BAUSTJ
BAUST Journal

Automatic Attendance System Using Face Recognition

Jahangir Alam SM[1*], Fahim Adnan[2], Torikul Islam Oli[3*], Md Khalid Syfullah[4*], Saad Ahmed[5*], Rohul Amin[6], and Md. Shameem Reza[7]

[1,2,3,4,5,6,7]Department of CSE, Bangladesh Army University of Science and Technology (BAUST), Saidpur, Bangladesh

emails: *[*1]jahangirxmu@gmail.com; [4]khalidsyfullah3@gmail.com; [5]saadahmed@baust.edu.bd; and [6]rohulamin@baust.edu.bd ('*' - corresponding author)*

ARTICLE INFO

Article History:

Received: 19th January 2025
Revised: 28th March 2025
Accepted: 30th April 2025
Published online: 7th May 2025

Keywords:

Face Recognition
Single Shot Detection
Deep Learning
Face detection
OpenCV

ABSTRACT

The face serves as a key representation of one's identity, making it an ideal biometric for applications in security and automation. One such application is attendance tracking in educational institutions, where traditional methods such as calling out names or using sign-in sheets can be time-consuming and error-prone, especially in large classrooms. These conventional approaches often consume 5 to 10 minutes of valuable class time, reducing instructional efficiency. Motivated by the need for a faster, more reliable, and less intrusive system, it has been developed an automated student attendance system based on face recognition technology. This system leverages advancements in computer vision and deep learning to streamline attendance management. It works by detecting faces in video frames, extracting facial features (embeddings), and recognizing students by comparing these embeddings with pre-stored records. it has been using OpenCV, a widely-used Python library for image processing, along with a Single Shot Detector (SSD) to detect and extract facial regions from each frame. During pre-processing, input images are scaled appropriately to retain crucial details. Once a student's face is recognized, their attendance along with their name, ID number, and timestamp is automatically recorded. By integrating face recognition into the attendance process, our system significantly improves accuracy, saves time, and enhances the overall efficiency of classroom management.

1. INTRODUCTION

To optimize performance, this proposed method restricts test and training images to frontal and upright facial images containing only a single face. To ensure consistent quality, both test and training photographs must be captured using identical devices. Students must register in the database to be recognized by the system, with enrollment possible through a user-friendly interface.

A facial recognition system is technology capable of identifying or verifying an individual from a digital image or video frame. The human face serves as a unique representation of personal identity. Facial recognition is therefore a biometric approach that identifies individuals by comparing their database-stored images with those captured in real-time (Bindu *et al.*, 2020). Contemporary facial recognition technologies are widely implemented due to their user-friendliness and remarkable capabilities. For instance, the FBI and airport security systems employ facial recognition to monitor drug activity, locate missing children, and identify suspects in criminal investigations. Facebook, a prominent social networking platform, incorporates facial recognition technology to enable users to tag friends in photographs (Sakshi, 2018). Similarly, Intel Corporation allows customers to access online accounts via facial recognition (Arun, 2017), while Apple enables users to unlock their iPhone X using face recognition (Nirmalya, 2017).

Facial recognition research began in 1960 with Woody Bledsoe, Helen Chan Wolf, and Charles Bisson introducing a method requiring administrators to identify facial features such as eyes, ears, nose, and mouth from photographs. Subsequently, the ratios and distances between these identified features and standard reference points were calculated and compared. In 1970, Goldstein, Harmon, and Lesk further advanced this research by automating the identification process using additional criteria such as lip thickness and hair color. Initially proposed principal

component analysis (PCA) as a solution to facial recognition challenges (Akshara, 2017).
Traditional student attendance methods—including roll calls and ID card verification—present numerous challenges. These approaches can interrupt teaching and distract students, especially during examinations. Moreover, circulating attendance sheets in large classes creates logistical difficulties, consuming valuable time and causing further disruptions (Ajinkya, 2014). Our proposed facial recognition-based attendance system addresses these issues by simplifying the process and minimizing distractions (Rahmat, 2020). This system eliminates the need for manual sign-ins, which can burden and distract students. The technology automatically identifies students, marking them present without interrupting class flow, while generating comprehensive attendance records that eliminate the need for manual counting or verification.

The system efficiently calculates attendance, saving instructors time and effort. Additionally, the automated facial recognition system helps prevent fraudulent attendance practices. Instructors no longer need to repeatedly count students, as the system records attendance in real-time. For optimal effectiveness, the system must identify students quickly within specific timeframes to avoid omissions. The facial features used for identification must remain consistent despite variations in background, lighting, pose, and facial expression. Consequently, system performance will be evaluated based on student recognition accuracy and image processing speed, ensuring reliable and efficient attendance tracking.

The proposed system aims to significantly reduce paperwork, eliminating the need for manual attendance recording. By automating this process, the system not only saves time but also ensures that attendance data is retrieved quickly and accurately. The system follows a series of key steps to complete the task, each of which is crucial for ensuring precise and reliable attendance tracking. These steps include image acquisition, face detection (Mayur, 2020), feature extraction, face recognition, and attendance recording (Mekala, 2019). With the new system, attendance (Abdelfatah, 2015) is captured through facial recognition, ensuring that data is accurate and minimizing the potential for errors. This eliminates the need for paper-based records and manual attendance marking, making the process more efficient. By automating attendance, the system helps save valuable class time that would otherwise be spent on manual roll calls. In essence, the system improves the overall efficiency of attendance tracking, providing several benefits over traditional methods. It streamlines the entire process, allowing instructors to focus more on teaching rather than administrative tasks. Additionally, by reducing time spent on attendance, it enhances the overall classroom experience. The primary objective of this project is to develop a fully automated student attendance system based on face recognition.

As it can be seen from Figure 1, To ensure that the attendance registration process is faster than with the previous system; to possess sufficient memory capacity to maintain the database; to identify facial components in video frames; to accurately recognize individuals' faces using the face database (Kowsalya, 2019) to classify features for identifying detected faces; to record attendance for recognized students; to create a database for attendance management; to provide both administrators and non-administrators with a user-friendly interface to access the attendance database; and to clearly communicate to users, via audio feedback, whether the face recognition process was successful (Shubham, 2021).

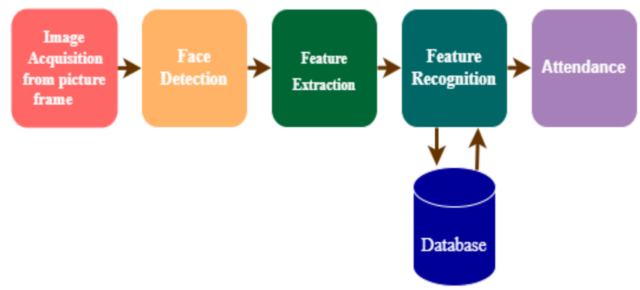

Figure 1: Block diagram of the general framework.

The proposed algorithm is designed to process images containing only a single face. When multiple faces appear in an image, each face is diminished in size, resulting in smaller facial regions. This reduction can lead to imprecise feature extraction, which adversely affects the system's performance. Consequently, the system will not perform facial recognition when it detects more than one face in an image. Furthermore, the Local Binary Pattern (LBP) algorithm, utilized for feature extraction, is particularly sensitive to image quality and can be significantly compromised by blurred or low-resolution images. LBP functions as a texture-based descriptor that extracts local grayscale features from small regions across the image, making it essential that both test and training images maintain consistent quality. To achieve high accuracy, test and training images must be captured using identical devices, ensuring uniformity in image quality and feature consistency. This approach details a facial recognition-based automated student attendance system. The proposed method identifies individuals by comparing input images, captured from video frames, with pre-trained images stored in the database. The system can detect and localize faces in the input facial image obtained from the video stream. Additionally, during the pre-processing stage, the system enhances image contrast and minimizes the effects of varying illumination conditions. Facial features are extracted using the LBP algorithm, which helps stabilize the system by providing consistent results. The proposed system achieves 90% accuracy for high-quality images and 82.31% accuracy for low-quality images.

The motivation behind this work is to address the inefficiencies of traditional attendance systems, which are typically time-consuming, prone to human error, and vulnerable to manipulation. With advancements in computer vision and deep learning, there exists a significant opportunity to introduce automated, non-intrusive, and reliable solutions in educational settings. By leveraging facial recognition technology, The aim is to create a seamless and intelligent system that not only saves educators' time but also ensures accurate record-keeping, ultimately contributing to smarter classroom management and enhanced student engagement.

2. LITERATURE REVIEW

In this system, the SSD is employed for facial detection. A camera strategically positioned in the classroom captures students' facial features, and the detected faces are extracted for subsequent processing. The database stores a dataset of 10 images per student, which is used to compare input faces with stored images during facial recognition (Yuslinda, 2018). The Local Binary Pattern Histogram (LBPH) method facilitates the facial recognition process (Patil, 2014). The procedure follows four essential image processing steps: first, LBPH extracts the image's histogram by dividing it into local regions. These histograms are then combined to create a comprehensive face descriptor. The system calculates the distance between the biometric data of the input (probe) image and the stored (trained) image. Recognition occurs when the calculated distance falls below a predetermined threshold. Upon successful recognition, the student's name is automatically recorded in an Excel spreadsheet to document their attendance.

It was identified limitations in several biometric systems used for automatic attendance tracking, including Radio Frequency Identification (RFID), fingerprint, and iris recognition systems (Arun, 2017). While the RFID card system is straightforward and simple to implement, it is vulnerable to fraudulent activity since users can check in for friends by using their ID cards as shown in Table 1. This significantly increases the potential for attendance fraud. Fingerprint recognition, despite its widespread use, presents efficiency challenges. Although effective, the verification process is time-consuming, requiring users to form lines and undergo individual verification. This undermines the time-saving purpose of automated systems, making fingerprint recognition less suitable for time-sensitive environments such as classrooms. Iris recognition offers superior accuracy due to the intricate patterns present in the iris; however, it raises privacy concerns as it captures more detailed biometric information. It is also more invasive compared to alternatives like facial recognition. Facial recognition (Senthamil, 2014), while capturing less detailed information than iris scanning, provides an optimal balance between accuracy and user convenience. It requires no direct physical contact and is less intrusive, making it an appropriate solution for student attendance systems. Despite the availability of alternatives such as voice recognition, these methods generally demonstrate lower accuracy and reliability compared to other biometric approaches. Consequently, facial recognition emerges as the recommended option for implementing a more efficient, secure, and user-friendly student attendance system.

Face detection refers to the process of identifying a person's face within digital images (Anagha, 2021). Applications designed for face detection utilize algorithms to determine if an image is a positive image (containing a face) or a negative image (lacking a face). To achieve high accuracy, these algorithms require extensive training on large datasets comprising hundreds of thousands of facial and non-facial images (Himanshu, 2018). Once trained, the algorithms can analyze an image and answer two key questions: i) Does this image contain any faces? ii) If so, where are they located?

Table 1
Advantages and disadvantages of different biometric systems.

System type	Advantages	Disadvantages
RFID card system	Simple	Fraudulent usage
Fingerprint system	Accurate	Time-consuming
Iris recognition system	Accurate	Less accurate
Voice recognition system	Accurate	Privacy invasion

This system is designed to detect human faces in images or video footage by identifying facial features from input images supplied to the system. Currently, it can be utilized a webcam to capture input images; however, it was planned to incorporate high-quality cameras in the future to achieve enhanced image resolution and clarity. For accurate face detection, it is essential to define a general structural framework of human faces. Common facial features such as eyes, nose, mouth, chin, and forehead serve as the foundation for identifying facial presence within an image (Akash, 2021). The primary objective of face detection is determining the position and dimensions of a face within an image. Once detected, faces are transferred to facial recognition algorithms for further processing, enabling various applications including identification and authentication (Anitha, 2020).

Recent studies have continued to refine and enhance the performance of face recognition-based attendance systems through more advanced detection and recognition methods (Ankur, 2020). It was developed a smart attendance system using RetinaFace (Pratik, 2016) for face detection and FaceNet for recognition, achieving a high accuracy rate of 99.11% with efficient processing speeds (Warman, 2023). Similarly, it was employed MTCNN for face detection combined with DeepFace and a FaceNet-512 pretrained model, demonstrating robust accuracy and performance across various conditions (Essien, 2023). It was focused on the educational context in Nigeria, designing a smart attendance system that achieved 100% face recognition accuracy while also mapping courses to lecturers based on specialization (Musa, 2023). It was introduced improvements to real-time face detection (Debadrita, 2021) in classroom environments through the use of RetainNet, incorporating face filtering and alignment to increase robustness and precision (Zong, 2024). These recent advancements emphasize the importance of integrating deep learning frameworks and enhanced preprocessing methods to improve both accuracy and usability in practical, large-scale academic environments.

It is leveraged the OpenCV library for this purpose. OpenCV uses a deep-learning-based face detection approach that employs the SSD framework with a ResNet backbone, offering an advanced alternative to other OpenCV SSD models that typically utilize MobileNet as their base network. SSD performs both object localization and classification in a single forward pass through the network, making it highly efficient. This framework predicts category scores and bounding box offsets for a set of predefined default bounding boxes using convolutional filters applied to feature maps. SSD's ability to produce predictions at multiple scales and aspect ratios ensures high accuracy, even with low-resolution images as shown in Figure 2. Unlike older methods such as YOLO, or those using object proposal techniques that divide images into segments to identify possible object locations, SSD processes the entire image in one step. It eliminates the need to revisit the image or apply multiple convolutional neural networks, resulting in faster and more accurate face detection.

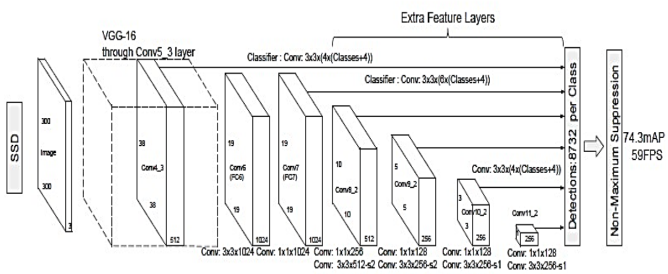

Figure 2: Single shot detection process.

In real-world scenarios, data is often collected in large volumes, making manual processing impractical. This necessitates feature extraction, where a feature represents information in an image, with facial feature extraction being crucial for face recognition. Selecting appropriate features can be complex and time-consuming. An effective feature extraction algorithm must remain consistent and stable across various conditions to ensure high accuracy. Multiple feature extraction methods exist for facial recognition. It was recommended Principal Component Analysis (PCA) for this purpose (Nithya, 2015) to implement the PCA in a face-recognition-based student attendance system. PCA is widely acknowledged for its robustness and computational efficiency. It preserves data variations while eliminating unnecessary correlations among original features and has been a prominent feature descriptor in object detection applications.

PCA functions as a dimensionality reduction algorithm that compresses facial images, represented as matrices, into single column vectors. It centralizes image data by subtracting the average value from each image. The principal components of facial image distribution, known as Eigenfaces, encode significant variations among known facial images. Each training image contributes to forming Eigenfaces, enabling them to represent unique characteristics of the training dataset. During recognition, both training and test images are projected onto the Eigenface space to create projected images, and their Euclidean distance is calculated to identify matches. Since PCA relies on all trained facial images, the extracted features reflect correlations in the training set, and recognition results heavily depend on training image quality and diversity. After face detection, the system crops the face from the image, and features are specifically extracted for further analysis. Besides PCA, the LBPH method is another widely adopted approach. In LBPH, LPB images are computed first, followed by histogram creation to generate facial recognition templates. A template comprises data representing unique and distinctive features of the detected face. Both methods demonstrate effective ways to extract features from detected faces, supporting reliable face recognition systems.

To construct our OpenCV face recognition pipeline, it is employed deep learning in two primary steps. First, face detection identifies the presence and location of faces in images without performing identification. Second, it is extracted 128-dimensional feature vectors, or "embeddings," to represent each face numerically. The neural network calculates these embeddings using a triplet loss function, ensuring that embeddings of anchor and positive images are closer while pushing negative image embeddings further apart. This approach enables the network to learn robust and discriminative embeddings suitable for face recognition. Initially, an image or video frame enters the pipeline. Face detection identifies the face's location, and optional preprocessing steps like computing facial landmarks and face alignment may be performed. Face alignment refines images by adjusting facial geometric structure through translation, rotation, and scaling, potentially improving recognition accuracy in certain pipelines. After preprocessing and cropping, the system proceeds with further processing.

To evaluate system performance, various databases are utilized, including datasets from previous researchers that incorporate variable conditions such as lighting and facial expressions. These datasets help validate and study system performance. Accuracy is calculated by dividing matched images by total tested images and multiplying by 100. The system's performance is also analyzed using custom databases. Literature reviews indicate that determining recognition accuracy is a common approach for justifying system performance. This process ensures reliable results during testing and evaluation phases.

For practical implementation, the system employs a fixed CCTV camera in classrooms to capture student images. Detected faces are stored in a database and compared with existing records using the Eigenfaces methodology. To confirm matches, a 3D face recognition technique compares new images to stored database images. Upon finding a match, the system processes the image for attendance marking. Attendance is recorded for matched student images, and information is transmitted to a central server managing the student database. The server generates attendance reports and sends SMS notifications to parents of absent students. The system software, installed on a laptop, enhances report generation and data management features, ensuring seamless operation and accountability in the attendance process.

3. METHODOLOGY

Facial recognition technology is increasingly popular in many applications due to its numerous advantages. The system requires a database to trace and track individuals and mark their attendance. The data is uploaded by associating each person's image with a unique ID and name. Before capturing images, the system prompts for the ID and name as prerequisites. More than 100 grayscale images are captured using OpenCV, serving as input for the Haar cascade. Haar Cascade converts these images into binary images and encodes them in binary form. Once the images are input, the system is trained via a "train image" option available on the screen. Each step is implemented based on its specific functionality as shown in Figure 3.

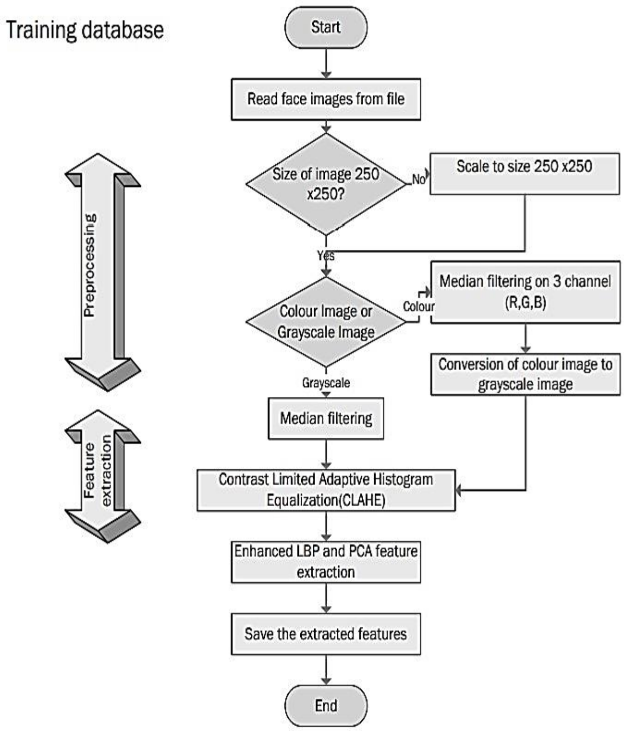

Figure 3: Flowchart of the proposed approach (training part).

This system implements a face recognition-based student attendance method. The process starts with capturing images through a user-friendly interface, followed by pre-processing the facial images. The next steps include feature extraction, subjective selection, and the classification of facial images for recognition. PCA is employed as a statistical method to summarize large data tables into a smaller set of indices, enabling easier visualization and analysis, ultimately marking attendance. Both LBP and PCA feature extraction methods are utilized and analyzed to compare stored images with input images. LBP is enhanced in this approach to minimize the effect of lighting variations, making it a widely used method. Additionally, an algorithm combining enhanced LBP and PCA is designed for subjective selection, significantly improving accuracy. The steps involved in the process, including training the database, will be elaborated in subsequent sections, with the corresponding flowchart provided as shown in Figure 4.

In our project, we have incorporated a speech system to enhance the functionality of the face recognition system. When the system detects a student's face, the integrated sound system announces the student's name aloud. This ensures that both the student and the teacher receive instant confirmation that the attendance has been successfully recorded. The speech system also acts as a notifier; when attendance begins, it signals audibly that the process has started.

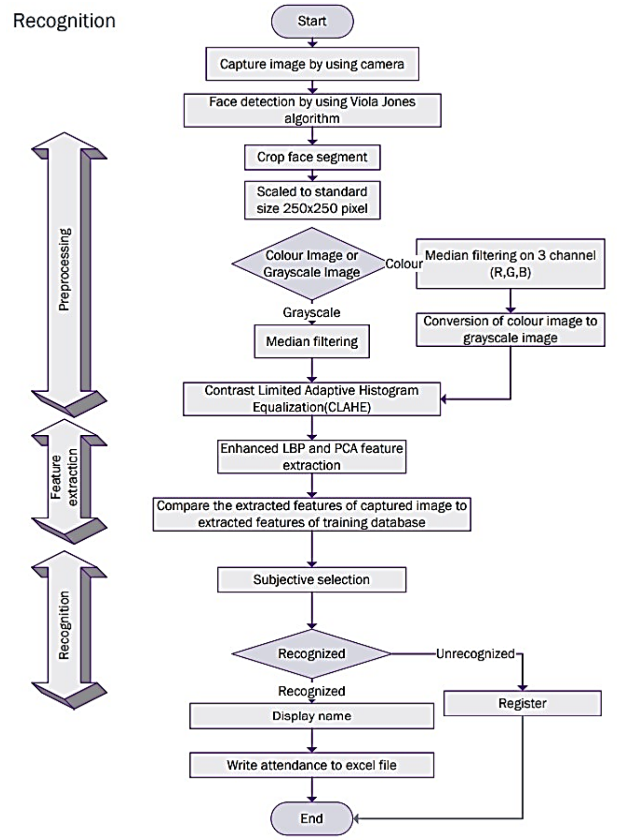

Figure 4: Flowchart of the proposed approach (recognition part).

This addition makes the system interactive and user-friendly. The sound system is fully automated and operates seamlessly through Python code. Once face recognition is initiated, the system compares the input image with the stored images in the database. Upon finding a match, it retrieves the associated name, converts it into audio, and announces it aloud. This audio announcement confirms which student's attendance has been recorded. Additionally, the system announces the exact time the attendance is taken, providing an added layer of transparency and accuracy. This innovative approach not only enhances the usability of the system but also helps in reducing errors in our project. By integrating the speech component, It is ensured real-time feedback, making the system more efficient and dependable. The audible confirmation reassures the students, teachers, and other users about the accuracy of the attendance process and eliminates any potential confusion. Overall, this addition significantly improves the functionality and practicality of the system.

As shown in Figure 5, the admin holds the highest level of privileges in the system, as they are responsible for its overall management and design. The admin's role includes registering teachers and assigning them unique IDs. They are also tasked with capturing images of students and adding these records to the database. Additionally, the admin has the authority to view and update the details of both students and teachers as needed.

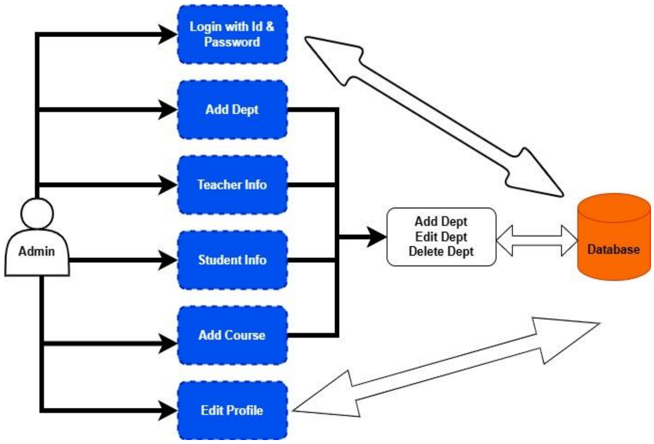

Figure 5: Use case of admin module.

Non-Functional Requirements represent the key characteristics and attributes that ensure the efficient and seamless operation of the system. These requirements include: the system must perform processes accurately and precisely to minimize errors; it should allow for easy modification and updates; any bugs or issues identified should be simple to fix; the system must ensure security and maintain the privacy of student data; it should be user-friendly and easy to understand; and the execution of operations should be quick and efficient.

In our project, the student images used for the database are captured using either a laptop's built-in camera or a mobile phone camera. Each student contributes four images—two designated for the training set and two for the testing set as shown in Figure 6 and Figure 7. The images captured with the laptop's built-in camera are considered low-quality, while those captured with a mobile phone camera are categorized as high-quality. The database includes images of 26 students in the low-quality set and 17 students in the high-quality set. By comparing the recognition rates between these two sets, we aim to analyze and evaluate the system's performance based on image quality.

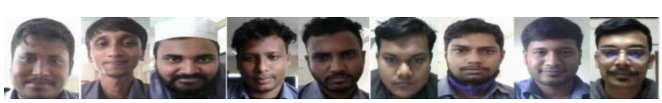

Figure 6: Sample of high-quality images.

Figure 7: Sample of low-quality images.

For the proposed approach, the input image should be frontal, upright, and contain only a single face. The system is capable of recognizing students both with and without glasses, but students are required to provide images of themselves wearing glasses as well as without them. This ensures the system is trained to recognize them accurately even without glasses. High-quality images should be captured to achieve the best results, and it's crucial that both the training and testing images are captured using the same device to avoid discrepancies in image quality. Students must complete a registration process to be included in the recognition system. Registration can be done easily using a user-friendly interface, providing a seamless experience. By adhering to these conditions, the system will be able to perform optimally. This section describes the tools and techniques employed in the project. The main focus of the project is image processing and face recognition. We utilized Laravel, Python programming, and relevant libraries to build the system. Python is a high-level, object-oriented programming language that was first created by Guido van Rossum in 1991. Initially introduced as Python 0.9.0, it was developed as the successor to the ABC programming language. Python 2.0, released in 2000, brought additional features, such as list comprehension and garbage collection. The latest major version, Python 3.0, was released in 2008. Python is widely used in a range of domains, including machine learning, software development, mathematics, data analysis, and more. It works across multiple platforms, including Windows, Linux, Mac, and Raspberry Pi. The language's simplicity and ability to support test-driven development have contributed to its massive popularity. Python allows for the creation of web applications, desktop applications, and complex scientific and numeric systems, making it versatile and widely applicable. Its concise syntax makes it easier for developers to write and understand code with fewer lines. In this project, Python is specifically used for face detection.

Laravel is another crucial component in this project. This open-source PHP framework is known for its robustness, ease of use, and adherence to the Model-View-Controller (MVC) design pattern. Laravel helps to create highly structured and pragmatic web applications by reusing components from various frameworks. It offers an extensive set of built-in features and is particularly popular for backend development. For the backend of our project, it was used Laravel, as it offers a rich set of functionalities that greatly accelerate web development. Using Laravel saves significant time compared to developing a web application from scratch, making it an ideal choice for our project's web development needs.

Composer is an essential tool that manages dependencies and libraries for projects, making it easier to set up and manage web development frameworks like Laravel. It allows users to create projects while handling dependencies efficiently. With Composer, third-party libraries can be installed seamlessly, and all dependencies are documented in a file named composer. Json, which resides in the source folder of the project. In Laravel, the Command Line Interface (CLI) is called Artisan, which provides a set of commands to streamline the web application development process. These commands, inherited from the Symfony framework, add extended features in Laravel version 9.2.0. Laravel offers a range of key features, such as modularity, testability, routing, configuration management, schema builders, template engines, authentication, Redis queues, events, and command buses, making it an optimal choice for building web applications. The framework enhances scalability, saves time by reusing components from other frameworks, and improves resource management by supporting namespaces and interfaces, which aid in the organization and efficient handling of resources.

Python provides numerous options for developing graphical user interface (GUI) applications, with PyQt5 standing out as one of the top choices. PyQt5 is a cross-platform GUI toolkit, providing Python bindings for Qt v5. It simplifies the creation of interactive desktop applications with its rich set of tools and straightforward design, making application development more efficient. A typical GUI application is composed of both the front-end and back-end. PyQt5 includes QtDesigner, a tool that enables users to create front-end interfaces quickly using a drag-and-drop method. This feature streamlines the development process, allowing developers to spend more time on the back-end functionality. It was used PyQt5 to develop the full user interface for our project. PyQt5 comes with a wide range of pre-built components, including buttons, menus, lists, tabs, and more, which can be easily integrated into applications. The included Qt Designer tool is a significant advantage, speeding up the development process and making the user interface design more straightforward and less complex as shown in Figure 8.

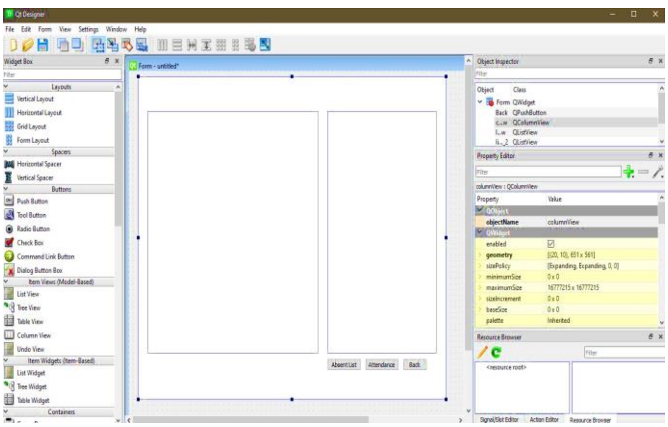

Figure 8: UI design using pyqt5

OpenCV is a versatile, cross-platform library primarily used for developing real-time computer vision applications. It was introduced in 1999 with the aim of advancing CPU-intensive applications. Initially developed in C++, OpenCV has bindings for other popular programming languages such as Java and Python, making it widely accessible across different environments. OpenCV is compatible with a variety of operating systems, including Linux, Windows, and macOS. The library's core focus revolves around image processing, video capture, and video analysis, which includes advanced features like face detection, object detection, and scene reconstruction. Its widespread applications extend to fields like Artificial Intelligence, Cloud Computing, and Edge AI, enabling computers to process and understand visual inputs at scale. With the help of OpenCV’s comprehensive algorithms, AI-driven vision applications can perform tasks such as automated inspection, handwriting recognition, object classification, and even recognizing critical situations based on visual data. OpenCV’s main strength lies in its vast library of over 2,500 algorithms, which are designed to deploy computer vision and machine learning models efficiently. Additionally, OpenCV can take full advantage of hardware acceleration and multi-core processors to deliver enhanced performance for real-time applications. As it simulates human vision through software, OpenCV intersects closely with other domains like image processing, pattern recognition, and photogrammetry. Notable advantages of OpenCV include its open-source and free nature, optimization for low RAM usage, and its broad compatibility across operating systems.

Dlib is an open-source suite that provides a wide range of machine learning tools and functions, all written in C++ and released under the permissive Boost license. Dlib offers powerful features for classification, regression, image processing, numerical algorithms, and networking. Among its many functionalities, it excels in object pose estimation, object tracking, face detection, and face recognition. Though it operates cross-platform, it is often used alongside OpenCV, particularly when the workflows involve facial capture and analysis (both recognition and detection) in Python environments. Dlib stands out for its efficient performance, accuracy, and low latency, making it a popular choice for real-time face recognition in mobile applications. In the realm of facial recognition and computer vision, Dlib has become an indispensable library due to its balanced resource usage, speed, and the quality of its results. Dlib remains an ideal resource in facial recognition pipelines, even as newer technologies emerge.

For this project, it was utilized the Laragon database, which serves as a powerful development environment suitable for various programming languages, including PHP, Node.js, Python, Java, Go, and Ruby. Laragon is praised for being fast, lightweight, easy to install, and highly portable, making it an excellent solution for building and managing modern web applications. Laragon’s architecture is designed for both stability and performance, focusing on simplicity and flexibility. The platform is optimized to be as lean as possible, providing only what is necessary to support development. Laragon operates with minimal configuration requirements, enabling users to quickly set up and start their projects without much hassle. Its control panel includes easy-to-use start and stop buttons, while the database management system is run through Heidi SQL. One of the key benefits of Laragon is its ability to simulate a real Linux development environment on a Windows machine. It enables seamless access to web directories with clean, user-friendly URLs and a sleek interface. Despite its power, Laragon is remarkably compact, with the core binary size being less than 2MB, and it consumes minimal system resources, running with less than 4MB of RAM. Unlike many development environments, Laragon does not rely on Windows services but instead uses its own lightweight service orchestration system. This allows Laragon to manage processes asynchronously and non-blocking, ensuring smooth performance without compromising speed or stability.

4. IMPLEMENTATIONS

This section outlines the implementation of the algorithm used to design the system, as well as the process of testing and evaluating the system. The application was developed using the PHP Laravel framework for the backend, and PyQT5 was utilized for the frontend of the project as shown in Figure 9. The system follows a three-layered architecture, each serving a specific function to support the entire project workflow.

The presentation Layer is responsible for presenting the user interface to the user. It contains all the visible elements of the application that users interact with, such as buttons, forms, and options for managing information. With this interface, users can easily perform tasks such as adding or deleting data from the system.

The application layer is the core of the system and manages the entire operation of the application. This layer handles all essential processes, including logging into the system, detecting and analyzing faces, recognizing individuals, and ultimately marking attendance. This layer is crucial for coordinating all the components that ensure smooth system functionality.

The data layer is dedicated to storing and retrieving all data used by the system. Key information, such as the names and images of students and teachers, is saved in the database. Once a student's face is detected and recognized, the attendance data is stored within the database for future reference. The database is essential in supporting the overall process, as any facial recognition work depends on the stored image and associated information. Before starting the recognition and attendance processes, ensuring that the face database is set up correctly is a crucial step.

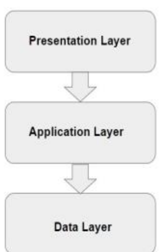

Figure 9: System design.

The face database plays a critical role as it is used as a reference point during the recognition process. For this project, we have chosen the Laragon database, which is used for storing and organizing data efficiently. To ensure better management and organization of information, the database is structured by department, with each department further divided by the level and term of the students. Additionally, courses are classified according to subjects, streamlining the allocation of specific student data based on their academic trajectory. When a new student is added to the system, their data is entered, including their name, roll number, picture, and other necessary details through an intuitive interface. Only the admin has the privilege to enter this data, ensuring security and accuracy. Once the data is entered, students are assigned to specific courses according to their level and term, and their attendance is managed accordingly. Attendance records are systematically stored in the database. These records can later be accessed and used to calculate attendance statistics, which are then automatically compiled into an Excel sheet for record-keeping and analysis. The following is an image that shows the structure of our entire database setup, providing an overview of how the data is organized as shown in Figure 10.

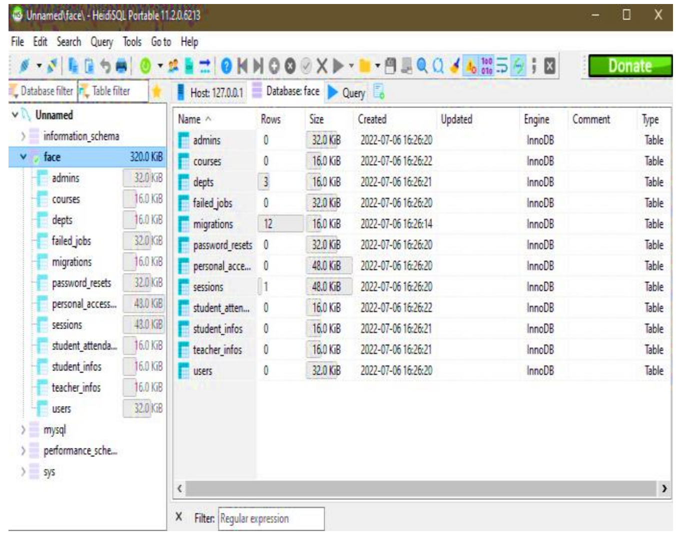

Figure 10: Face database model and its corresponding table.

The user interface is created by PyQt5. It is the GUI library of python as shown in Figure 11. It can be easily design different types of fields by using designers. Our interface comes with three steps which have to complete to enter the main system. First, the user has to select which way he wants to take attendance. Then he says in which department, which batch, which course. Ultimately the user will decide whether to take attendance, train, or look at total attendance. Only if a new student is added, it is not necessary to train once. In this way, the attendance is completed. Our user interface provides those functionalities. It provides predesigned templates. It can be also generated the code of those and it is also very easy with just one line of code.

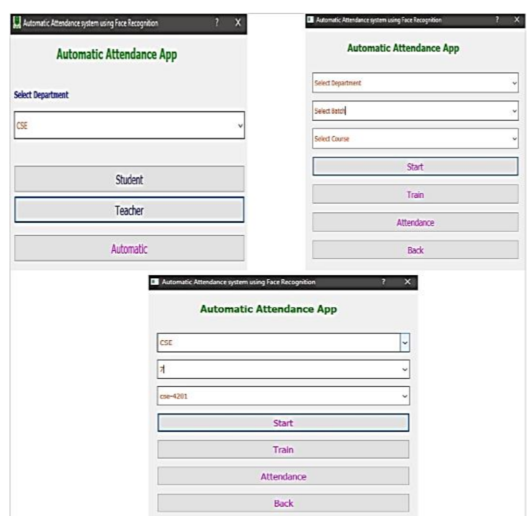

Figure 11: User interface GUI.

Figure 12: Admin interface.

Figure 12 shows what the admin sees after the admin has logged into the system. The admin can see the department, teachers, students, and courses here. Admin can also add information about the teachers, students, departments, & courses.

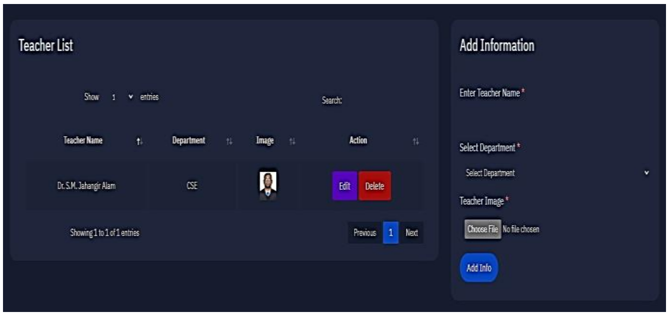

Figure 13: Teacher interface.

In the Figure 13, it can be seen the form that the admin needs to fill in to add a teacher to the system. Teacher information like their name, and department, and add a picture of them. Admin can also see all teachers' information here. Similarly, as shown in Figure 14, the admin can also add students to the system. The admin needs to provide students' information which is shown in Figure 15. Student information is filled in by the admin, first, they insert the name of the student, then they use the email, id, phone number, department, and the path of the student image. All the field is required because without those our system is not able to distinguish between students. The image path of the data set of a student is provided. The scanned image is compared with the images stored in the data set from this path. Teachers' information is also taken in our system because based on that information which can be given our teachers access to the system and also it helps to assign courses to our teachers.

During the development of the project, several issues arose that presented minor obstacles to the process. Initially, a Graphical User Interface (GUI) was created to assist users in storing their portraits for the formation of a face database. This GUI enabled easy interaction with the system. To build the interface, an external library was downloaded; however, the library had many limitations, particularly its inability to support image file types other than the .gif format. As a result, the images could not be displayed through the window. Additionally, there was a lack of sufficient face images in the initial database as only a small number of volunteers were willing to contribute their photos for testing purposes. This limitation was overcome by conducting research and finding a prepared face database that could be downloaded online. The downloaded database was normalized and converted into grayscale, which made the testing process much more convenient. Another challenge faced during development was that the website could only be accessed locally by devices. In addition, dealing with the pre-processing of captured images posed significant challenges.

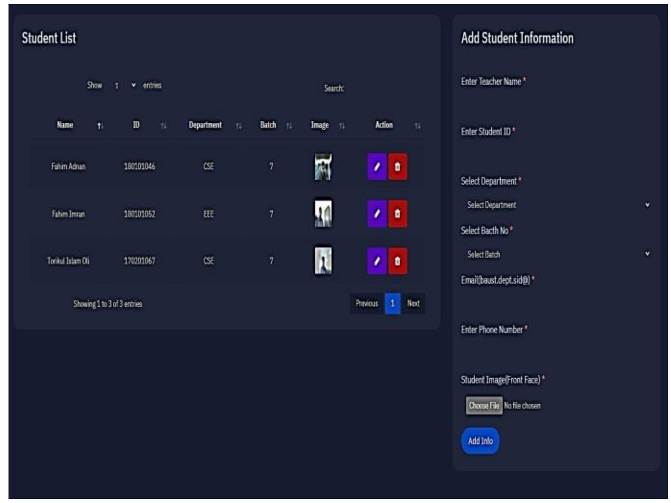

Figure 14: Student interface.

Fortunately, many of these problems were resolved by consulting online resources for recommended solutions. Overall, building a face recognition system proved to be manageable, especially with a sufficient background in the underlying processes. Most of the complex algorithms required are already provided in libraries, requiring mainly an understanding of how to integrate them into the developing system. It was made it a priority to minimize these issues and addressed them systematically.

The proposed system aims to automate attendance tracking for various organizations and address the shortcomings of traditional manual attendance methods. The system calculates attendance based on subjects and has been divided into two main parts: the frontend and the backend. The frontend side includes the GUI built with PyQt5, a Python library, which serves as the client interface. The backend side comprises the logic and functionality of the system, based on Laravel, a PHP framework, which drives the operations of the server-side logic. The system design was structured to meet the project requirements, incorporating various interfaces, components, architecture, and a well-organized database. These elements were integrated to provide the admin with an intuitive and efficient experience. By using this system, the admin can easily manage all functions, improving overall workflow efficiency. The system can be divided into multiple modules for users to interact with. These modules include image capture, face detection, and database development. The image capture module records the student's photo, which is then processed by the system. The face detection module uses algorithms to enhance the efficiency of facial recognition by pinpointing the face's location within the image. Finally, the database development module employs a biometric enrolment system to securely associate each student with their face and details, effectively storing this data in the system's database. These key modules work seamlessly together to create a fully automated and efficient attendance management system.

5. RESULTS AND ANALYSIS

The Face Recognition-based Smart Attendance System using the Internet of Things (IoT) is designed to be user-friendly and operate with high efficiency. The system is fully automated once student registration is completed and the dataset is created. This results in a significant reduction in time, improved efficiency, real-time updates, precise attendance tracking, and easy report generation in spreadsheet format. One of the key features of this approach is the user interface built using PyQt5, which is intuitive and easy to interact with. The interface includes several buttons, each performing specific functions. For example, the "train" button is used to train the system on new students by adding their facial images to the database. This training only needs to be done once for each new student. The "start" button initializes the camera and triggers the automatic face recognition process based on the faces detected by the system. Additionally, the system allows users to browse and select facial images from a specific database to test the recognition functionality as it can be seen from Figure 15.

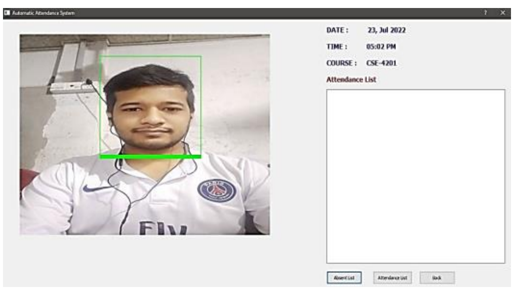

Figure 15: User's interface after start detecting.

The system follows a series of image processing steps to complete face recognition. Initially, it performs face detection using various techniques such as feature extraction, image analysis, and face detection algorithms. After detecting a face, the system compares the detected image to the stored image data in the database for recognition. The backend of the system is developed using Laravel, a PHP framework, while the database management is handled using Laragon. This structure allows the system to process images efficiently while maintaining data integrity and performance. To enhance the accuracy and reliability of face recognition, the system utilizes an enhanced LBP algorithm with a radius of two. This particular radius size has been selected based on extensive testing and evaluation. The choice of radius size and its effects on the system's performance are discussed further in the analysis section. In this chapter, we will present the results of the testing and showcase the final outcomes of the project, along with images to better illustrate the results and their implications.

Figure 16 shows once the start button is pressed, the process is automated. The face image is captured from the video recording frame and face recognition is performed. It also shows whether attendance is taken or not. The excel sheet of the attendance taken by our system. In Figure 17, it was calculated the attendance for the entire semester of our varsity and given the percentage and marks obtained accordingly. In this case, it is used the whole number 15 to calculate the score.

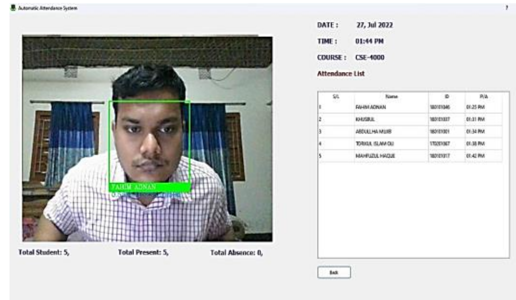

Figure 16: Real-time face recognition (automated).

	A	B	C	D	E	F	G	H	I
1									
2			Course No: CSE - 4000		Date: 29 May 2022				
3									
4		Total Student: 5			Total Present: 1		Total Absence: 4		
5									
6		ID	Name	P/A					
7		180101046	Fahim Adnan	4:14 PM					
8		180101037	Khusbul	Absent					
9		180101001	Mujib	Absent					
10		180101017	Mahfuzul	Absent					
11		170201067	Torikul	Absent					
12									
13									

Figure 17: Attendance in the spreadsheet for a single day.

This proposed approach introduces an effective method for performing face recognition in a student attendance system, leveraging the texture-based features found in facial images. Face recognition, in this context, refers to the process of identifying an individual by comparing their real-time captured image to previously stored images in the database. To ensure the system's effectiveness, the training set must be updated to reflect the most current images, taking into account the individual's appearance along with other critical factors such as lighting conditions.

The system applies the Single Shot Detection (SSD) object detection framework to detect faces in real-time. Once a face is detected, an algorithm is employed to extract the key features needed for face recognition. Before proceeding with feature extraction, several pre-processing steps are performed on the captured facial image to enhance recognition accuracy. Median filtering is utilized as a pre-processing technique because it effectively preserves image edges while eliminating noise, which could otherwise decrease system accuracy. To maintain consistency and facilitate recognition, the facial image is scaled to a standardized size. If the image is not in grayscale, it is converted into a grayscale format since the LBP operator, which is used for feature extraction, requires grayscale images. One of the common challenges in face recognition systems is inconsistent lighting, which can significantly affect accuracy. To address this, various strategies have been employed within this approach to mitigate the impact of non-uniform lighting. Before performing feature extraction, the cropped face image (Region of Interest or ROI) undergoes pre-processing to correct for lighting inconsistencies and improve the robustness of the recognition process.

	A	B	C	D	E	F	G	H
1								
2								
3		Dept: CSE	Batch No: 7	CSE-4000	tal Students: 5			
4								
5		Name	Id	Total	Percent(%)	Number		
6								
7		Khusbul	180101037	2	20%	3		
8		Adnan	180101046	10	100%	15		
9		Mahfuzul	180101017	0	0%	0		
10		Torikul	170201067	1	10%	2		
11		Abdullah	180101001	0	0%	0		
12								
13								
14								

Figure 18: Attendance in the spreadsheet for a whole month.

The proposed algorithm is designed to work with a single face in the image at a time. If multiple faces appear within the same image, each face will occupy a small region, leading to the extraction of inaccurate features from these regions, thereby decreasing the system's performance. As a result, if more than one face is detected, the system is unable to perform recognition as it can be seen from Figure 18. Additionally, the LBP algorithm used for feature extraction is highly sensitive to image quality and can be significantly impacted by blurred images. The LBP algorithm is a texture-based descriptor that extracts local grayscale features from small regions within the image. To ensure high accuracy, the training image and the test image must be captured with the same device and under similar conditions. Using a higher-quality camera improves results, but the built-in webcam on laptops, which is the default device used in this approach, often results in lower image quality due to its limited performance, producing darker and blurrier images.

To address this, the system works best when both the test image and the training image are captured in the same location under nearly identical lighting conditions. Additionally, a properly angled photo is crucial for the best performance. For simplicity, this system uses Eigenfaces for face recognition based on grayscale images. Converting color images to grayscale is an essential step, as shown in the paper. However, to further improve the algorithm's performance, the next section proposes solutions to address these challenges and increase system accuracy. In terms of image quality, several factors affect the performance and accuracy of the system. One of the most crucial steps is the application of various pre-processing techniques to standardize the images fed into the face recognition system. Face recognition algorithms are typically very sensitive to lighting conditions. For instance, if the model was trained with images captured in a dimly lit room, it may fail to recognize the same person in bright lighting conditions, highlighting the issue of luminance dependence. Moreover, other factors such as face positioning, size, rotation angle, hair and makeup, emotion, and lighting placement must be consistent within the images. Thus, image pre-processing filters are essential to handle such variations effectively.

To enhance performance, it is recommended to remove unnecessary pixels around the face such as the hair and background using an elliptical mask. This isolates the face region and reduces potential sources of variation. Furthermore, techniques such as color-based face recognition or advanced processing stages like edge enhancement, contour detection, or motion detection could improve results. While resizing images to a standard size is often necessary, it may distort the aspect ratio of the face. To avoid this issue, the system can be updated to resize images while preserving their aspect ratio. An automatic attendance system that uses face recognition is a highly effective and efficient solution for tracking attendance. It saves time, provides real-time recognition, generates automatic reports in spreadsheet format, and allows easy online updates. The system has been implemented using a Webcam, OpenCV, Haar Cascade, and Python. Haar Cascade, a prominent face detection algorithm, ensures the reliability and performance of the system. The Face Recognition-based Smart Attendance System using IoT is simple, efficient, and effective for real-world applications.

6. CONCLUSIONS AND FUTURE WORKS

Before the development of this project, traditional attendance-taking methods had many shortcomings, leading to significant challenges for institutions. The integration of facial recognition technology into the attendance monitoring system addresses these flaws by ensuring accurate attendance tracking while reducing human intervention. This system automates the complex tasks, saving valuable resources and streamlining the process. The only requirement for this solution is sufficient storage space for the face database, which can be handled by micro-SD cards. In this project, a face database has been successfully created to store student details like name, ID, department, and pictures. The face recognition system also operates effectively, and a user-friendly webpage with full functionality has been built. Although the database creation is hidden from users, they can interact with it via the web interface. Teachers can log into the system to take attendance, with the camera detecting student faces and recording their attendance in the database. The system's interface is designed for ease of use, allowing teachers and admins to view attendance reports, including total presence, absences, and percentage of attendance. The generated spreadsheet enables tracking of student attendance and related performance. Ultimately, this automated system improves the accuracy, speed, and convenience of attendance-taking while reducing errors and human effort, making the process more efficient. It helps meet the demand for automatic classroom evaluation, ensures students' attendance, and shows the promising potential of the system to enhance administrative functions.

In the future, this system could be implemented as part of the 'smart classroom' concept. However, its efficiency can be further improved through better hardware integration. For hardware enhancements, using high-speed storage drivers and increased RAM would result in faster training times and quicker model responses. Camera quality also plays a critical role; utilizing high-quality cameras with 720p or 4K resolution could provide the system with clearer video feeds, enabling the model to extract more accurate features. Enhancing these hardware components is essential for improving the overall accuracy and performance of the system, requiring us to also update the software to align with these enhancements. Software upgrades remain one of the most crucial improvements. Techniques such as separating person and pose information using unsupervised clustering could enhance accuracy. A common method for improving the model's performance is expanding the dataset, such as creating variations of frontal-face images to create better decision boundaries. Furthermore, the system can be made more flexible by allowing updates to the facial templates if a student undergoes significant facial changes. Another improvement would be refining the recognition algorithm to handle rotated faces more efficiently. Parameters for refining the algorithm include Diagnostics, Weight Initialization, Learning Rate, and more. Improving these aspects can contribute to developing a more robust recognition system. It was also aim to install the camera in the center of the classroom, allowing the system to capture everyone's attendance at once, though this would require investment. With this improvement, the system's utility will grow significantly. Future work could involve alerting students about their attendance via SMS. A GSM module could be integrated to send SMS alerts to the students' parents, providing further enhancements to this system.

This proposed approach comes with certain limitations. First, it requires the input image to be a clear, upright, and frontal image of a single face. The accuracy of the system may decrease under poor lighting conditions, and false recognitions can occur if the captured image is blurry. Since LBP is a texture-based feature extractor, both the test and training images must be of similar quality and captured with the same device to ensure optimal results. Furthermore, makeup worn by individuals in images may obstruct key facial features, affecting the recognition accuracy. Upgrading the camera and improving lighting can mitigate some of these issues, enhancing the system's stability and reducing dependency on external lighting conditions. In the current approach, a built-in laptop camera is used, but its poor lighting source limits the effectiveness of the system. For better results in the future, using a higher-quality camera and a stronger light source would be beneficial. This would help reduce the impact of environmental lighting on both test and training images. Moreover, incorporating a multi-face recognition system, rather than limiting it to a single face, could enhance the system's efficiency. The test and training images in this approach are highly sensitive to the capture device, so using the same device for both would yield better performance. To improve the system further, other algorithms could replace LBP. For instance, Artificial Intelligence (AI) algorithms, like Convolutional Neural Networks (CNN), are gaining popularity in face recognition tasks. CNN can perform effectively even with variations in the training images, provided the system has a sufficiently large database. However, CNN requires substantial data to achieve higher accuracy, though it performs exceptionally well with smaller, well-curated datasets. Additionally, in the preprocessing phase, techniques like affine transformation can be applied to better align facial images based on key coordinates, such as the eyes' position.

REFERENCES

Abdelfatah A. T., Mohammad A. A., & Omaima N. A. AA. (2015). Real-Time Group Face-Detection For An Intelligent Class-Attendance System. I.J. *Information Technology And Computer Science.* May 2015. Vol. 04 Issue. 01. Page 66-73.

Adediran, A. O., Hammed, F. O., & Musa, B. B. (2023). Smart Attendance System using Face Recognition for Tertiary Institutions in Nigeria. *International Journal of Engineering Research & Technology (IJERT)*, 12(07), July 2023. ISSN: 2278-0181.

Ajinkya P., & Mrudang S. (2014). Implementation Of Classroom Attendance System Based On Face Recognition In Class. *International Journal Of Advances In Engineering & Technology.* July 2014. Vol. 7, Issue 3. Pages 974-979.

Akash S., Abhishek G., & Shreya B. (2021). Automated Attendance System With Face Recognition. *International Journal of Engineering Applied Sciences And Technology*. February 2021. Vol. 5, Issue 12. Pages 233-241.

Akshara J., Akshay J. T. L., & Krishna Y. (2017). Automated Attendance System Using Face Recognition. *International Research Journal Of Engineering And Technology (Irjet).* Jan - 2017. Vol. 04 Issue. 01. Pages 1467-1471.

Antony J. A., Chitrakala P., & Senthamil S.K., (2014). Face Recognition-Based Attendance Marking System. *International Journal Of Computer Science And Mobile Computing.* February- 2014Vol.3 Issue.2. Pages. 337-342.

Anagha V., Akash S., Sayali G., & Neha T. (2021). Student Attendance System Using Face Recognition. *Irish Interdisciplinary Journal Of Science & Research (Iijsr)* April-June 2021 Vol.5, Iss.2. Pages 84-91

Anitha G., Priyanka D., Sunitha D.P., & Vidhya S.J. (2020). Face Recognition Based Attendance System Using Mtcnn and Facenet. *International Research Journal of Engineering And Technology (Irjet).*January 2020. Volume 6, Issue 8. Pages No:189-195

Ankur S. B., Chandra L., Sandy K., Untung R., & Widya F. (2020). Design Of Face Recognition Attendx For Recording Student Attendance Data Based On Artificial Intelligence Technology. *International Journal Of Engineering Applied Sciences And Technology.* May 2020. Vol. 5, Issue 1. Pages 485-487.

Ankita K., Adib R., Pratik S., & Labeeq Q. (2016). Customized Nfc Enabled Id Card For Attendance And Transaction Using Face Recognition. *International Research Journal Of Engineering And Technology (Irjet).* Sep-2016. Volume: 03 Issue: 09. Pages 1366-1368.

Ansa G. A., & Essien, E. I. (2023). Deep Learning-Based Face Recognition Attendance System. *Global Journal of Engineering and Technology Advances*, 14(01), 015–025. DOI: 10.30574/gjeta.2023.14.1.0152

Arun K., Amar P. Z., Chetan J. B., Nikhil D. B., & Sudesh V. K., (2017). Attendance System Using Face Recognition And Class Monitoring System. *International Journal On Recent And Innovation Trends In Computing And Communication.* February 2017.. Vol. 5 Issue. 2. Pages 273 – 276.

Ashim S., Dwijen R. P., Nirmalya K., & Mrinal K. D. (2012). Study Of Implementing Automated Attendance System Using Face Recognition Technique. *International Journal Of Computer And Communication Engineering.* July 2012. Vol. 1, Issue. 2. Pages 100-103.

Bindu S.M., Dona J., Martin M., Thomas M., & Vineetha V. (2020). Automatic Attendance System Using Face Recognition. *International Journal For Research In Applied Science & Engineering Technology (IJRASET).* July 2020, Volume 8 Issue 7. Pages 769-773

Chowdhury S. M., Mahmuda K., Partha C., Sk. F. I., & Saifur R. (2020). Automatic Student Attendance System Using Face Recognition. *International Journal Of Engineering And Advanced Technology (Ijeat).* February 2020. Volume-9 Issue-3. Pages 93-99.

Debadrita G. (2021). Real-Time Attendance System Using Face Recognition Technique. *International Journal Of Engineering Applied Sciences And Technology*. 2021. Vol. 5, Issue 9, Pages 258-261.

Eueung M., Nopendri, Rahmat M., & Syifaul F. (2020). Attendance System Using Machine Learning-Based Face Detection For Meeting Room Application. *International Journal Of Advanced Computer Science And Applications*. 2020, Vol. 11, Issue. 8. Pages 286-293.

Feng, J. & Zong, C. (2024). Research on Intelligent Attendance Management System Based on Face Recognition. *Proceedings of SPIE – The International Society for Optical Engineering*, Vol. 13071, 2024. DOI: 10.1117/12.3025538

Himanshu T., Ratish A., Mahesh P., & Sachin G. (2018). Live Attendance System Via Face Recognition. *International Journal For Research In Applied Science & Engineering Technology (Ijraset)*. April 2018. Volume 6 Issue 4. Pages: 3891-3898

Jain M. D., Karhe R. R., & Patil C. S., (2014). Student Attendance System And Authentication Using Face Recognition. *International Journal Of Engineering Research & Technology (Ijert)*. July – 2014. Vol. 3 Issue 7. Pages 373-375.

Kama A. O., Muhammad A. M. N., Roslina M., Saiful I. S., Shahrani S., & Yuslinda W. M. Y. (2018). Realtime Internet Based Attendance Using Face Recognition System. *International Journal Of Engineering & Technology*. January 2018. Vol. 7, Issue. 3. Pages 174-178.

Kowsalya P., Pavithra J., Sowmiya G., & Shankar C.K. (2019). Attendance Monitoring System Using Face Detection &Face Recognition. *International Research Journal of Engineering And Technology (IRJET)*. Mar 2019.Volume: 06 Issue: 03. Pages 6629-6632.

Kusuma G. P. A., & Warman M. H. (2023). Smart Attendance System using Face Recognition based on RetinaFace and FaceNet. Communications in Mathematical Biology and Neuroscience, 2023 (2023), Article ID: 122. DOI: 10.28919/cmbn/7872

Mayur S., Minakshi V., Priya J., & Sujata J. (2020). Automatic Attendance System Using Face Recognition Technique. *International Journal Of Recent Technology And Engineering (Ijrte)*. May 2020. Volume-9 Issue-1. Pages 2134-2138.

Mekala V, Manimegalai M., Nandhini K., & Vibin M. V., (2019). Face Recognition Based Attendance System. *International Journal Of Innovative Technology And Exploring Engineering (Ijitee)*. October 2019. Volume-8 Issue-12. Pages 520-525.

Nithya D. (2015). Automated Class Attendance System Based On Face Recognition Using Pca Algorithm. *International Journal Of Engineering Research & Technology (Ijert)*. December-2015. Vol. 4 Issue 12. Pages 551-554.

Prateek K., Sakshi P., Shelesh G., & Ravi K. (2018). Face Recognition Based Smart Attendance System Using Iot. *International Journal Of Computer Sciences And Engineering*. May 2018. Vol.-6, Issue-5. Pages 871-877.

Shubham K., & Sagar K. (2021). Review Paper On Contactless Attendance System Based On Face Recognition. *International Research Journal Of Engineering And Technology (Irjet)*. Jan 2021. Volume: 08 Issue: 01. Page 1678-1682.