MFLCNN Based Ai Attendance Wizard: The Next Generation App for Real-Time Attendance Evaluation Using Deep Learning

T.S.Urmila[1]*, N.Gnanasankaran[2], C.Jayapratha[3], G.B.Govindaprabhu[4]

[1]Assistant Professor, Department of Computer Science, Thiagarajar College, Madurai, Tamilnadu, India, Email: tsurmila4@gmail.com
[2]Assistant Professor, Department of Computer Science, Thiagarajar College, Madurai, Tamilnadu, India, Email: sankarn.iisc@gmail.com
[3] Professor and Head of the Department, Department of Computer Science, Karpaga Vinayaga College of Engineering and Technology, Madhuranthagam, Tamilnadu, India, Email: jayaprathaclement@gmail.com
[4] Research Scholar, MKU23PFOS10909, Madurai Kamaraj University (MKU), Madurai, Tamilnadu, India, Email: prabhupri.pp@gmail.com
*Corresponding Author

Received: 17.07.2024 Revised: 13.08.2024 Accepted: 10.09.2024

ABSTRACT
A growing demand for efficient attendance evaluation systems is driving innovation. This research uses an interactive app to evaluate attendance in real-time attendance for college or office. Monitoring and managing attendance is made easier with automated face identification and analysis. In the interactive app, users can continuously capture, process, and evaluate attendance records. It is used here to compare the proposed attendance evaluation system to traditional methods. It is possible to solve the Traditional Method's problems with an authentic and smart attendance system. In attendance systems, biometrics such as face recognition and fingerprint recognition are becoming more prevalent. A face's unique features make it recognizable. Using an interactive app, this work evaluates and predicts the attendance of students in real-time. In the beginning, all departmental, academic, staff, and student information is uploaded to the server along with a student's unique ID. Using the staff's credentials, the Interactive App will capture real-time multi-face images for transmission to the server to generate an AI model (MFLCNN Classifier). A notification marks attendance by matching images with students in the database. It is demonstrated in this study that multi-face attendance evaluation in real time can improve accuracy, convenience, and reliability. It also reduces the time of the traditional attendance process.

Keywords: App Based Attendance System, Multi face detection, AI Modeling, Django, MFLCNN, FFT, Soft Thresholding, DRCGT.

1. INTRODUCTION

A lack of efficiency and limitations plague traditional attendance systems, whether manual or semi-automated. It is time-consuming and susceptible to errors to take attendance manually in classrooms, meetings, or organizational settings. The use of semi-automated biometrics such as RFID, fingerprint recognition, and Footprint-based biometrics have attempted to address these issues, but have often been associated with high implementation costs, maintenance challenges, and privacy concerns.It is a challenge for every education system to collect and manage student attendance. Moreover, marking attendance takes up much lecture time and energy from the teacher. The manual attendance process can be time-consuming and error-prone. These issues can be solved by Automatic Attendance Systems. Among other automation technologies (such as RFID), biometric attendance stands out (11). To identify a person, biometrics take into account their unique biological and physical traits. Most commonly used biometrics are palm prints, fingerprints, faces, and iris scans.

The fast-paced and technology-driven environment today demands a robust, cost-effective, and efficient attendance evaluation system that can seamlessly integrate with educational institutions, corporate offices, and other organizational settings (16). A good attendance marking system should not only streamline the process but also make it accurate, reliable, and resistant to manipulation. With recent advancements in computer vision and facial recognition technologies, we can address traditional attendance systems' limitations. It is now possible to develop an automated, real-time attendance evaluation system without having to rely on manual intervention or specialized hardware. There are no two faces that are exactly the same, so facial recognition is usually a successive model. An automatic attendance system (AAS) estimates a student's presence or absence in a classroom using face recognition technology. With a high-definition monitor video streaming service, the machine can detect the presence of all students in the classroom.

This work proposes a novel method for evaluating attendance via a mobile application. The AI model identifies and marks individual attendance based on a comprehensive database of facial images. With the proposed solution, existing methods are addressed by offering a user-friendly, cost-effective, and highly accurate system that can be easily integrated into various settings. Using face recognition technology, the proposed work will automatically estimate whether a student is present in the classroom. Using this system, students' images are captured on both a website and a mobile app.

1. In this framework, an intelligent attendance system is provisioned through a multi-face recognition system that predicts the individual faces of students.
2. A key objective of the work is to construct attendance that would generate timely reports and provide an intelligent attendance system in place of other attendance systems.
3. This work stores information such as department, staff, students, and day-order details, and uploads student's facial images with unique IDs to the server.
4. A real-time image will be captured and matched with the pre-uploaded face data during attendance.

In the rest of the paper, the information is organized as follows. There is an outline of related work in Section The working mechanism of the proposed design is explained in Section 3. In Section 4, a performance analysis is presented. The key triumphs are discussed as well as the future errands in Section 5.

2. RELATED WORKS

In (1), Matilda et al. developed an attendance observation system, which was a way to distinguish students from advanced images or videos. Identifiers were created for each individual image of a group of students. After recognizing the faces and matching them with the training data, the system trained. Students who are missing from a digital video or frame indicate that they are absent from the classroom, and the admin is notified of their absence. Robotized executive involvement was presented by Menakadevi et al. (2). By recognizing students via web cam, this framework imprints their engagement by remembering them after they enter the classroom via web cam.

A novel solution to attendance tracking difficulties, disengagement, and fake attendance during virtual learning was presented by Anzar SM et al. (3). Based on the Dlib open-source library, this system used facial recognition. Using a revolving camera, Gupta SK et al. (4) present an unobtrusive facial recognition-based smart classroom attendance management system. The system employs Max-Margin Face Detection (MMFD) for face detection, and Inception-V3 CNNs for student identification. In the experiments, train and test accuracy were 97.67% and 96.66%. By utilizing biometrics, P. Roy et.al (5) developed a real-time monitoring system of student attendance in educational institutions. The system has three stages: enrollment, attendance, and data storage. Finger enrollment and attendance are reported using the fingerprint module.

According to Helmi et.al (6), the proposed face recognition system can handle grooming, color differences, and minor medical conditions. According to the results, the proposed system can improve attendance processes. In their proposed system, Souza et al. (7) hope to automate student attendance by using facial recognition technology. This work automates, simplifies and simplifies attendance marking and management. This work uses image processing to recognize faces. The process of comparing the processed image with the existing record marks attendance in the database. Siva Narayana et.al (8) describe traditional methods of tracking attendance as difficult. A significant challenge facing face recognition is the need for efficient and automated methods for registering attendance. Currently, fingerprints and RFID tags are routinely used to automatically mark attendance.

Diwakar Dhillon et.al (9) developed a facial recognition system for automating attendance. The system can be used in a variety of fields. By using face identification, the application eliminates proxy attendance. By examining test and training photographs, this method determines which students are present and which aren't. An Automatic Attendance System in a classroom is demonstrated by Aziza Ahmedi et al. (10). A video clip of the classroom is captured and stored in the database, and these videos are converted to frames. In addition to face detection techniques, such as Ada-boost, local binary pattern algorithms (LBP) and Histograms of Oriented Gradients (HOG) are used for feature extraction.

Although several attendance management systems have been proposed and implemented over the years, there are several limitations that prevent widespread adoption. It is time-consuming, especially in large classrooms or organizations, prone to human errors such as misidentification or proxy attendance, and lacks real-time tracking capabilities. Despite improvements in lighting and angles, traditional face recognition systems lack accuracy due to lighting variations, face mask occlusions, and privacy concerns. The time-consuming process of taking and storing individual student images leads to large datasets for training and prediction in attendance systems. Due to limited training data, facial recognition may fail, and the attendance process itself can be slow since each student must present themselves. Furthermore, classroom cameras and centralized recording can be expensive and tedious. Through a critical analysis of these limitations, this paper highlights the need for a more robust, accurate, and efficient attendance evaluation system, pointing out the potential advantages of the proposed solution and supporting the research.

3.PROPOSED WORK

The proposed model is intended for Web and Mobile it is developed using android and Django. As shown in Figure 1, the staff open an app with their credentials and upload images of students. The image is sent to the image tracker by the server. The server already contains a variety of trained ML models. In this model, the faces in the image are detected by using facial recognition algorithms from the group of students images. A student's face is linked with his/her register number and updated in the database and Result manager handled this process. To confirm attendance, the server sent updated student information to the app. The process involves three major steps, beginning with finding an appropriate database that contains multiple images of each face. Based on the database images, a face recognizer is trained and tested on the faces it has been trained to recognize.

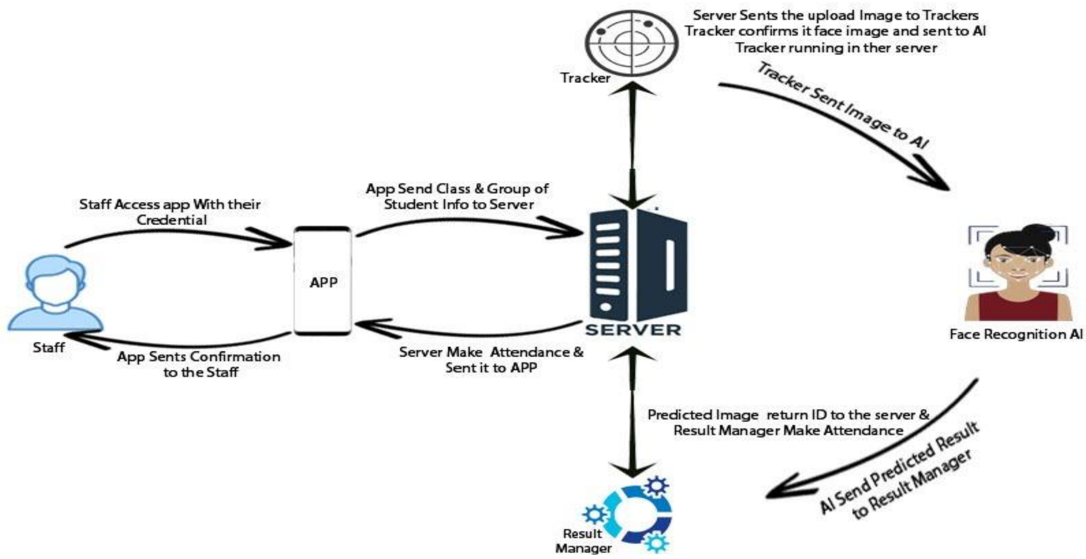

Figure 1: Proposed Framework Architecture

The proposed work dataset contains real-time data from various department students in Thiagarajar College, Madurai. The learners are photographed five times (left facing, right facing, upward looking, downward looking, straight facing). Each image is stored in a folder named on the student register number. Using a web cam or mobile camera, students are captured and a single student will be photographed from a variety of angles and gestures and the images are pre-processed. A region is created by cropping the images, which will then be used for recognition. Afterwards, resize and position the cropped images. Once uploaded, the images will be saved as the names of the students. Table 1 shows the department those who contribute in this work.

Table 1: Dataset Sample

Department	Code	Students	Images (4 or 5 Images / Student)
Tamil	D1	41	205
CS	D2	35	175
CA	D3	39	195
English	D4	38	190
BBA	D5	28	140
DS	D6	19	95

3.1 Process of the work

From user registration to attendance marking, monitoring, and system maintenance, these detailed steps cover the entire attendance system workflow. Utilizing face recognition technology, different user groups (admin, department, students, and tutors/staff) ensure collaboration and efficiency.

i. **Admin Setup**: In the system, the admin creates and manages departments. It is the administrator who assigns tutors or staff members to specific departments or courses. It is the responsibility of the admin to manage the user accounts and oversee the overall administration of the system.

ii. **Student Registration**: Students are identified by a unique Register Number and Date of Birth (RD), which serve as login credentials for each student, which are used for authentication. By using their credentials, students are able to log into the system (website or mobile app) and access their information. Their face images are uploaded in five poses: left, right, up, down, and straight. Each uploaded face image is associated with the student's ID in the system's database.

iii. **Training Set Generation**: In the department layer, courses, graduations (UG, PG, M. Phil, Ph.D.), and student details are managed. For each course or class, the department layer generates a training set using the face images uploaded by students. With the training set, the system trains a facial recognition model using a machine learning algorithm.

iv. **Attendance Marking**: Before class begins, a notification is sent to the tutor/staff that the training set is ready. A tutor or staff member logs in to the mobile app. With the mobile app, tutors and staff capture a single group image of students in the classroom. Through the mobile app, the group image is uploaded to the server. Upon receiving the group image, the server forwards it to the appropriate machine learning model. The model separates individual faces from the group image using face detection algorithms. By comparing the detected faces with the training set, the facial recognition model detects the individual faces. The system marks the attendance of the corresponding student if a match is found. In the database, attendance records are updated. Students and tutors can be notified when attendance is marked.

v. **Attendance Monitoring and Reporting**: Using the website or mobile app, students can view attendance records. Course/class attendance reports can be accessed by tutors/staff. Admins can monitor the performance of the system and access attendance reports.

vi. **System Maintenance and Updates:** The admin can perform system maintenance tasks like user management, database backups, and software updates. As needed, department layers can update course information, student lists, and tutor assignments. It is possible for students to update their faces or personal information. Any issues with attendance marking can be reported by tutors/staff.

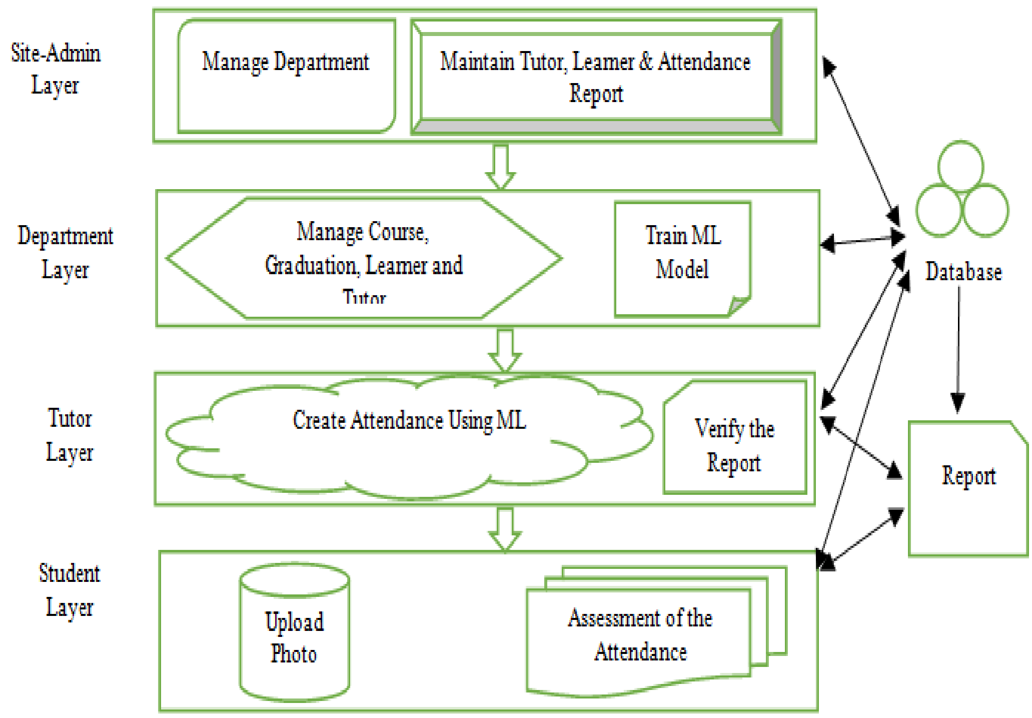

Figure 2: Process Flow of the Work

3.2 Site- Admin Layer

Figure 2 shows the process of the proposed work. As the top-level authority, Site-Admin is in charge of overall system administration. Through this layer, multiple departments, tutors, and learners can be managed efficiently, facilitating centralized control and coordination. Within the system, tutors and learners are managed by the site-admin. By assigning specific privileges and access levels, site administrators can assign tutors or instructors specific tasks. As a result, tutors are able to monitor attendance, record attendance data, and evaluate learner participation. Tutors can be assigned to multiple courses, their access permissions can be updated, and seamless communication and coordination between tutors and learners can be ensured. The site administrator can manage learners in the attendance evaluation system. Manage user accounts, ensure data integrity, and create learner profiles. Learner requests may be reviewed and authorized by the site administrator, and discrepancies and exceptions may be handled by them.

3.3 Department Layer

Within the department, the department layer acts as a point of communication and coordination between the admin, tutors, and students. The system distributes notifications, announcements, and updates related to attendance, courses, and other departmental activities. Grievances or issues about the attendance system or related processes can also be handled by the department layer. In every department, there are many courses. It is difficult to keep track of all the student information in each course and their graduation. It manages courses, graduations, learners, and tutors.

a) Graduation Management:

It deals with graduations at different levels, including undergraduate, postgraduate, M. Phil, Ph.D., etc. Using this information, students are categorized and grouped based on their graduations, which may differ in course requirements and schedules. Student data can be organized and managed efficiently within the department using graduation details.

$$\mathrm{Graduation(G)} = \{\mathrm{UG, PG, M.Phil, Ph.D} \dots \mathrm{N}\}$$

b) Course Management:

Within the department, the department layer maintains and updates the list of courses. It includes adding new courses, updating course details (e.g., course codes, titles, credits), and archiving or removing obsolete courses. In order to assign courses and classes to students and tutors/staff, course information is crucial.

$$\mathrm{Course(C)} = \{\mathrm{B.Sc(CS), B.A(Tamil)}, \dots \mathrm{N}\}$$

c)Tutor/Staff Management:

Within the department, the department layer manages tutors and staff assigned to various courses. It includes adding new tutors/staff, updating their information (e.g., contact details, assigned courses), and removing tutors/staff as needed. Accessing the attendance system and assigning tutors to appropriate courses and classes relies heavily on tutor/staff information.

$$\mathrm{Tutor(T)} = \{\mathrm{T_1, T_2} \dots \mathrm{T_N}\}$$

d)Student Management:

Information about students is maintained by the department layer, including their registration numbers, names, and contact information. Existing student records can be updated and removed as needed, and new students can be added. Each student is associated with their respective courses and graduations, ensuring accurate attendance tracking.

$$\mathrm{Student\ (S)} = \{\mathrm{S_1, S_2} \dots \mathrm{S_N}\}$$

e) Training Set Generation:

Using the face images uploaded by students enrolled in the course, the department layer generates a training set for each course. In the training set, the Deep Learning model is trained for facial recognition and attendance marking. Whenever student enrollment or face image updates occur, the department layer ensures that the training set is updated. Preprocessing is done before training images, including Image ROI Cropping,

i) Image ROI Cropping using DRCGT

A cropping algorithm uses an adaptive ROI to determine the most informative region of an image. The Gabor texture analysis in adaptive ROI cropping simulates human perception of different orientations and frequencies by using Gabor filters. By using this technique for region cropping, facial features are more conspicuous and visible. The following algorithm 1 shows how the DRCGT works with the input images. The method generates texture-enhancing response maps by convolution input images with Gabor filters. An orientation-based feature map highlights dominant textures based on mean and maximum responses. The threshold is used for detecting suspicious regions based on a specific threshold. Using morphological operations like closing, padding, and obtaining the bounding box of the largest connected component, an ROI is created. Algorithm 1 describes the processing of DRCGT.

Algorithm 1: Dynamic Region Cropping with Gabor Textures

Algorithm: Dynamic Region Cropping with Gabor Textures (DRCGT)
Input: I_{In} - Input image
Output: I_{ROI} - Cropped ROI image

1. Read the input image I_{In}
2. Initialize K and λ
3. Gk(x,y), k=1 to K //
4. Gk(x,y) = I_{In}⊗GaborFilterk, k=1 to K // Gabor filters to get response maps
5. μk(x,y) = mean(Gk(x,y)), Mk(x,y) = max(Gk(x,y)), k=1 to K // compute mean and max
6. F(x,y) = (μ1,...,μK,M1,...,MK) // Concatenate μ and M to form feature map
7. H(x,y) = (Mk(x,y) > τ*max(Mk)) // threshold max
8. R(x,y) = close(H(x,y)) // Morphologically closed image
9. B = bounding_Box(large_Connected_Component(R))
10. B = pad(B, P)
11. I_{ROI} = crop(I_{In}, B)
12. Return I_{ROI}

End Algorithm

The ROI provides the opportunity to analyze or diagnose a suspected region based on an image cropped based on this ROI. For an effective visual comparison, you can compare the original image and the cropped ROI side by side.

ii) High Quality Motion De-blur

In digital photography, motion blur is a common artifact, especially when using mobile devices or when lighting is low. When the camera is moving relative to the subject during exposure time, this occurs. When it comes to facial recognition applications, such as attendance systems, accurately addressing motion blur is crucial. During motion blur, image details are smeared directionally. Modeling it as a convolution between the sharp image and the blur kernel is possible:

B = I * K + N

Where, B is the blurred image, 'I' is the original sharp image, K is the blur kernel (also called Point Spread Function or PSF) and N is additive noise

Motion blur is removed from a single image using HQMD (17) (High Quality Motion Deblurring). In order to achieve Maximum a Posterior (MAP), blur kernel estimation and unblurred image restoration should be combined. A high quality image is produced by iteratively updating the blur kernel and estimating the latent image. The HQMD refines the blur kernel to estimate the latent image until convergence is achieved. A novel probabilistic model is presented to address common artifacts encountered in existing deblurring methods. They reduce ringing artifacts by modeling the spatial randomness of noise in the blurred image by constraining contrast in low contrast areas. Algorithm 2 describes the flow of the de-blurring work.

Algorithm 2: High Quality Motion De-blur

Algorithm: High Quality Motion De-blur
Input: I_{Blur} **-** Blurred image
Output: I_{DeBlur} **–** Deblurred image

1. Set parameters: λ1, λ2, κ1, κ2, γ.
2. Initialize λ1 and λ2 based on user input.
3. Repeat until convergence:
4. Estimate the latent image and refine the blur kernel.
5. Update the values of λ1 and λ2 by dividing them by κ1 and κ2, respectively.
6. Adjust the weight γ iteratively to ensure convergence.
7. Use an optimization scheme that alternates between updating the blur kernel and estimating the latent image.
8. Incorporate a variable substitution scheme and iterative parameter re-weighting technique to efficiently optimize the algorithm.
9. Check for convergence based on a predefined threshold or number of iterations.
10. Return Obtain the deblurred image as the final result of the algorithm.

End Algorithm

Noise in camera images is a random variation in the image signal caused by several factors. Poor lighting can lead to noise as cameras increase ISO sensitivity to capture more light, resulting in image noise and washed-out photos. High ISO settings further exacerbate noise levels, and most cameras increase noise reduction at higher sensitivities, which can degrade image quality by increasing noise and smoothing out fine details. Additionally, high pixel density reduces the size of each pixel, decreasing the light each pixel captures and making the sensor more susceptible to noise.

3.4 Noise Removal Using FFT and Soft Thresholding

A fundamental task in image processing is denoising to remove noise while maintaining important features. Denoising 2D grayscale images using FFT and soft thresholding in the frequency domain is described in this theory. Using a 2D grayscale image and padding it if necessary to ensure its dimensions are multiples of an 8 x 8 block size, the process begins. Through this step, edge information is not lost due to incomplete blocks. Paddered image is divided into 8 x 8 blocks with no overlap. To facilitate localized noise reduction and image preservation, each block undergoes independent processing.

Each block is transformed from the spatial domain to the frequency domain using a 2D FFT. By dividing the block into its constituent frequencies, low frequencies represent smooth areas and high frequencies represent finer details and noise. By taking the absolute values of the complex FFT coefficients, we can calculate the magnitude spectrum. Within a block, the magnitude spectrum represents the strength of each frequency component. A mean of the magnitude spectrum is computed to adaptively determine the threshold for soft thresholding. As a baseline, this mean represents the block's overall frequency content. Multiplying this mean by a predefined factor yields an adaptive threshold. A proportionally higher threshold will be applied to blocks with higher frequency content, allowing effective noise reduction.Soft thresholding is applied to attenuate noise in the frequency domain. For each frequency component (i, j) within the block:

- Frequency component magnitude relative to adaptive threshold determines attenuation factor.
- By multiplying the original FFT coefficient by this attenuation factor, noise-related frequency components are effectively reduced.

By using an exponential function, soft thresholding minimizes artifacts that may occur from abrupt cutoffs (as in hard thresholding). A 2D FFT is used to transform frequency data to spatial domain after soft thresholding in frequency domain. The denoised block in the spatial domain is obtained by taking the real part of the complex array. Afterwards, the denoised blocks are assembled to reconstruct the final denoised image. By removing any padding initially added to the image, we ensure consistency and completeness.

vi) AI Modeling

Let the model remember photos from a training set. Here is an example tree structure for train_dir:

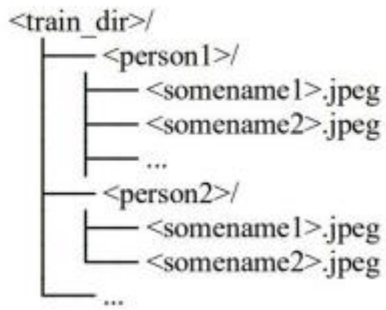

Figure 3: Treestructure of the Training Directory

For each person in the class, let the model loop through each training image ($Train_{Img}$=(IT_1,IT_2..IT_N), Where N is associated with the number of students in the class). A set of training images is called $Train_{Img}$, and a set of face encodings is called E. Suppose x is the image want to add to $Train_{Img}$. To check if there are people in the training image, let y be the label of the current image x, and p(i) be the number of people. When $Train_{Img}$ does not contain any people,

$$\text{if } p(x) == 0: Train_{Img} \cup \{x\} \text{ and } E = E \cup \{face\_encode(x)\}$$

where face_encode(x) is the encoding of the face in the image x.

3.5 MFLCNN: Multiple Face Labelled CNN

Among the applications of facial recognition in computer vision are security and social media. Recently, deep learning has replaced traditional biometric methods with holistic pattern recognition using holistic patterns. For this task, convolutional neural networks (CNN) are highly effective. Face recognition models are benchmarked against the Labeled Faces in College (LFC) dataset. An ideal testbed for assessing model performance is this dataset, which contains a diverse collection of facial images. These datasets have been used by previous models, including MobiFace and FaceNet, to achieve remarkable accuracy.

For this study, cropped facial images of 200 students total 1,000 images. There are 700 training images and 300 testing images, divided into training and testing sets. Images are first resized to 256x256 pixels. A pretrained CNN requires pixel values to be preprocessed before input. To facilitate efficient training and accurate classification, pixel values are normalized and labels are one-hot encoded.The MFLCNN architecture comprises multiple convolutional and dense layers designed to extract and combine hierarchical facial features. Figure 4 shows the MFLCNN Architecture.

Figure 4: MFLCNN Architecture

In the convolutional layer, features are learned from low to high level, while in the dense layer, they are consolidated for classification. A MFLCNN model is trained using the training set for 20 epochs. The parameters of the model are optimized using an appropriate optimizer and loss function. During training, the accuracy and loss are monitored on both training and validation sets to ensure generalization. In the post-training stage, the model is evaluated on 300 unseen images. Measured in terms of accuracy, the model is able to identify the face labels correctly. This test set illustrates the MFLCNN's robust generalization capabilities after 20 epochs. Using convolutional layers to learn hierarchical features and dense layers to classify, we obtain state-of-the-art performance on the LFC dataset. An automatic training scheduler is proposed to ensure the MFLCNN remains effective and up-to-date. It retrains the model periodically to improve accuracy and adaptability to new facial images.

According to facial recognition data, the department layer generates attendance reports for each course and class. These reports can be accessed by tutors and staff assigned to the respective courses, allowing them to monitor student attendance and intervene if necessary. The department layer may also provide consolidated attendance reports for the entire department, which can be reviewed by admins. Figure 5 shows the structure of the database.

Figure 5: Database Structure of this work

3.6 Tutor Layer

This process refers to the role of a tutor or instructor who oversees attendance-taking. As soon as the training set was built, notification was sent to Tutor. It is important the tutor ensures all learners' faces are visible and takes a single photo of all learners using the APP.

$Train_{Img}$ can receive the Attendance System once the training set of student faces is ready. It takes a single image of the group of students using the APP with their credentials after confirming all students' faces are visible. The APP sends a single image (I), current date and time ($D_{Current}$), and tutor ID ($Tutor_{ID}$) to the server. In $Train_{Img}$, the appropriate ML Model is selected automatically based on the subjects of the students. In the Tracker, the APP receives I, and the ML Model matches it to the existing dataset of student faces S. In the database DB, the model returns the student's REG_{ID}. Upon receiving the stored information, $Train_{Img}$ verifies, confirms, and stores it into the Database DB.

The credential verification is performed through mobile apps and allows the subject and class staff to take attendance. In the following step, staff can take group images of students through the App, which are then sent to the server. For the generation of the AI model, the received group of students' images is obtained. After that, the AI model-MFLCNN classifier recognizes the uploaded image with the face image uploaded by the students.The $Face_{Detect}$ algorithm identifies faces in the captured students' group images. Faces are detected and $compared_{Img}$ with images in the database. When a match ($Match_{Img}$) is found, the identified face, R, of the student is marked as present. Staff verify the predicted result, $Predict_{Img}$, through the mobile app. Through the mobile app, the staff captures the group or class of student images G_{Img} during each class session and submits them to the server within the target timeframe T_{Time}. In G_{Img}, faces are detected and compared with database images. An attendance mark is made if a match ($Match_{Img}$) is found. After the session, the faculty member T_i who is handling the session is shown a list of absentees.

3.7 Student Layer

Each learner, Li, is identified with their Register Number and Date of Birth (RD), which serves as their login credential. To avoid errors in the attendance marking process and improve the robustness of the face recognition system, each student needs to upload multiple poses 5) of face images, PoseImg = {Pose1, Pose2, ..., Pose5}.

In real-world settings, multiple poses are necessary for facial recognition systems. By capturing left-right and top-bottom views along with the straight pose, the system can handle variations in head orientation. Multi-pose approaches improve facial recognition algorithms' accuracy and reliability by extracting comprehensive facial features. Aside from facial expressions, partial occlusions from accessories, and varying lighting and camera angles, it takes into account environmental factors. The system is more robust and effective in diverse environments when it uses multiple poses.

A 180-degree rotation is achieved through the use of Left-Right and Top-Bottom face images, along with the straight pose image, Image Straight. A single person identification, ID, is stored for the five pose face images. Once all students upload their photos on the server, a notification, N, will be sent to the department, D, so that it can make the training set, TrainImg. By rotating the images in different directions, the system can account for different head orientations and positions by using multiple poses of face images.Using a Register Number and Date of Birth as login credentials assumes that each student's RD is unique and not shared with anyone else. Duplicate RDs may not be distinguished by the system if there are multiple students with the same RD. By uploading students' face images and creating the training set, the system can recognize and identify each student's face during attendance tracking using machine learning algorithms. Based on the features of each student's face, the system can determine whether the student is present or absent. Figure 6 displays the sample dataset images.

Figure 6: Sample Dataset

4. ALGORITHMS

This section includes the algorithms employed for the process of detecting the faces from the group of the students, placing attendance for the particular classes and attendance report system process is illustrated as follows:

Algorithm 3: Face Detection Algorithm

Input: T = {T_1,T_2, ...T_n} – Tutor Details, S= {S_1, S_2, ... S_n} – Students Details, S_{Img}- Students Image set
Output: P_{Label} – Predicted Label
Parameters: N- Total no. of Instances
Algorithm: Face Detection

1. Read the student details S (name, RD, Department…)
2. Fetch the facial images of the students S
3. If staff_ credentials= True, then
4. For each i=0 to N
5. Load the $Train_{Img}$ for the face prediction ($Train_{Img}$={IT_1,IT_2..IT_N})
6. Train the MFLCNN Classifier model with $Train_{Img}$
7. Set the corresponding Student ID as Training labels
8. For j = i+1 to n
9. Calculate distance between training samples d(i,j)
10. sort the distance from smaller to largest by sorted(d)
11. Select training samples from smallest distance
12. nearest_neighbors = {i | d(i,x) ∈ sorted(1:K)}
13. Count the occurrence probability and weighted probability
14. Fetch the images based on the labels
15. if Current_Image not exists then
    a. Compare the captured facial image to the stored student facial images
    b. Predict the closest matching student_ID number
    c. predicted_label (P_{Label}) = MFLCNN_classifier(x,$Train_{Img}$)
16. End if
17. End for
18. End For
19. End if
20. Return P_{Label}

This algorithm 3 automatically marks attendance by detecting faces and predicting student identities based on MFLCNN. The algorithm uses tutor and student information, as well as student facial images, to predict student IDs. First, it reads the student's name, roll number, and department to retrieve their facial image. Staff credentials are used to load all student training images and train the MFLCNN model using these images and student IDs. Each new student image is compared to all training samples, and the closest K neighbors are selected based on their labels to determine the most likely student ID. It compares the image to stored ones and uses the MFLCNN model to predict the closest matching student ID, finally returning the predicted label.

Algorithm 4: Attendance Process

Input: $Stud_{Img}$ – Students Current Image
Output: Att – Mark Attendance
Algorithm: Attendance Process

1. Build a training set of student faces (S)
2. $Train_{Img}$ confirms that all students' faces are visible
3. Takes a single image (I) of the group of students using the app.
4. $Train_{Img}$ collects the student's credentials, current date and time ($D_{Current}$)
5. and tutor ID ($Tutor_{ID}$) through the app and sends it to the server.
6. The Tracker receives image (I) from the app and forwards it to the appropriate ML model.
7. The model returns the student registration ID (REGID)
8. Check if the predicted student ID has not marked attendance yet
9. If match found:
10. Mark attendance for that student
11. Update the attendance record
12. Notify the student that attendance is marked

End Algorithm

As input, the algorithm 4 takes an image of students and marks their attendance. To recognize student faces, it builds a training set. Upon capturing a new image, it detects student faces and returns their IDs. Then it checks if the student has already marked attendance. Upon failing to mark attendance for a student, the student's records will be updated. In this way, all captured images are processed.

Algorithm 5: Attendance Monitoring and Reporting

Input:

- $S = \{s_1, s_2, ..., sn\}$ (set of students)
- $RD = \{(r_1, d_1), (r_2, d_2), ..., (r_n, d_n)\}$ (register number and date of birth for each student)
- $Img = \{Img_1, Img_2, ..., Img_n\}$ (set of face images for each student, where $Img_i = \{pose_1, pose_2, ..., pose_5\}$)
- $T = \{t_1, t_2, ..., t_n\}$ (set of tutors/staff)
- $D = \{d_1, d_2, ..., d_n\}$ (set of departments)
- $C = \{c_1, c_2, ..., c_n\}$ (set of courses)

Output:

- $Att = \{att_1, att_2, ..., attn\}$ (set of attendance records for each student)

Algorithm: Attendance Monitoring

1.Admin Setup

For all d_i in D, create department d_i
For all t_i in T, for all c_n in C, assign tutor t_i to course c_n

2.Student Registration

For all s_i in S:
Register with unique credentials (r_i, d_i) in RD
Upload $Img_i = \{pose_1, pose_2, ..., pose_5\}$
Store Img_i in the database along with student details s_i

3.Training Set Generation

For all c_i in C:
$S^c = \{s_n \mid s_n \text{ is enrolled in course } c_i\}$
$TrainSet^c = \bigcup_n Img^j$, for all s_n in S^c
Train the machine learning model MLM^c using $TrainSet^c$

4. Attendance Marking:

For each class session k:
Tutor/staff t_i logs in and confirms readiness for attendance marking
t_i captures a group image GImg of students in the classroom
Upload GImg to the server
$Faces = \{f_1, f_2, ..., f_n\}$ = detect individual faces from GImg
For all f_i in Faces:
Compare f_i with $TrainSet^c$ using MLM^c
If match found with student s_n:
Retrieve details of s_n
att_n = present
Else:
att_n= unknown/unrecognized
Update Att in the database
Send attendance notifications to students and tutors/staff

5. Attendance Monitoring and Reporting:

For all s_i in S,
s_i can view att_i
For all t_i in T,
for all c_i
assigned to t_i, t_i can access attendance reports for c
Admin can monitor overall Att

6. System Maintenance:

Admin performs system maintenance tasks
For all d_i in D, update course information, student lists, and tutor assignments for d_i
For all s_i in S, s_i can update their face images or personal information
For all t_i in T, t_i can provide feedback or report issues

End Algorithm

In this algorithm 5, sets, elements, unions, and other operations are described using proper mathematical symbols and notations. Students register, training sets are generated, attendance is marked using facial recognition, attendance monitoring and reporting, and system maintenance is discussed. Multi-student, course, class session, and other aspects of the system are handled with loops and conditionals.

4.1 Implementation And Experimentation

In the proposed architecture, Client Side development done with HTML, CSS, JQuery. Sever side and Deep learning model developed with Python (Djangon) is used in ATOM, Anaconda, and MySQL and executed on a Windows 10 64-bit system environment with an Intel Xeon Processor with 16 GB RAM and 1 TB hard drive. Face images are collected from different departments in the college for test data. After applying the De-Blurring algorithm, the blurred images are de-blurred as shown in Figure 7.

Figure 7: De-Blur Image

Figure 8 shows the denoised image for the input image. The figure depicts that the original image and its denoised image after applying filtering process.

Figure 8: De-Noised Image

The separate face detected images are shown in Figure 9. After the pre-processing the input image, the image is feed to place the attendance for the particular class, then the group of the student's image is processed with the FaceDetection Algorithm to detect every single faces of the class is depicted in Figure 9.

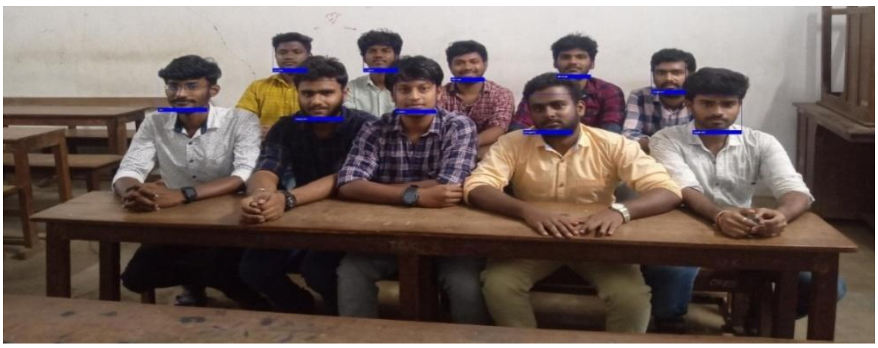

Figure 9: Detected Images

According to the performance metrics presented, Department D6 is the best performer across all measures, with the highest accuracy (99%), lowest error rate (1%), and best sensitivity, specificity, and F1-score is shown in Table 2. It appears that departmental performance improved from D1 to D6, with D5 and D6 significantly outperforming the rest.

Table 2: Performance Metrics for Department Wise Evaluation

Department	Accuracy	Error Rate	Sensitivity	Specificity	F1-Score
D1	96.5	3.5	96.5	97.2	96.84
D2	97	3	97.5	96.5	96.9
D3	97.5	2.5	98.5	97.5	97.9
D4	98	2	97.7	96.6	97.14
D5	98.5	1.5	98.5	98.5	98.5
D6	99	1	99.4	99.1	99.2

The evaluation of the department wise analysis of accuracy and error rate is shown in Figure 10. Then the performance metrics such as sensitivity, specificity and F1-Score is illustrated in Figure 11.
Accuracy in the context of a student face detection system for attendance refers to the percentage of correctly identified student faces out of the total faces detected by the system. Error Rate for this system represents the percentage of incorrectly identified or missed student faces out of the total faces detected(22,23).

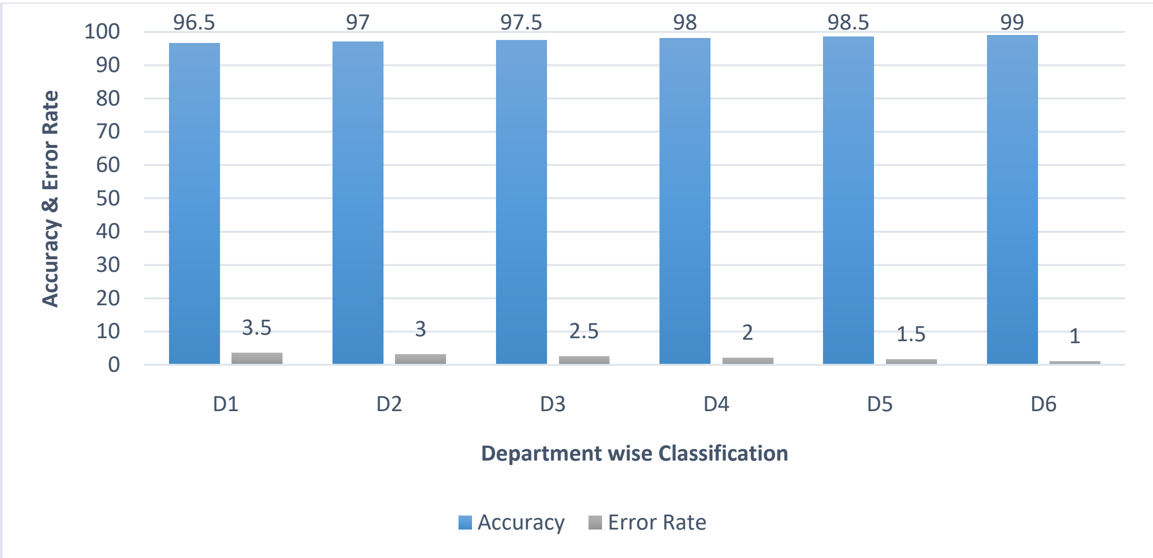

Figure 10: Accuracy and Error Rate

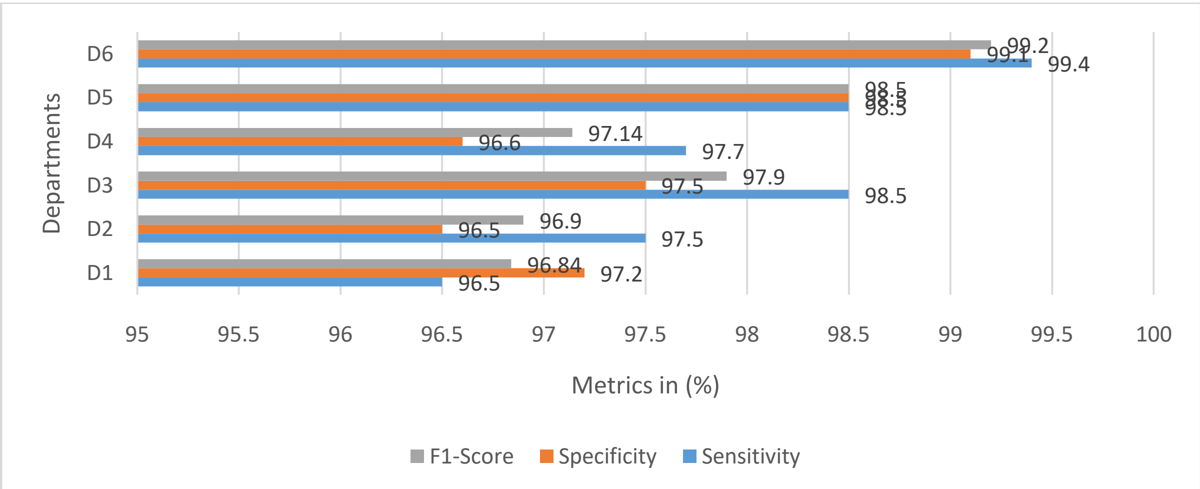

Figure 11: Specificity and Sensitivity

In comparison to KNN and Gradient Boost, MFLCNN shows superior performance across all measures is shown in Table 3. The test has the highest accuracy (97.5%), lowest error rate (2.25%), and best sensitivity, specificity, and F1-score. However, Gradient Boost falls short of MFLCNN, outperforming KNN. According to this comparison, MFLCNN is the most effective classifier, followed by Gradient Boost, and KNN is the least.

Table 3: Classification Techniques Performance Evaluation

Classifier	Accuracy	Error Rate	Sensitivity	Specificity	F1-Score
MFLCNN	97.75	2.25	98.01	97.56	97.74
KNN	87.5	12.5	85.2	80.4	82.73
Gradient Boost	91.3	8.7	90.7	95.8	93.18

Figure 12compares MFLCNN, KNN, and Gradient Boost accuracy and error rates. MFLCNN outperforms the other two with the highest accuracy (97.75%) and lowest error rate (2.25%). The figure suggests MFLCNN is the most effective classifier, followed by Gradient Boost, and KNN is the least reliable.

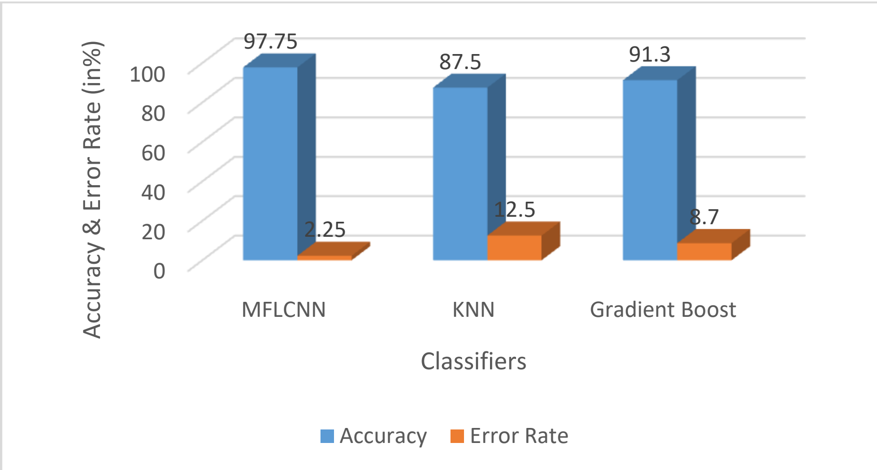

Figure 12: Accuracy and Error Rate for Classifiers

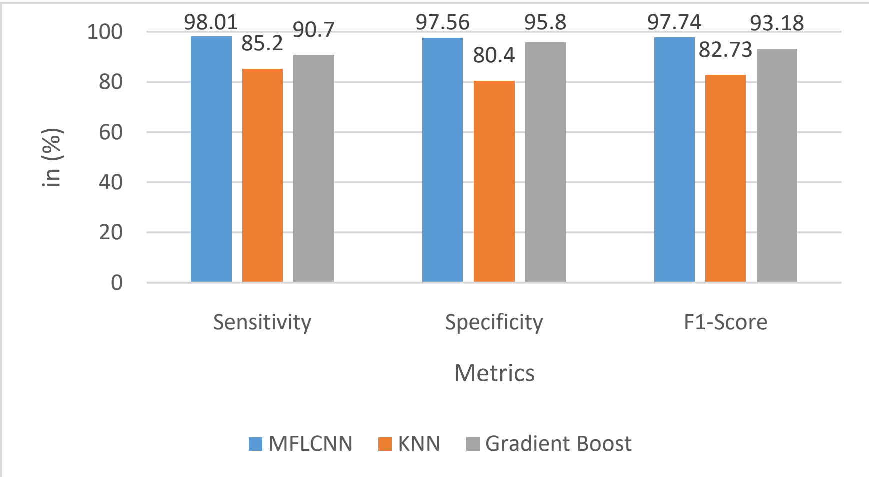

Figure 13: Performance Evaluation for the Classifiers

The MFLCNN consistently outperforms both other classifiers in all three metrics, including Sensitivity (98.01%), Specificity (97.56%), and F1-Score (97.74%). As shown by the data, the MFLCNN is the most balanced and effective classifier among the three, demonstrating superior performance in both positive and negative identification (Sensitivity and Specificity), along with overall balanced performance (F1-Score).

4.2 Discussion

The work (18) proposes an attendance management system using face recognition. It uses the Haar cascade algorithm for face detection and the Local Binary Patterns Histogram (LBPH) algorithm for face recognition. The system captures real-time images, matches them with a pre-trained dataset, and automatically updates attendance in an Excel sheet. This system faces several limitations, including single-face recognition, limited real-time interaction, reliance on local processing, basic attendance recording, and limited flexibility in handling various environmental conditions. The work by Sanika Kulkarni et al. (19) proposed an automated attendance system using facial recognition technology. The system uses CCTV cameras to capture student images as they enter the classroom or institution, automatically recognizing and marking their attendance without manual intervention. The approach employs techniques like HOG (Histogram of Oriented Gradients) for face detection and a deep convolutional neural network for face encoding. It may struggle with varying lighting conditions, camera angles, and multiple face recognition simultaneously. The system also appears to rely on local processing, which could limit scalability for larger institutions.

Joel Biju et al. (20) presented an automated attendance system using facial recognition technology. The Haar Cascade classifier is used to detect faces, PCA is used to extract features, and LBPH is used to recognize faces. Different lighting conditions, facial poses, and occlusions may cause the system to fail. For larger institutions, it may restrict scalability due to local processing. Using facial recognition technology, (21) sought to modernize traditional attendance tracking. With this desktop-based application, attendance management is simplified, saving time and resources. Although the system is accurate, it suffers from limitations like lighting conditions and privacy concerns. Changes in student appearances can also hamper facial recognition's effectiveness.

To address these issues, the proposed solution combines advanced artificial intelligence algorithms with robust data protection measures. Through a comprehensive set of features, Attendance Wizard addresses common limitations of facial recognition attendance systems. It incorporates multi-face recognition to process large groups efficiently, cloud-based processing to enhance scalability, and an interactive mobile interface to improve user access. For improved accuracy and adaptability to different environmental conditions, the app uses deep learning algorithms and a MFLCNN Classifier. Continuous real-time data capture and processing, automated notifications, and integration with existing databases are also available. By enhancing the Attendance Wizard, you will get an even more efficient, scalable, and accurate attendance tracking solution.

5. CONCLUSION

A mobile app and face recognition technology can improve the attendance monitoring and management process significantly compared to traditional methods, as demonstrated in this study. By matching students' faces with their images in the database, the system can mark attendance more accurately than manual methods. Traditional paper-based or manual attendance systems cannot capture student images continuously and evaluate attendance in real-time. The automated attendance marking system reduces human errors and makes the system more reliable. This system may save time by quickly processing student images and marking attendance. This research, however, has some limitations. The proposed system relies on face recognition technology for accuracy and reliability. It should be noted that environmental factors and image quality can have an impact on the accuracy of face matching. It may be necessary to improve the AI model further and to collect a larger database of students in order to refine it further.

Future research would be able to improve the system in many different ways, including making it robust enough to handle images that have poor quality, blurriness, different lighting conditions, etc., which may affect the ability to match faces. This app can be enhanced in the future so that real-time attendance updates can be sent to students, parents and faculty members, improving transparency among all stakeholders. The system can be integrated with class schedules and timetables in order to automatically determine which students are required to attend a specific class depending on their class schedule. There is a possibility that this can reduce the amount of false positives.

REFERENCES

[1] Matilda S,Shahi K. Student Attendance Monitoring System Using Image Processing. IEEE International Conference on System. Computation, Automation and Networking ICSCAN. 2019; 1-4. doi: 10.1109/ICSCAN.2019.8878806.

[2] Menakadevi T, Mahesh kumar G, Manoj kumar M, Pavan Kalyan P, Manu S. Automated Attendance System Using Image Processing. International Research Journal of Engineering and Technology IRJET. 202;.08(SI):105-107.

[3] Anzar SM, Subheesh NP, Panthakkan A, Malayil and Ahmad. Random Interval Attendance Management System RIAMS: A Novel Multimodal Approach for Post-COVID Virtual Learning. IEEE Access. 2021; 9(1):91001-91016. doi: 10.1109/ACCESS.2021.3092260.

[4] Gupta SK, Reddy Guddeti RM. CVUCAMS: Computer Vision Based Unobtrusive Classroom Attendance Management System. IEEE 18th International Conference on Advanced Learning Technologies ICALT. 2018; 1(1):101-102. doi: 10.1109/ICALT.2018.00131.

[5] Roy P, Saha P, Aditi SW. An Automated and Scalable Tool for Fingerprint based Biometric Attendance Management System. International Conference on Electronics, Communications and Information Technology ICECIT. 2021; 1-4, doi: 10.1109/ICECIT54077.2021.9641418.

[6] Abbas Helmi RA, Salsabil bin Eddy Yusuf S, Jamal A, Bin Abdullah MI. Face Recognition Automatic Class Attendance System FRACAS. IEEE International Conference on Automatic Control and Intelligent Systems I2CACIS. 2019; 50-55. doi: 10.1109/I2CACIS.2019.8825049.

[7] D'Souza JWS., Jothi S, Chandrasekar A. Automated Attendance Marking and Management System by Facial Recognition Using Histogram. 5th International Conference on Advanced Computing and Communication Systems ICACCS. 2019; 66-69. doi: 10.1109/ICACCS.2019.8728399.

[8] Siva Narayana A, Ramanjineyulu T, Raj Kumar K. Employee Attendance Management System Using OpenCV. International Journal of Creative Research Thoughts. 2022;10(6):39-46.

[9] Diwakar Dhillon, Gosiya Kaleem, Deepanshu Kumar, Mohit kumar. Smart Attendance System using Face Recognition. International Journal of Innovative Science and Research Technology. 2022; 7(4):925-930.

[10] Aziza Ahmedi, Suvarna Nandyal. An Automatic Attendance System Using Image processing. The International Journal of Engineering and Science. 2015; 4(1):01-08.

[11] BasheerS, Nagwanshi KK, Bhatia S. FESD: An Approach for Biometric Human Footprint Matching Using Fuzzy Ensemble Learning. IEEE Access. 2021; 9(1):26641-26663. doi: 10.1109/ACCESS.2021.3057931.

[12] Jain AK, Chen Y, Demirkus M. Pores and ridges: High-resolution Fingerprint Matching Using Level 3 Features. IEEE Transactions on Pattern Analysis and Machine Intelligence.2007; 29(1):15-27.

[13] Sinha GR,Pyae Sone Oo. Introduction to Biometrics and Special Emphasis on Myanmar Sign Language Recognition. Springer e-books. 2019; 1-23. doi: 10.1007/978-3-030-30436-2_1.

[14] Sinha G, Patil S. Biometrics: Concepts and Applications. Publisher: Wiley.2013; USA

[15] 15Wild P. Single-sensor hand and footprint-based multimodal biometric recognition. 2008; Available: http://wavelab.at/papers/Wild08a.pdf.

[16] Tippavajhala Sundar Srinivas, Thota Goutham, Senthil Kumaran. Face Recognition based Smart Attendance System Using IoT. International Research Journal of Engineering and Technology. 2022; 9(3):182-186.

[17] Shan Q, Jia J, Agarwala A. High-quality Motion Deblurring from a Single Image. ACM Transactions on Graphics. 2008; 273(1):1-10. doi: 10.1145/1360612.1360672.

[18] Yadav A, Kumar S, Gupta V, Kumar S. Attendance Management System using Face Recognition. Zenodo CERN European Organization for Nuclear Research. 2023; doi:https://doi.org/10.5281/zenodo.7577140

[19] Kulkarni Sanika,Choudhari Deepti. Attendance Monitoring System using Face Recognition. 5th ICCIP Conference. 2023; 1-8. doi:doi.org/10.2139/ssrn.4669196.

[20] Biju Joel, Sairam, Shreya Kumar, Kishore. Enhancing Attendance Management Systems Using Facial Recognition. International Journal of Engineering Research and Technology (IJERT). 2024; 13(1):1-5. doi:10.17577/IJERTV13IS010022.

[21] Pradyumna J, Touqeer Khan, Kiran Kumar. Smart Attendance System using Face Recognition. International Journal for Multidisciplinary Research IJFMR. 2023; 54(1):1–3. https://www.ijfmr.com

[22] Govindaprabhu GB, Sumathi M. Ethno medicine of Indigenous Communities: Tamil Traditional Medicinal Plants Leaf detection using Deep Learning Models.Procedia Computer Science. 2024; 235(1):1135-1144. https://doi.org/10.1016/j.procs.2024.04.108.

[23] Govindaprabhu GB, Sumathi M. Safeguarding Humans from Attacks Using AI-Enabled (DQN)Wild Animal Identification System. International Research Journal of Multidisciplinary Scope (IRJMS), 2024; 5(3): 285-302.DOI: 10.47857/irjms.2024.v05i03.0697