



## FACE RECOGNITION ATTENDANCE SYSTEM

**Tanishq Sahay<sup>1</sup>, Sudarshan Srinivas<sup>1</sup>, Yash Agarwal<sup>1</sup>, Sudarshan V<sup>1</sup>, Mrs. Madhura J<sup>2</sup>**

<sup>1</sup>Students, Department of Information Science and Engineering Dayananda Sagar College of Engineering Bangalore, India

<sup>2</sup>Assistant Professor, Department of Information Science and Engineering, Dayananda Sagar College of Engineering Bangalore, India

Email: [tanishqsahay30@gmail.com](mailto:tanishqsahay30@gmail.com), [nattsudarshan@gmail.com](mailto:nattsudarshan@gmail.com), [Madhu-ise@dayanandasagar.edu](mailto:Madhu-ise@dayanandasagar.edu), [yashagarwal1301@gmail.com](mailto:yashagarwal1301@gmail.com), [sudarshanv13@gmail.com](mailto:sudarshanv13@gmail.com)

### ABSTRACT

In terms of time and staff workload, traditional methods for documenting student attendance in a classroom, such as roll-call and sign-in sheets, have been shown to be inefficient. Furthermore, they are vulnerable to human mistakes and phony attendance, both of which lead to inaccuracies in the data recorded. Research has been carried out to improve how we keep track of class attendance. Some of the recommended alternatives, however, are both pricey and impracticable. The majority of solutions also overlook the issue of phony attendance. This paper outlines a low-cost method for keeping track of student attendance. In this day of quickly growing new technologies, there is no reason why a critical educational practice like attendance should be considered in the traditional, monotonous way. It is challenging to manage big groups of students in a classroom using the traditional way. It is not advised since entering data into a system takes time and has a significant risk of error. Real-Time Face Recognition is a useful way of dealing with the daily attendance of a big number of pupils. Many algorithms like Haar Cascade Classifier for face detection and LBPH for face recognition have been used, but our suggested model, which is implemented in Python and the OpenCV library, determines the positive and negative features of the face for face recognition. The Tkinter GUI interface is used for user interface purposes.

*Index Terms* - Real-time Face Recognition, Haar Cascade Classifier, Python, OpenCV, Tkinter

## 1. INTRODUCTION

This is a project about Image Based Attendance System for Educational Institutions. In this chapter, the problem, research objectives, contributions to the society and the background information of the project will be discussed in detail.

### A) The Problem

The most serious issue, according to the previous attendance management system, is indeed the accuracy of the data collected. This is because attendance may not be recorded personally by the original person; in other words, a specific person's attendance may be taken by a third party without the institution's knowledge, which violates the accuracy of the data. If the institution establishes enforcement, it may have to waste a lot of human resources and time, which is not practical at all. As a result, all of the previous system's recorded attendance is unreliable for analysis. The previous system's second flaw is that it takes too long to complete. Assume a student signs his/her attendance on a name list in about 1 minute. Only about 60 students can sign their attendance in one hour, which is obviously inefficient and time consuming. The third issue is the legitimate interested party's ability to access that information. For example, most parents are very concerned about tracking their ward's exact whereabouts to make sure that they regularly attend college/school classes. However, in the previous system, parents had almost no access to such information. As a result, the previous system must be evolved in order to maximize efficiency, data accuracy, and information accessibility for those authorized parties.[1]

### B) Research Objectives

In general, there are two main methodologies for dealing with human face recognition: the feature-based methodology and the brightness-based methodology. The feature-based methodology makes use of key point features present on the face, known as landmarks, such as the eyes, nose, mouth, edges, or other distinguishing characteristics.[2] As a result, only a portion of the previously extracted image is covered during the calculation process. The brightness-based methodology, on the other hand, consolidates and computes all aspects of the given image. It is also referred to as holistic or image-based methodology. Because the entire picture must be considered, the brightness-based methodology requires more time and is also more complicated.

Various advancements are made during the process of this face recognition approach, but the two most important steps are face detection and face recognition. [3072] To begin, images of students' faces will be required to track attendance. This image can be captured using the camera, which will be installed in the classroom at a point where the entire classroom can be seen. This image will be used as input by the system.

For accurate face recognition, the image should be enhanced using image processing techniques such as gray-scale conversion and histogram equalization. The image will be passed to face detection after the image quality has been improved. Face recognition follows the face identification process.

### C) Significance And contribution

The primary goal of this paper is to create a smart attendance management system based on facial recognition that will address the issues that exist in other automated systems in use in today's day and age. The main strategy that must be taken is to compare a recent photo of a student to some images that were purposefully taken and stored in a database, which can then be used to mark attendance if the images in the database match the real-time image.

### D) Objectives

Many of today's attendance management systems are inefficient and lack information sharing. As a result, the following constraints will be countered and further improved in this project:

- Students will become more diligent about attending classes. This is because a student's attendance can only be taken personally, and any absentees will be detected by the system. This not only trains students to be punctual but also avoids immoral ethics such as signing attendance for their friends.
- The institution can save a lot of resources because enforcement is now done through technology rather than human supervision, which wastes a lot of human resources for a minor process.
- The application can run on any device in any location as long as there is Wi-Fi or an Ethernet connection, making the attendance system portable and able to be placed anywhere. For example, the device could be placed at the classroom's entrance to take attendance.
- It saves a significant amount of money because it eliminates all paperwork, and it saves time because all computations are automated.

## 2. METHODOLOGY

We propose a low-cost solution in recording student attendance by employing face detection technique. Our solution consists of four stages: image acquisition, face detection, attendance registration, and attendance monitoring. We named our proposed solution IBAS (Image Based Attendance System). The method is intended to increase staff productivity and decrease workload, which will ultimately increase the accuracy of the attendance record. In general, fingerprint, retinal scan, access cards, and other methods would be used to take attendance. The face recognition approach is used in the paper's suggested strategy. The idea of Haar cascades and the LBPH algorithm have been used to the task of identifying faces in a given image. The objective is to take attendance in a classroom without the teacher's involvement. The advantage of Haar cascades over current techniques is their quicker face detection speed. For each user, a separate Haar cascade will be produced. Further, gather positive or face-containing photos to train the classifier that creates the Haar cascade files. For face detection, the same files can be utilized. Thus, Haar cascades offer the chance to apply them for object identification in general as well as face recognition. In this work, four stages of implementation have been suggested. The first step, focuses on face detection images of video that was collected for two seconds. Additionally, grayscale images are created from each frame. The grayscale image is converted in stage two and stored in the dataset for training in stage three. Last but not least, if an input image with trained faces is provided, the faces and attendance are both identified. The student ID mentioned to each particular face while creating the dataset.

### A) Face Detection

The primary Viola and Jones face detection algorithm runs for 150 frames. To adapt the original techniques to the various real-time applications, several developers and academics have refined them over the past few years. By just applying the face detection algorithm to the segmented region after background subtraction, a method for reducing computational complexity is used. Our face detection approach is implemented using a wavelet transform. Wavelet coefficient subsets serve as a representation of the shape of the item. Integral pictures have been utilized to calculate the Haar features. By calculating the variance difference between the black and white regions, the rectangle feature values have been created. The technique of an integral image and squared integral image is used to calculate these features.

### B) Database Creation

The first step in using an automated system to keep track of attendance would be to register each student in the class. It is necessary to adequately train the faces that need to be identified. Therefore, using the first step of face detection, the faces of all the necessary people had to be extracted from the various photographs and recorded as a dataset as a grayscale image with dimensions of 200x200. Assuming that  $x$  belongs to a unit, the variety of photos that contain  $x$  are provided as input at this point. The faces in the input photographs are found because it is the first stage, and they are then identified and turned into grayscale images.

After the conversion, each file is labeled with a unique ID (Student ID and USN) to further recognize their identity. In order to improve the accuracy of face recognition, the faces of all members have to be trained with different conditions.

### C) Face Recognition

The faces of all members under various conditions are currently stored in a dataset. After all of these photos were trained, NumPy arrays were produced. In order to label the test dataset obtained from class, the trained classifier file has been saved. The image that represents every member of a class will be used as the input image. It begins by utilizing a face detection technique to find all of the faces and identifying them using a local binary pattern histogram (LBPH). After faces are recognized, grayscale images are created. The face has also been recognized using the learned classifier. Each identified face will get a Student ID label, which further aids in keeping track of attendance.

### 3. ALGORITHMS

#### A) LBPH:

Face recognition is essentially the task of identifying a person based on their facial appearance in computer science. In the past two decades, it has greatly increased in popularity, largely due to new techniques created and the excellent quality of the most recent recordings and cameras. The Local Binary Pattern (LBP) texturing operator labels each pixel in an image by thresholding its immediate surroundings and treating the result as a binary number. Furthermore, it has been discovered that using LBP in conjunction with HOG descriptors significantly enhances detection performance on specific datasets. We can express the images of faces using a straightforward data vector by using the LBP in conjunction with histograms. As LBP is a visual descriptor it can also be used for face recognition tasks, as can be seen in the following step-by-step explanation.

- 1) Parameters: the LBPH uses 4 parameters:
  - Radius: the radius is used to build the circular local binary pattern and represents the radius around the central pixel.
  - Neighbors: the number of sample points to build 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: We must first train the algorithm. We must use a dataset containing the facial photographs of the persons we wish to identify in order to accomplish this. In order for the algorithm to identify an input image and provide you with a result, we also need to set a Student ID for each image.
- 3) Applying the LBP operation: The initial computational phase of the LBPH is to produce an intermediate image that, by emphasizing the face features, more accurately describes the original image. The algorithm does this by utilizing a sliding window idea based on the radius and neighbors of the parameter. 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). The matrix's central value must then be used as the threshold, which is what we must do next. We establish a new binary value for each neighbor of the threshold value. The matrix will now only have binary values. Each binary value from each point in the matrix must be concatenated, line by line, into a new binary value. The central value of the matrix, which is actually a pixel from the original image, is then set to this binary value after being converted to a decimal value. At the conclusion of this process (the LBP technique), we obtain a new image that more accurately captures the traits of the original image.
- 4) Extracting the Histograms: As we have an image in grayscale, each histogram (from each grid) will contain only 256 positions (0-255) representing the occurrences of each pixel intensity. Then, we need to concatenate each histogram to create a new and bigger histogram.
- 5) Performing the face recognition: The algorithm has already been trained at this point. Each histogram produced serves as a representation of one of the training dataset's images. Therefore, given an input image, we repeat the process for the new image and produce a histogram that symbolizes the image.

Simply compare two histograms and return the image with the closest histogram to identify the image that matches the input image. The distance between two histograms can be calculated using a variety of methods, such as the Euclidean distance, chi-square, absolute value, etc.

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. 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.

#### B) HCC

A Haar classifier, or a Haar cascade classifier, is a machine learning object detection program that identifies objects in an image and video. The algorithm can be explained in four stages:

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

It's important to remember that this algorithm requires a lot of positive images of faces and negative images of non-faces to train the classifier, similar to other machine learning models.

- 1) **Calculating Haar Features:** Gathering the Haar features is the initial stage. In a detection window, a Haar feature is effectively the result of calculations on adjacent rectangular sections. In order to calculate the difference between the sums, the pixel intensities in each region must first be added together. Identifying these elements in a large photograph can be challenging. This is where integral images come into play because the number of operations is reduced using the integral image.
- 2) **Creating Integral Images:** Without going into too much of the mathematics behind it, integral images essentially speed up the calculation of these Haar features. Instead of computing at every pixel, it instead creates sub-rectangles and creates array references for each of those sub-rectangles. These are then used to compute the Haar features.
- 3) **AdaBoost Training:** In essence, Adaboost selects the top features and trains the classifiers to use them. The algorithm can detect objects by using a "strong classifier" that is made by combining several "weak classifiers." By sliding a window across the input image and computing Haar characteristics for each area of the image, weak learners are produced. This distinction is contrasted with a learnt threshold that distinguishes between non-objects and objects. Since these are "weak classifiers," creating a strong classifier requires a lot of Haar features in order to be accurate.
- 4) **Implementing Cascading Classifiers:** Each level of the cascade classifier is made up of weak learners. It consists of a sequence of phases. A highly accurate classifier can be created from the mean prediction of all weak learners by employing boosting during the training of weak learners. The classifier either chooses to go on to the subsequent region (negative) or decides to indicate that an object was identified (positive) based on this prediction. Stages are made to reject negative samples as quickly as possible because the bulk of the windows don't contain anything of interest.

---

#### 4. SOFTWARES USED

##### A) MySQL:

The open-source MySQL server relational database management system is a key component of web-based applications. Due to the fact that data is stored and transferred via the internet, databases and related tables are a common component of many websites and applications. For data operations including querying, sorting, filtering, grouping, editing, and combining the tables, a MySQL server is needed.

A Python driver called Python MySQL Connector aids in the union of Python with MySQL. The conversion of data types between Python and MySQL is possible using this Python MySQL library. The MySQL Connector API is developed entirely in Python and doesn't need any external libraries.

This paper used a table called student containing columns like Student ID (Primary Key), Name, Department, Course, Year, Semester, Division, Gender, DOB, Mobile No, Address, USN (Unique, Non-NULL), Email, Teacher Name, Photo Sample.

##### B) OpenCV:

OpenCV is the huge open-source library for computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today's systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human. When integrated with various libraries, such as NumPy, python is capable of processing the OpenCV array structure for analysis. To identify image pattern and its various features we use vector space and perform mathematical operations on these features. The first OpenCV version was 1.0. OpenCV is released under a BSD license and hence it's free for both academic and commercial use. It has C++, C, Python and Java interfaces and supports Windows, Linux, MacOS, iOS and Android. When OpenCV was designed the main focus was real-time applications for computational efficiency. OpenCV enables us to do even more complex tasks relatively easy. There are for example routines, which detect face(s). The following sequence of commands does just that.

```

face_cascade=cv2.CascadeClassifier
( C
:\Users\...\haarcascade_frontalface_
default. x m l )
eye_cascade=cv2.CascadeClassifier
( C :\Users\...\haarcascade_eye. x m l
)img=cv2.imread( lena . j p g )
gray=cv2.cvtColor
(img, cv2.COLOR_BGR2GRAY)
faces=face_cascade.detectMultiSca
le (gray, 1.3, 5)
for (x,y,w,h) in faces:
cv2.rectangle (img, (x,y) , (x+w,y+h) ,
(255,0,0) , 2)
roi_gray = gray[y:y+h,

```

### C) Python:

Python is a programming language designed to make it simple for programmers of all levels to translate concepts into working code. In the field of machine learning, it is now the most widely used, developed, and well-supported programming language. High-level, all-purpose Python is a very well-liked programming language. The most recent version of the Python programming language, version 3.9, is utilized for web development, machine learning applications, and all other cutting-edge software technologies. For beginners, the Python programming language is ideal. The Python Imaging Library (PIL) is a free library that enhances support for accessing, modifying, and saving a wide range of image file types. Its last release was in 2009, but work has stalled. Thankfully, Pillow exists. This actively-maintained derivative of PIL is simpler to install, compatible with Python 3, and operates on all major operating systems. The library includes fundamental image processing features like color space conversions, point operations, and filtering with a built-in collection of 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 since the background consists of code written in C/C++ but is also easy to code and deploy (due to the Python wrapper in the foreground). This makes it a great choice to perform computationally intensive computer vision programs. Python provides various options for developing graphical user interfaces (GUIs). The default GUI library for Python is called Tkinter. The combination of Python and Tkinter makes it quick and simple to develop GUI apps. An effective object-oriented interface for the Tk GUI toolkit is provided by Tkinter. It's simple to build a GUI application using Tkinter. All you have to do is carry out the subsequent actions:

- Import the Tkinter module.
- Build the GUI program's primary window.
- Include a widget from the list above in the GUI application.
- Enter the main event loop to take action against each event triggered by the user.

---

## 5. DATA FLOW

The Face Recognition System DFD is a diagram used to show the overall data management of the project. It has 5 main levels that shows the Face Recognition System data handling which is the DFD Level 0, 1, 2,3 and 4. These DFD levels illustrate the Face Recognition data management concept from the basics up to specific details.

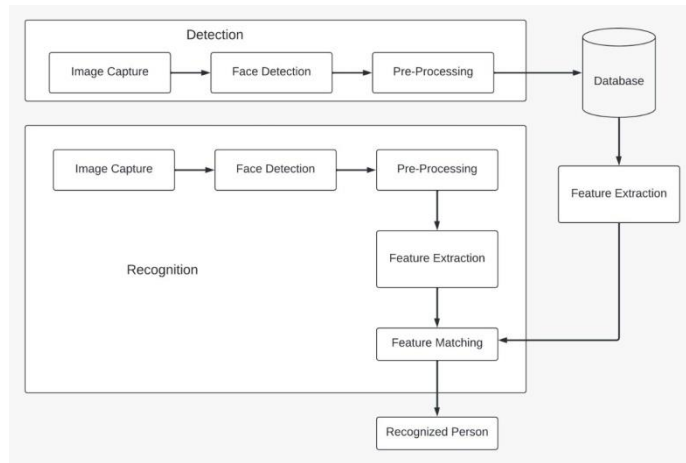


Fig. 1. System Architecture

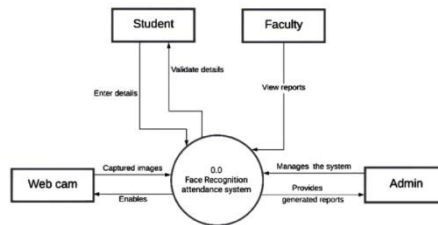


Fig. 2. Context Level

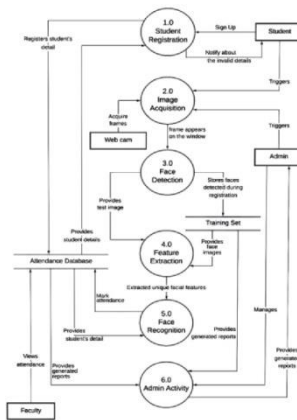


Fig. 3. Level 0

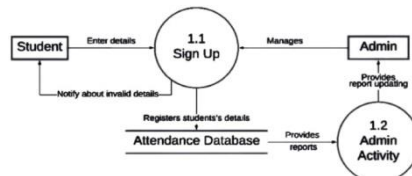


Fig. 4. Level 1 (Student Registration)

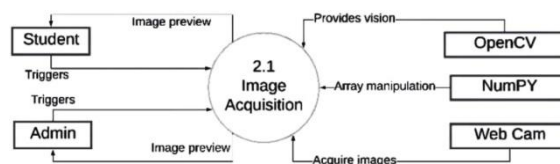


Fig. 5. Level 2 (image acquisition)

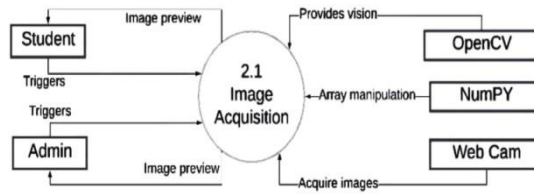


Fig. 6. Level 3 (Face detection)

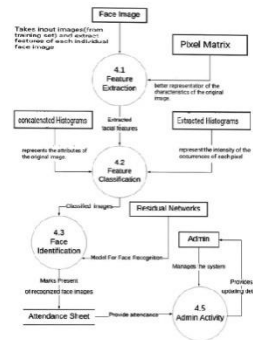


Fig. 7. Level 4 (feature extraction and face recognition)

6. RESULTS AND DISCUSSION

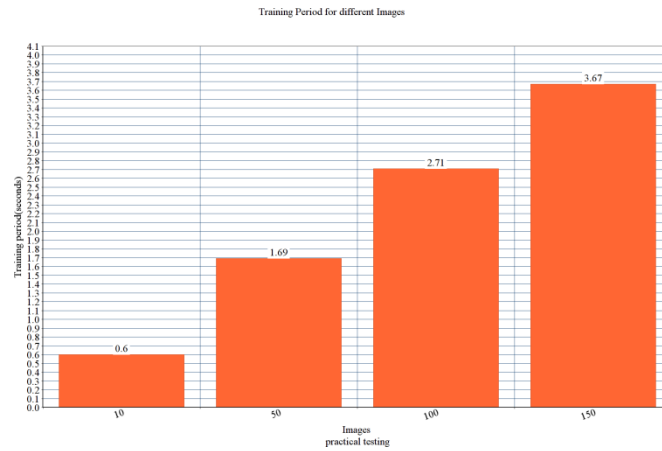
- The users can interact with the system using a GUI. Here, users will be mainly provided with three different options such as, student registration, faculty registration, and mark attendance.
- The students are supposed to enter all the required details in the student registration form. After clicking on the register button, the webcam starts automatically.
- The webcam will capture 150 images in-order to create the image dataset and then terminate automatically
- At the time of forming the image dataset, each student will get designated using an id number. While recognition, when the test student image matches with the dataset then the details of the student in the attendance excel sheet will be marked with a time-stamp, if the test student image does not get matched with the dataset, then it will not be marked present and all the unmatched student will be marked as absent after a certain period of time.

The following Images shows the nature of the system when it is fed with different size of datasets. Here we compare 3 groups of 2 data. Fig-8 depicts the comparison between recognition rate of the system with different camera angles. Fig-9 depicts the comparison between the training time and the number of images in the data set. And, finally, Fig-10 compares the recognition time it takes 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

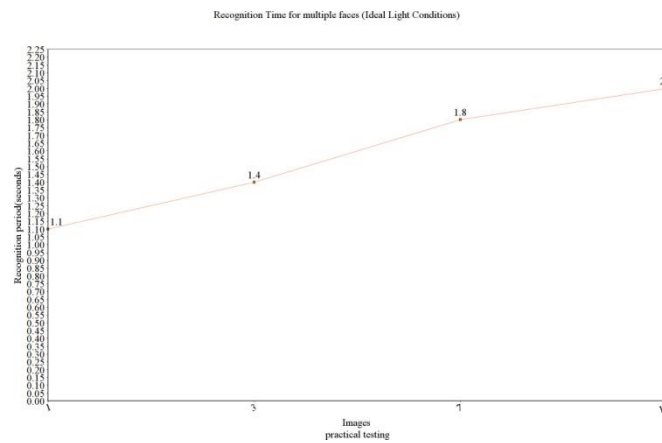
Fig. 8. Recognition Rate at different face angles

For Left view, between,0 and 45 degrees the recognition rate is 99-100 percent. after 45 degrees the rate starts decreasing and goes to zero at 90 degrees angles. For Right view, between,0 and 45 degrees the recognition rate is 97-100 percent. after 45 degrees the rate starts decreasing and goes to zero at 90 degrees angles. And, for 0 degrees, the recognition rate is 100 percent.



**Fig. 9. Training Period of different images**

the training period to train 10 images is 0.6 seconds, 50 images is 1.69 seconds, 100 images is 2.71 seconds and 150 images is 3.67 seconds. The recognition time for a single



**Fig. 10. Recognition Time for multiple faces**

face is 1.1 seconds. Similarly, the recognition time for 3 and 7 images are 1.4 and 1.8 seconds respectively. and, the recognition time for 10 faces is approximately 2 seconds

## 7. CONCLUSION

In this system, an attendance system is implemented for a lecture, section or laboratory by which the lecturer or teacher can record students' attendance. It saves time and effort, especially if there are a lot of pupils in the lecture. The goal of the automated attendance system is to minimize the shortcomings of the conventional (manual) approach. The application of image processing techniques in the classroom is demonstrated through this attendance system. This technique can enhance an institution's reputation in addition to simply assisting with the attendance system. The study also aims to highlight the project's enormous potential in the field of machine learning.

- The bad lighting in the classroom can occasionally have an impact on image quality, which negatively impacts system performance. This can be remedied in the latter stages by enhancing the video quality or employing algorithms.
- Advanced processors can be used to improve processing time of images
- GSM can be used to send attendance details of student to their respective parents
- The GUI can be made more interactive by allowing students to check their attendance details with necessary limitations



---

**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 Implementing a Smart Attendance Management System Based on Face Recognition Technique using OpenCV and Machine Learning”, 2021
- [4] Organization, S. D. I. W. C., Kuster, B. (2018). Face Detection and Face Recognition in Python Programming Language. The Seventh International Conference on Informatics and Applications (ICIA2018).
- [5] Emami, Shervin Suci, Valentin. (2012). Facial Recognition using OpenCV. Journal of Mobile, Embedded and Distributed Systems.
- [6] Yuvaraj, C. Srikanth, M. Kumar, V. Srinivasa Murthy, Y.V. Koolagudi, Shashidhar. (2017). An approach to maintain attendance using image processing techniques. 1-3. 10.1109/IC3.2017.8284353.
- [7] Nirmalya Kar, Mrinal Kanti Debbarma, Ashim Saha, and Dwijen Rudra Pal. Study of Implementing Automated Attendance System Using Face Recognition Technique. International Journal of Computer and Communication Engineering, Vol. 1, No. 2, July 2012.
- [8] Suma S L, Sarika Raga, “Real Time Face Recognition of Human Faces by using LBPH and Viola Jones Algorithm,” International Journal of Scientific Research in Computer Science and Engineering, Vol.6, Issue.5, pp.6-10, 2018.