we need to do next: set a new binary value for each neighborhood of the threshold. Our matrix now contains only binary values. Each binary value for each point in the matrix needs to be concatenated to a new binary value row by row. The central value of the matrix (actually the pixel in the original image) will be set to this binary value after being converted to a decimal value. At the end of this process (LBP technique), we obtain a new image that more accurately captures the features of the original image.

4. Extract the Histograms: Since we have a grayscale image, each histogram (from each grid) contains only 256 positions (0 to 255) that represent the occurrence of each pixel intensity. Next, we need to concatenate each histogram to create a new, larger histogram.

5. Perform face recognition: The algorithm has already been trained at this point. Each histogram created serves as a representation of one of the images in the training data set. So, given an input image, we repeat this process for a new image and create a histogram that represents the image.

It simply compares two histograms and returns the image with the most similar histogram, identifying the image that corresponds to the input image. The distance between two histograms can be calculated using various methods such as Euclidean distance, chi-squared, absolute value, etc.

Thus, the output of the algorithm will be the ID of the image with the most similar histogram. The algorithm should also return the calculated distance, which can be used as a measure of "confidence". A threshold and the "confidence" are then used to automatically estimate whether the algorithm has correctly recognized the image. If the confidence level is lower than the defined threshold, the algorithm can be considered successful.

B) HCC:

The Haar classifier or Haar cascade classifier is an object detection machine learning program that identifies objects in images and videos. The algorithm can be described in four steps:

- Calculating Haar Features
- Creating Integral Images
- Using Adaboost
- Implementing Cascading Classifiers

It is important to remember that, like any other machine learning model, this algorithm requires a large number of positive face images and negative non-face images to train the classifier.

1. Calculating Hair Features: Detecting hair features is the first step. In the detection window, the Haar features are actually the result of the calculation of adjacent rectangular sections. To calculate the total difference, we first need to add up the pixel intensities of each region. It can be difficult to identify these elements in a large photo. This is where integral images come into play, as they reduce the number of operations.
2. Creating an Integral Image: Without going into too much detail about the math behind it, integral images essentially speed up the computation of these Haar features. Instead of computing each pixel, we create sub-tangles and then create array references for each of those sub-tangles. These are then used to compute the hair properties.
3. AdaBoost Training: Essentially, Adaboost selects the most important features and trains a classifier to use them. The algorithm is able to detect objects using a "strong classifier" created by combining multiple "weak classifiers". Weak learners are created by sliding a window over the input image and computing the Haar features for each region of the image. This distinction is contrasted with a learned threshold that distinguishes between non-objects and objects. Because these are "weak classifiers", to create a strong classifier, you need many Haar features to be accurate.
4. Implementing a Cascade Classifier: Each layer of a cascade classifier consists of weak learners. It consists of a sequence of phases. By using boosting while training the weak learners, a highly accurate classifier can be built from the average prediction of all weak learners. The classifier decides to either proceed to the next region (negative) or indicate that an object has been identified based on this prediction (positive). Since the majority of the window does not contain anything of interest, phases are created to reject negative samples as soon as possible.

4. SOFTWARES USED :

A) OpenCV

OpenCV is a huge open-source library for computer vision, machine learning and image processing, which plays a vital role in real-time operations that are very important in today's systems. It can be used to process images and videos to identify objects, faces and even human handwriting. Combined with various libraries such as NumPy, Python can process and analyze OpenCV's array structures. It uses vector spaces to identify image patterns and their various features and performs mathematical operations on these features. The first OpenCV version was 1.0. OpenCV is released under the BSD license, so it is free to use for both academic and commercial use. It has C++, C, Python and Java interfaces and supports Windows, Linux, MacOS, iOS and Android. When developing OpenCV, the focus was on real-time applications for computational efficiency. With OpenCV, more complex tasks can be completed relatively easily.

B) Python:

Python is a programming language designed to help programmers of all experience levels easily translate concepts into working code. In the field of machine learning, it is the most widely used, most developed, and most supported programming language today. A high-level, general-purpose programming language, Python is a very popular programming language. The latest version of the Python programming language, version 3.12, is used for web development, machine learning applications, and all other cutting-edge software technologies. The Python programming language is perfect for beginners. The Python Imaging Library (PIL) is a free library that supports accessing, modifying, and saving various types of image files. Its last release was in 2009, but work on it has stalled. Luckily there's Pillow. This actively maintained fork of PIL is easy to install, compatible with Python 3, and runs on all major operating systems. The library includes basic image processing functions such as color space conversion, point operations, and filtering with a collection of built-in convolution kernels. One of the most popular libraries for computer vision applications is OpenCV (Open-Source Computer Vision Library). The Python API for OpenCV is called OpenCV. OpenCV-Python is not only fast because it consists of code written in C/C++ in the background, but it is also easy to code and deploy (due to the Python wrapper in the foreground). This makes it a great choice for running computationally intensive computer vision programs. Python offers a variety of options for developing Graphical User Interfaces (GUIs). Python combined with the Flask framework allows you to develop GUI apps quickly and easily. Flask provides an efficient object-oriented interface to a GUI toolkit. Using Flask, you can easily create GUI applications. Just follow these steps:

- Import the Flask framework.
- Build the flask app program.
- Include HTML and CSS from the list above in the Flask application.
- Enter the main event thread to take action against each event triggered by the user.

C) Flask:

Flask is a web framework that allows developers to build lightweight web applications quickly and easily using the Flask library. It is developed by Armin Ronacher, president of the International Group of Python Enthusiasts (POCCO). It is essentially based on the WSGI toolkit and the Jinja2 template engine. It provides a simple and elegant way to create web services, APIs, and dynamic websites. Flask allows you to define routes, handle requests, and render templates using Python code. Its minimalist design allows developers to focus on specific features without unnecessary overhead. Whether you're building a small project or prototyping, Flask's flexibility and ease of use make it a great choice. Features of Flask framework are:

- Microframework as it keeps its core simple and scalable.

- WSGI toolkit defines a common interface.
- Uses Jinja2 as its template engine.
- Design philosophy is Pythonic and conventional.

5. DATA FLOW :

The DFD for a facial recognition system is a diagram showing the overall data management of the project. It has five major levels that show the data processing of the facial recognition system, namely DFD levels 0, 1, 2, 3, and 4. These DFD levels show the concepts of facial recognition data management from basic to specific details.

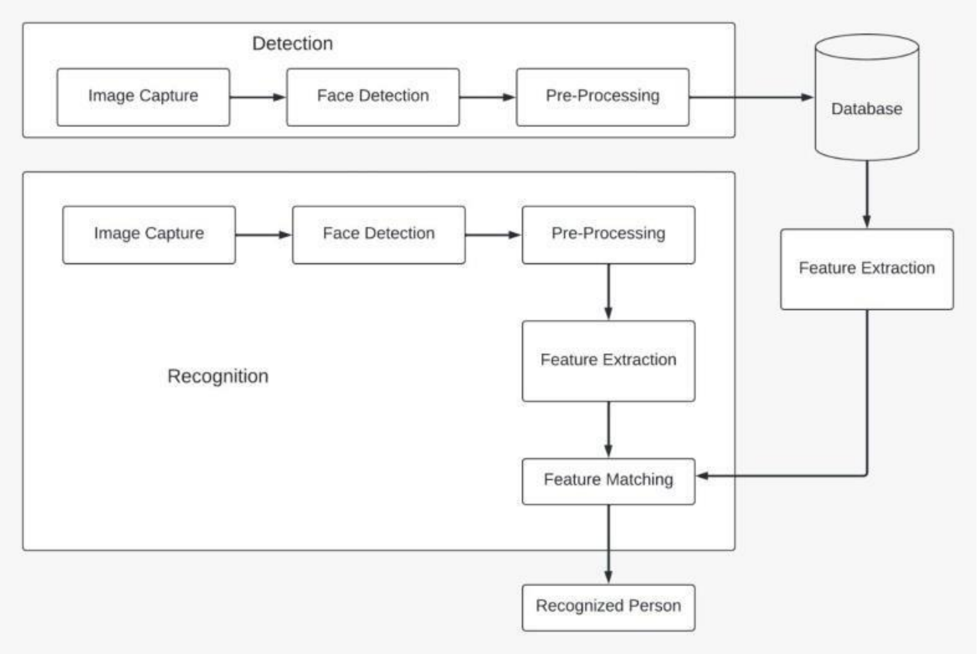


Fig. 1. System Architecture

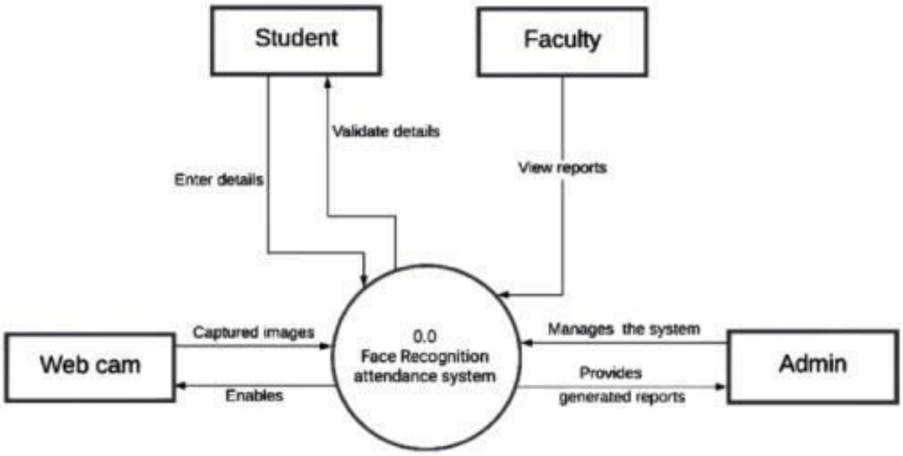


Fig. 2. Content Level

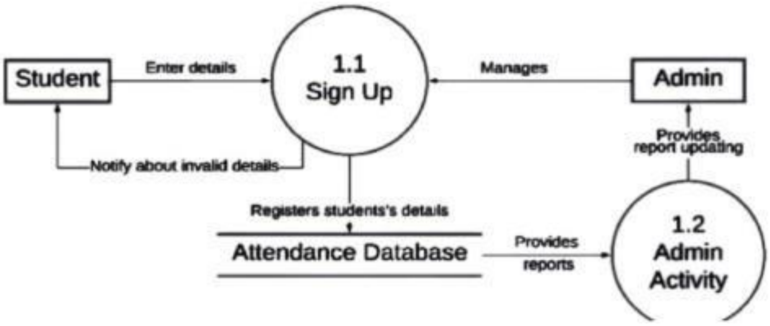


Fig.4 . Student Registration

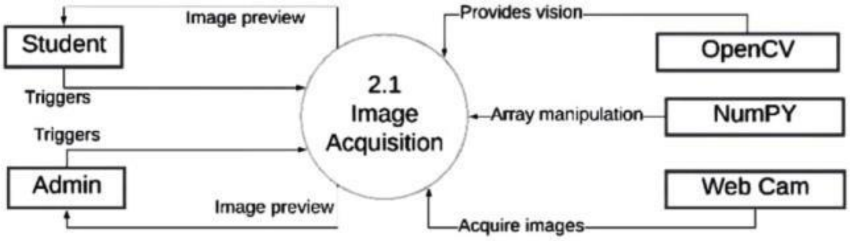


Fig. 5.. Image Acquisition

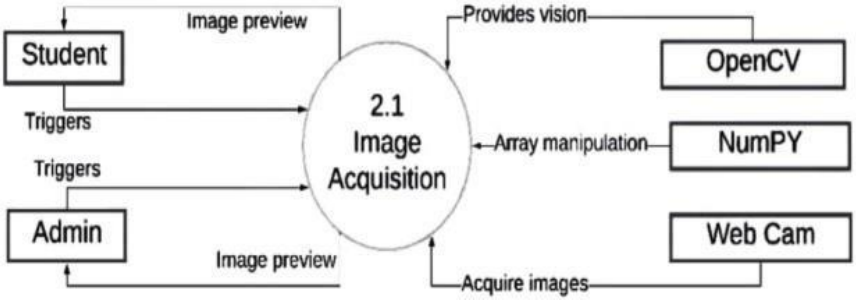


Fig. 6. Face Detection

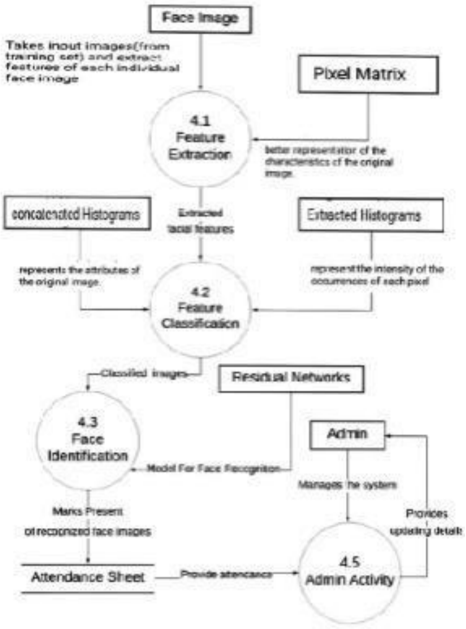
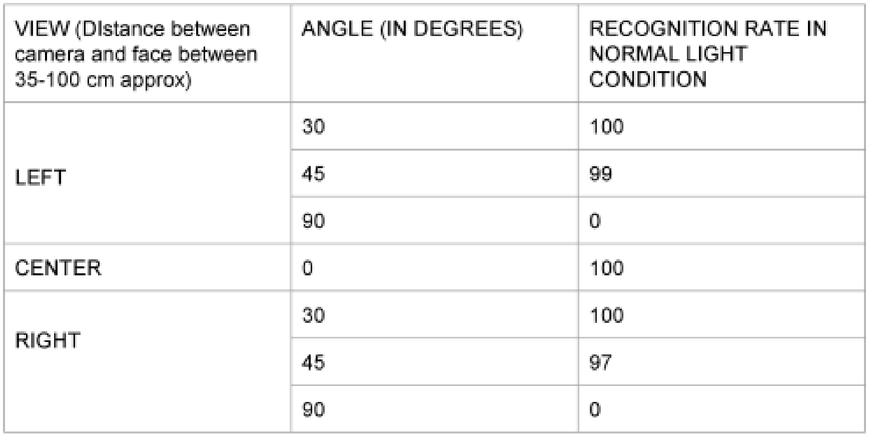

Face Image
Takes inout images(from training set) and extract features of each individual face image
Pixel Matrix
4.1
Feature
Extraction
better representator of the characteristics of the original image.
concatenated Histograms
Extracted
facial features
Extracted Histograms
represents the attributes of the original image.
represent the intensity of the occurrences of each pixel
4.2
Feature
Classification
Classified images
Residual Networks
4.3
Face
Identification
Model For Face Recognition
Admin
Manages the system
Marks Present
of recognized face images
Provides
updating details
Attendance Sheet
Provide attendance
4.5
Admin Activity

Fig. 7. Face Recognition

6. RESULTS AND DISCUSSION :

- The user can interact with the system through a GUI, where the user is mainly provided with three different options: student registration, faculty registration, and attendance evaluation.
- Students must fill in all the required details in the Student Registration Form. Once they click on the “Register” button, the webcam will start automatically.
- The webcam will capture 150 images, create an image dataset with the specified ID, and then automatically exit.
- When the image dataset is created, each student is assigned an ID number. If the test student's image is matched with a record during recognition, the student's information is time-stamped in an Excel attendance table. If the test student's image does not match a record, the student is not marked as present, and all unmatched students are marked as absent after a period of time.
- The following images show how the system performs when fed with datasets of different sizes. Here we compare three groups with two data each. Figure 8 shows a comparison of the detection rate of the system at different camera angles. Figure 9 shows a comparison of the training time and the number of images in the dataset. Finally, Figure 10 compares the recognition time required for the system to recognize n number of faces.



VIEW (DIstance between camera and face between 35-100 cm approx)	ANGLE (IN DEGREES)	RECOGNITION RATE IN NORMAL LIGHT CONDITION
LEFT	30	100
	45	99
	90	0
CENTER	0	100
RIGHT	30	100
	45	97
	90	0

Fig. 8. Recognition at different face angles

From the left, the detection rate from 0 to 45 degrees is 99-100%. Beyond 45 degrees the speed starts to decrease and reaches zero at a 90 degree angle. From the right, the detection rate from 0 to 45 degrees is 97-100%. Beyond 45 degrees the speed starts to decrease and reaches zero at a 90 degree angle. At 0 degrees the detection rate is 100%.

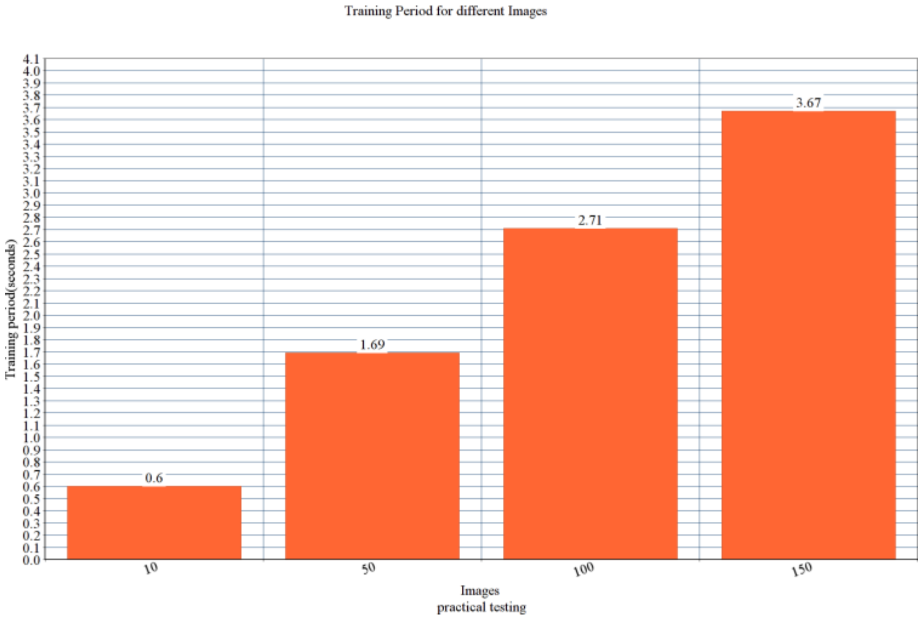


Fig. 9. Training Period of different images

The training time for 10 images is 0.6 seconds, for 50 images it is 1.69 seconds, for 100 images it is 2.71 seconds, and for 150 images it is 3.67 seconds. The detection time for one face is 1.1 seconds. Similarly, the detection times for 3 and 7 images are 1.4 and 1.8 seconds, respectively. The recognition time for 10 faces is around 2 seconds.

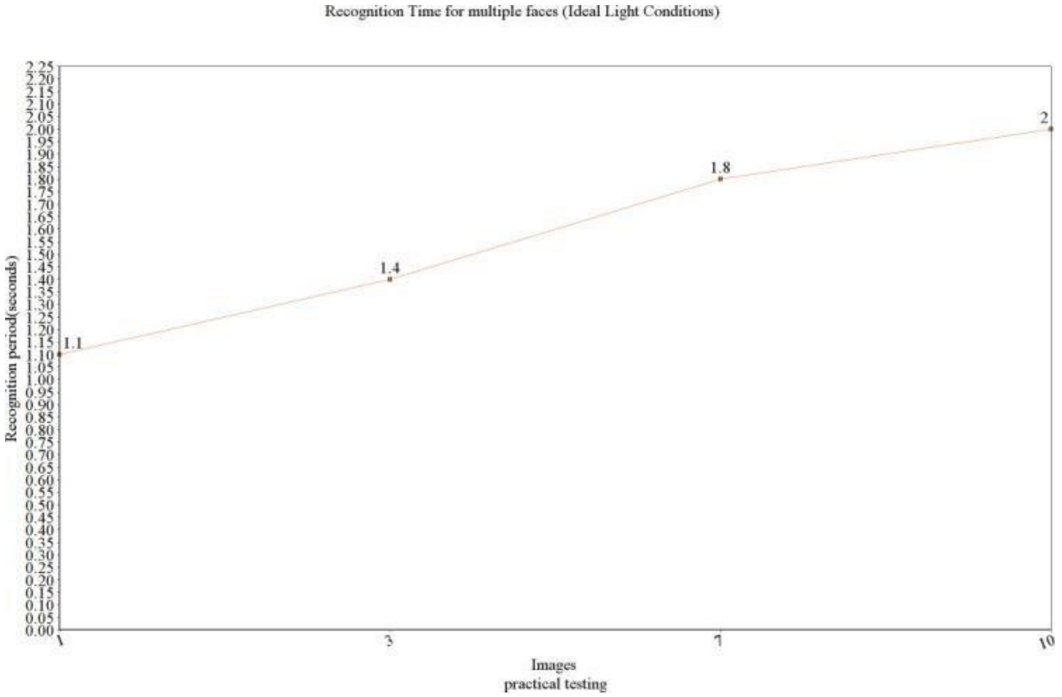


Fig. 10. Recognition for multiple faces

7. CONCLUSION :

In this system, an attendance system is implemented for a lecture, section or laboratory, allowing lecturers or teachers to record the attendance of students. This saves time and effort, especially when many students attend a lecture. The goal of an automated attendance system is to minimize the drawbacks of traditional (manual) approaches. The application of image processing technology in classrooms is demonstrated through this attendance system. This approach enhances the reputation of educational institutions and makes it easy to support attendance systems. Furthermore, the study aims to highlight the great potential of projects in the field of machine learning.

- Poor lighting in the classroom can affect the image quality and hence the performance of the system. This can be addressed at a later stage by improving the video quality or using algorithms
- Advanced processors can be used to improve the processing time of the images
- GSM can be used to send the attendance data of students to the respective parents
- The GUI can be made more interactive by allowing students to view the attendance confirmation data with necessary restrictions

REFERENCES :

1. Shreyak Sawhney, Karan Kacker, Samyak Jain, Shailendra Narayan Singh , Rakesh Garg, “Real-Time Smart Attendance System using Face Recognition Techniques”, 2020
2. Naman Gupta, Purushottam Sharma, Vikas Deep, Vinod Kumar Shukla, “Automated Attendance System Using OpenCV”, 2019
3. Krishna Mridha, Nabhan Tawjih Yousef, “Study and Analysis of Im- plementing a Smart Attendance Management System Based on Face Recognition Technique using OpenCV and Machine Learning”, 2021