Attendance 2.0: Facial Recognition Meets Web Technology

Nashita Nawar Prapti[1], **Rakibul Hasan Showrov**[2], **Nabil Hasan Mumat**[3], **Kanij Fatema Mim**[4], and **Md. Shahriar Hussain**[5]

[1]Undergraduate student, Department of Electrical and Computer Engineering, North South University, Dhaka
[2]Undergraduate student, Department of Electrical and Computer Engineering, North South University, Dhaka
[3]Undergraduate student, Department of Electrical and Computer Engineering, North South University, Dhaka
[4]Undergraduate student, Department of Electrical and Computer Engineering, North South University, Dhaka
[5]Senior Lecturer, Department of Electrical and Computer Engineering, North South University, Dhaka
[1]nashitanawar@gmail.com, [2]rakibulhasanshowrov@gmail.com, [3]nabilhasan162451@gmail.com,
[4]kfmim2616@gmail.com, [5]shahriar.hussain01@northsouth.edu

Abstract

Facial recognition systems use complex algorithms and artificial intelligence, or AI, to recognize and authenticate folks based on their unique traits. This is a groundbreaking, non-contact, or highly efficient technique. These systems record and analyze an individual's facial features, including the distance between their eyes, nasal shape, chin, and other unique features. Several machine learning and deep learning algorithms, involving MTCNN+FaceNet+SVM, MTCNN+FaceNet+KNN, Random Forest, VGG16, CNN-Sequential Based, and EfficientNetB0, were utilized in this project. We found accuracy for testing, validation, and training for the previously mentioned algorithms. Applying MTCNN+Facenet+SVM created the best accuracy. The result was 100% accuracy in training, 100% accuracy in validation, and 100% accuracy in testing. In contrast to the other algorithms, it is better at predicting faces. Furthermore, intending to incorporate our facial identification system effortlessly, we established a comprehensive Django-based website. Real-time monitoring of attendance, dynamic front-end interfaces, and authentication for users are all displayed on this website. Given that our Django platform employs SQLITE3 for the database and HTML, CSS, and JavaScript for the front end, it ensures a user-friendly experience while maintaining robust performance and security.

Keywords: Face Recognition, attendance, Django-based website, machine learning, deep learning, MTCNN+FaceNet+SVM, MTCNN+FaceNet+KNN, Random Forest, VGG16, CNN-Sequential Based, EfficientNetB0.

1. INTRODUCTION

1.1 Background and Motivation

For organizational or organizational leadership to be efficient and transparent in several situations attendance tracking is essential. Traditional approaches for handling enrollment, which rely on manual inputs or identification cards, are typically prone to mistakes. The invention of facial recognition technology offers an innovative approach to upgrade attendance systems in reaction to these restrictions. Face recognition technology assesses and contrasts facial characteristics to identify or verify humans. It requires time and energy to call a student's name or roll number for attendance. All of these issues might be solved with an automated attendance system.

Face recognition enhances accuracy and efficiency compared to traditional, error-prone, time-consuming attendance methods. Ensuring that only authorized people can mark their attendance, may improve security. Furthermore, it enables contactless attendance, which reduces sickness, provides accurate timestamps for attendance, reduces the chances of fraud or time manipulation by employees or students, and offers to boost satisfaction.

1.2 Purpose and Goal of the Project

In numerous instances, recording attendance is crucial for organizational or organizational leadership to be transparent and efficient. The system's primary function is to identify and recognize faces in real time, match them to data in the database, and record attendance. The goal is to improve the time-consuming manual attendance procedure and create a more accurate system than existing solutions. Calling a student's name or roll number for attendance requires effort and time. A restricted attendance system can address each of these issues. The system is intended to be user-friendly, for administrators, students, and faculty. The initiative intends to reduce instances of proxy attendance by implementing facial authentication, ensuring that attendance records accurately represent the presence of registered individuals. It enables the system to adjust to changing environments, lighting, and variations in facial appearances.

2. RESEARCH LITERATURE REVIEW

2.1 The article "Face Recognition Attendance System Using HOG and CNN Algorithm" [1] offers a creative method for automating the recording of attendance in a classroom. The writers discuss the drawbacks of conventional attendance records and offer a workaround that makes use of facial recognition technology. The suggested system's [1] adaptability to a range of real-world issues, including changes in lighting, head movements, and camera distances from the face, is one of its strong points. The device uses video footage that is carefully positioned throughout the classroom to identify faces through facial detection and recognition. The article provides an in-depth description of the system's architecture, including key steps like image acquisition, face detection, feature extraction, and SVM comparison, and its Excel-based attendance records. There was no mention of a central Attendance Management System in the Article.

2.2 The article "Face Detection & Recognition from Images & Videos Based on CNN & Raspberry Pi" [2] presents a real-time surveillance framework for facial recognition using Raspberry Pi and Convolutional Neural Network (CNN). The authors compare the classification accuracy of their CNN-based system with other methods, such as the Histogram of Oriented Gradient (HOG) feature extractor and state-of-the-art face detection and recognition techniques. The system's performance is evaluated under challenging conditions, such as masks or sunglasses and live videos. The results show high accuracy rates, with the system achieving the highest accuracy of 98% for the VMU dataset, 98.24% for face recognition, and 89.39% for the 14 celebrity datasets. These findings contribute to advancing computer vision techniques and have potential implications for security surveillance and IoT-based applications.

2.3 A research study on the creation of an intelligent attendance system that makes use of facial recognition technology is presented in the article titled "Intelligent Attendance System with Face Recognition using the Deep Convolutional Neural Network Method." [3] The goal of the system was to automatically identify and log lecture attendance from students. The study's justification is made evident in the paper, which also highlights the necessity of a more automated and efficient system and stresses the value of attendance in educational settings. Because facial recognition is biometric and can identify several people at once, the authors suggest it as a workable solution to the shortcomings of current attendance technologies like fingerprinting, RFID, and QR codes. The accuracy of the attendance system under various scenarios was assessed by the authors through trials with a group of sixteen pupils. The findings show that students' accuracy rates were 81.25% when they faced forward, 75.00% when they faced laterally, and 43.75% when they faced downward. The study's generalizability is limited by a small sample size and lacks detailed information on the training and testing process of a deep convolutional neural network, which could have added value to the study.

2.4 Wenjin Xu et al.'s paper, "Study on Facial Recognition Method Based on YOLOv5" [4], offers a deep learning strategy to overcome the drawbacks of facial recognition algorithms. As we know, eyebrows can be a good choice for research. The paper highlights the significance of eyebrows in facial recognition and the challenges associated with traditional methods that require manual extraction of pure eyebrow images. The authors introduce the YOLOv5-Attention method. It is important to note that the research mainly focuses on the application of brow recognition to generic facial recognition. Although this particular component has been thoroughly investigated and is substantiated by experimental findings, a more comprehensive discussion of the wider framework of facial recognition could have been provided.

2.5 In the article named "Efficient Face Recognition System for Operating in Unconstrained Environments,"[5] author Alejandra Sarahi Sanchez-Moreno, used Deep learning techniques FaceNet standard classifiers such as SVM, KNN, and Random Forest. For face detection, the system uses YOLO-face. The accuracy of the FaceNet+ SVM model was 99.7 %, FaceNet+KNN was 99.5 %, and FaceNet+RF model was 85.1 %. The system was capable of operating in highly complex environments. The limitation was when a GPU is available, the YOLO-Face scheme provides better recognition results with faster processing speed.

2.6 In an article entitled "Face Recognition Based Attendance System,"[6] authors Nandhini R, Duraimurugan N, and S.P. Chokkalingam used the CNN algorithm to detect faces. The system captures a video of students, which is then divided into frames, and these frames are stored in a database. The system was capable of extracting the image from the video and then giving the attendance. However, the limitation was it was not implemented in larger areas like in a seminar hall where it helps in sensing the presence of many people.

2.7 In the article "Performance Evaluation of Convolutional Neural Networks (CNNs) And VGG on Real-Time Face Recognition System", [7] authors Showkat Ahmad Dara, and S Palanivel processed color images to recognize and detect faces with a good deal of accuracy in a real-time scenario. CNN (Convolutional Neural Network), along with VGG-16, has been used to enhance recognition accuracy. Both of the classifiers were performed sequentially and were implemented with 1056 face images of 24 different persons. The accuracy of CNN was 69.09%, while VGG16 gave a higher accuracy of 96.53%. So, the limitations of this paper are that this dataset doesn't include human faces that come under several conditions that include camera variation, pose, scale, and illumination.

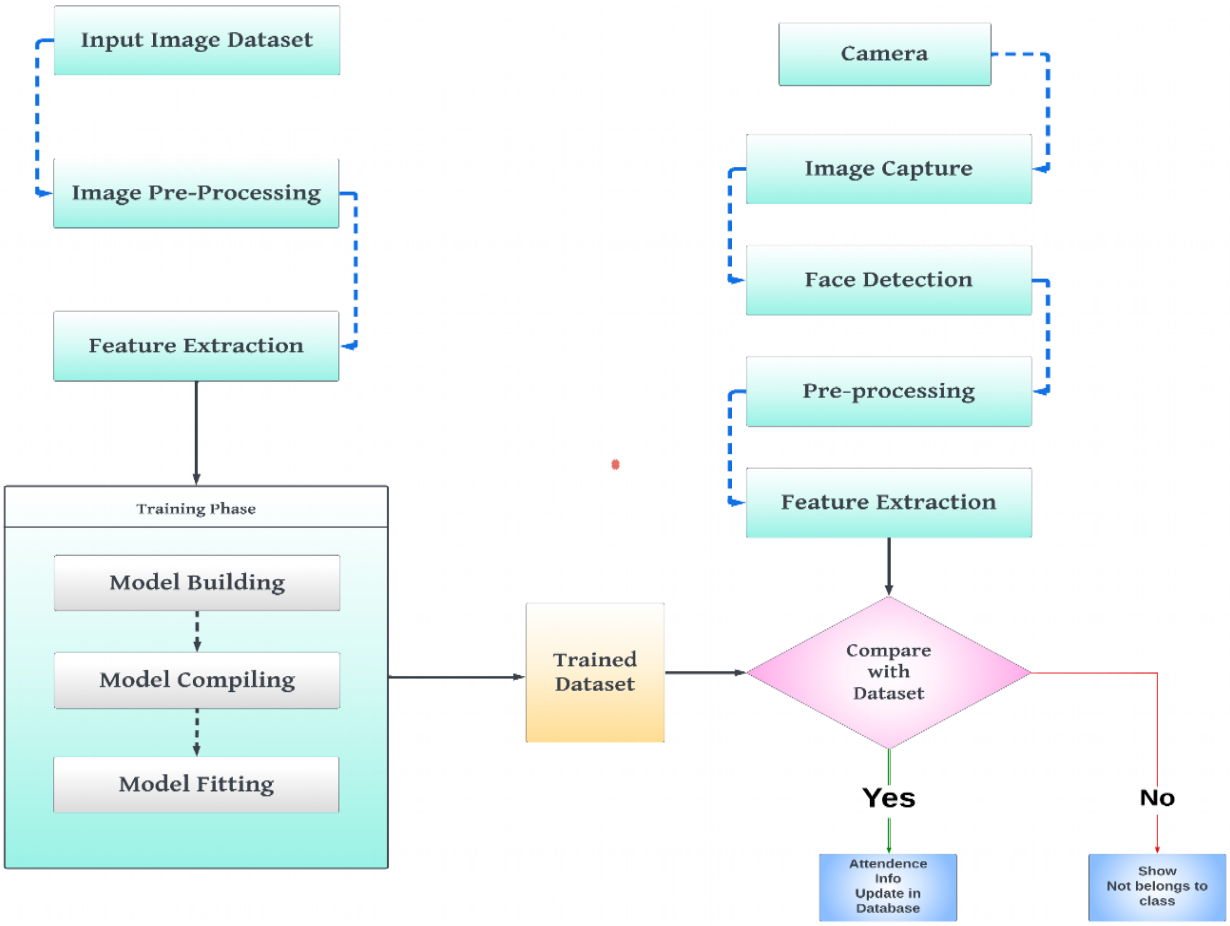


Fig. 1: Working Structure of The System

3. METHODOLOGY

3.1 System Diagram

In below figure 1, we can see the basic structure of how we have trained the model and how the system will take the attendance automatically. We have used several models and methods to train. For training all the models we had to follow some steps. At first, we collected images of 18 persons. Then we split the dataset automatically. But after some time for better performance, we augmented the images and split the dataset manually into three sets(train, validation, and test). Then we preprocessed the images and reshaped them to 160*160*3, as all the images are RGB images before feeding them to the extractor. After extracting the features we save the feature embeddings for further use. Then we passed the training set feature embeddings to the model for training it. After training the model we used the trained model to classify the validation set for understanding model performance and finally tested the model performance.

In the above image, we can see the structure of the whole system, where we have shown the fundamental structure of the website as well. We have developed a website to manage attendance and display student records. Not only that, our website can also keep the record of the faculty members and all their courses and the enrolled students of their sections for further evaluation. We have created two user types. One of them is the faculty members who have access only to their assigned courses and they can only manage attendance of these courses. The other type of user is called Admin users who have comprehensive control over the entire system and can access the database directly. Faculty members do not have direct access to the database. Students can enroll in multiple courses, and their attendance records are maintained only for the sections to which they were assigned. This ensures accurate tracking of student attendance, specific to each section. So when the faculty member takes the attendance the website will automatically initiate a camera that will continuously take images of the students entering the classroom and then preprocess the images as soon as the system collects each image. Then the system will extract the features and pass them to the model for classification. If the model predicts someone then the system will match it with the database if the student is enrolled in that section. If the student is enrolled in that section then the system will update the attendance of that student otherwise not.

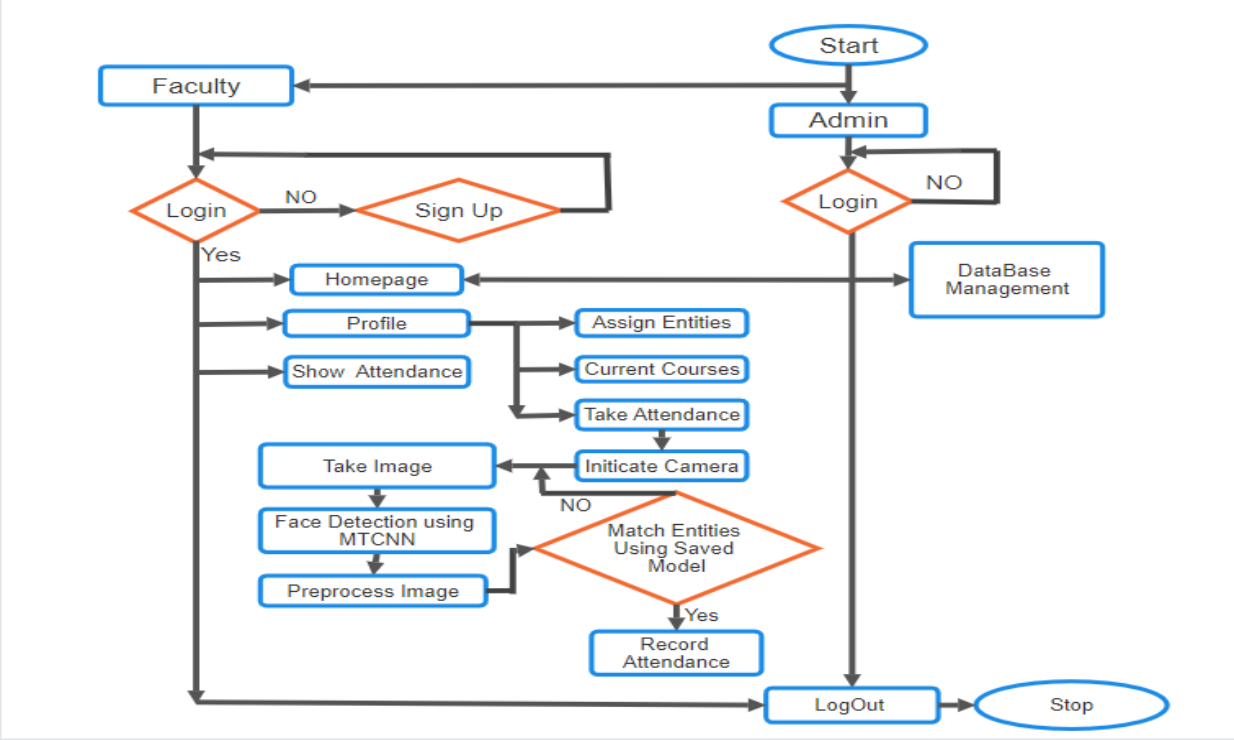


Fig. 2: Working Structure of the website-based attendance system

3.2 Hardware and/or Software Components

Different Model structure used in the system:

Feature Extraction Method of FaceNet Model:

In the below image, we can see the structure of the Facenet model. Facenet uses the inception module as well as skip connection along with its structure. It has a total of 22 layers and it gives a feature embedding of dimension 128 [9]. From which I have shown a tentative structure in the above image. It has max-pooling layers, Convolution layers, and also has inception module, and finally the fully connected layers [9]. We will show the inception module's benefits and working process later. Skip connections solve the vanishing gradient problem. When the model goes deeper accuracy gets saturated and for that, after each layer, the input value is also passed to the output of that layer [10]. So, the output after passing the input x through the layer can be expressed as:

H(x) = F(x) + x

where:

- x is the input to the layer,
- $F(x)$ is the output of the transformation applied to x,
- $H(x)$ is the final output after adding the skip connection.

In this context, $F(x)$ represents the output of the transformation applied to the input x. The skip connection adds the original input x directly to the output of the transformation $F(x)$. This allows the network to retain information from the input and helps improve the learning process by alleviating issues such as vanishing gradients and enabling the training of deeper networks. Finally, Facenet model gives a 128 D facial embedding where the same person's representation should be closer and other persons representation should be far apart in the embedding space[11]. This is achieved through the use of a cost function, specifically the triplet loss function.FaceNet typically employs a combined triplet loss and regularization function to mitigate overfitting and optimize embedding quality.

The goal of the triplet loss function is to ensure that an anchor image is closer to a positive image (same person) than to a negative image (different person) by a margin.

The triplet loss function can be formulated as:

$$L(a, p, n) = \max\left(d(a, p) - d(a, n) + \alpha, 0\right) \tag{1}$$

where:

- a is the anchor image.
- p is the positive image (same person as the anchor).

Tool	Functions	Other similar Tools (if any)	Why selected this tool
Camera	capturing Image for Dataset	No	To capture Image of a person
Opencv	Connect With Camera	—---	To capture real-time image by using camera
MTCNN	preprocess the image		MTCNN shows better performance for processing images.
FaceNet	Extract the feature		Because it is an advanced part of CNN (Convoluted Neural Network) developed by Google researchers
SVM,KNN.----------	classify and train the model		to see which model shows better accuracy
CNN	classify and train the model		to see which model shows better accuracy
EfficientNetB0	classify and train the model		to see which model shows better accuracy
VGG16	train the model		to compare the accuracies
Random Forest	classify and train the model		to compare the accuracies
Google Collab	Train and test basic Model	pycharm	Create ML model and Show their output
Processing Unit	To handle the image processing and facial recognition tasks	—---	As we know we have to process the data and tasks.
Networking	connect All Component efficiency	——	To connect all the component efficiently
Storage Unit	store the Image		We need to save the image of people
Django	Website framework		Building website and its backend
Database	Store Data		Can be used MySQL but we have used Sqlite3 database.

Fig. 3: List of Hardware/Software tools

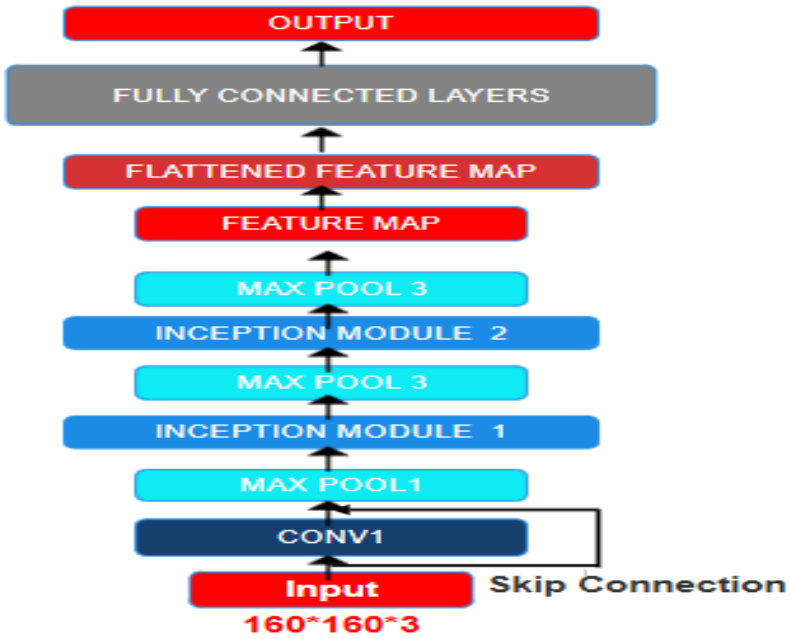


Fig. 4: Facenet model structure used for feature extraction [9]

- n is the negative image (different person from the anchor).
- $d(x, y)$ is the distance between the embeddings of images x and y. Typically, this is the Euclidean distance, i.e., $d(x, y) = \|x - y\|$.
- α is a margin parameter that ensures the distance between the anchor and negative is sufficiently larger than the distance between the anchor and positive.

The total loss for the FaceNet model can be expressed as:

$$\mathcal{L}_{total} = \frac{1}{N} \sum_{i=1}^{N} \max\left(d(a_i, p_i) - d(a_i, n_i) + \alpha, 0\right) + \lambda \|W\|_2^2 \tag{2}$$

where:

- N is the number of triplets.
- $\lambda \|W\|_2^2$ is the regularization term, with λ as the regularization strength and $\|W\|_2^2$ representing the L2 norm of the model parameters W. It helps prevent overfitting by penalizing large weights, encouraging smoother and more generalized features.

Inception module working process:

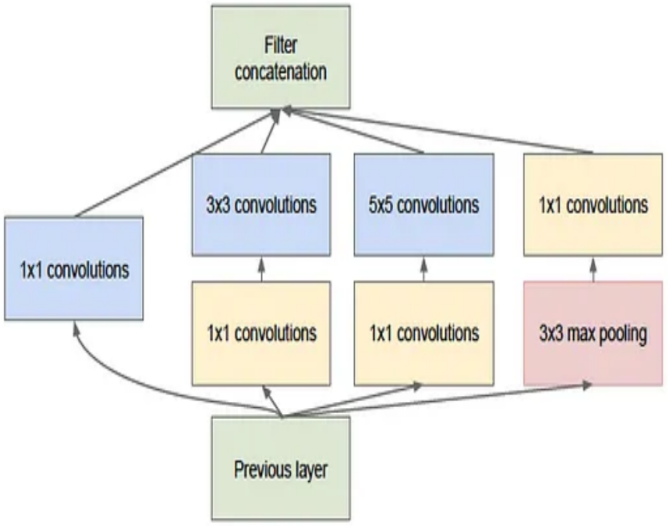


Fig. 5: Inception module structure [9]

The Inception module consists of multiple parallel convolutional paths with different filter sizes. This architecture allows the network to capture features at different scales efficiently. It also uses 1x1 convolutions for dimensionality reduction, enabling better performance with fewer parameters [9]. It also uses 3 x 3 max-pooling layers and then uses a 1 x 1 convolution layer after the max-pooling layer to reduce dimensionality. By using this module Facenet has reduced the dimension as well as the number of parameters which enabled the model to perform well and faster [9]. Thus, by using both skip connections and the Inception module, Facenet model has solved the vanishing gradient problem and also reduced the dimensionality and no of parameters, which finally led the model to go deeper without any degradation in performance [9].

3.3 Hardware and/or Software Implementation

The implementation of the Facial Recognition and Attendance System involves various components, including cameras, software libraries, processing units, networking infrastructure, and storage units. A well-known computer vision library called OpenCV is used to establish a connection with the camera and take pictures of people in real-time. The MTCNN (Multi-task Cascaded Convolutional Networks) technique is used to process the images efficiently because it has demonstrated strong performance in face detection and alignment tasks. The FaceNet model, an enhanced version of Convolutional Neural Networks (CNN) created by Google researchers, is used to extract facial information. FaceNet generates feature embeddings in high dimensions that capture distinct aspects of the face[9]. Algorithms such as Support Vector Machine (SVM), KNN (K-Nearest Neighbors), and others have been used for model training and classification to identify the individual based on the attributes that were collected. The fundamental model is trained and tested with tools like Visual Studio Code and Google Colab. For the image processing and facial recognition duties to be completed effectively, a suitable processing unit—a computer or server with enough computing power—is required. All system components must be connected via networking infrastructure to

facilitate data transfer and communication between the cameras, processing unit, and storage units. To store the collected photos, attendance records, face feature embeddings, and other pertinent data a storage unit is required. In addition to the advanced facial recognition system, we developed a comprehensive Django-based website to integrate and manage the attendance system seamlessly. The website is designed to provide a user-friendly and efficient experience with robust functionality and security. Key features of the website include user authentication for secure login and registration, a dynamic profile page with camera-based face extraction, and a management page where admins can add students, add courses, assign students to courses, view detailed lists of students and courses, and display attendance dates. The website also includes voice features for enhanced interactivity and real-time attendance tracking to reduce manual workload.

3.4 Front-end & Back-end

This Django application functions as a student attendance system with integrated connections to a broader student management system. It offers robust tools for tracking attendance, managing courses, and overseeing student enrollments, with all data stored in an SQLite3 database. The Course model handles the management of course sections, ensuring that each section has a unique combination of section number and name, with class timings validated and formatted in the 'HH AM/PM' format. Students are represented by the Student model, which is linked to their respective courses through the StudentCourse intermediary model, allowing for efficient enrollment tracking. The Attendance model records daily attendance for each student, providing accurate monitoring of class participation.

The application utilizes Django's built-in User model for secure authentication, differentiating user roles to control access. Although the system is designed to be database-agnostic and can easily be adapted to other databases, it currently uses SQLite3 for data storage. Access to the database is strictly controlled: only administrators have the authority to manage the database directly, ensuring data integrity and security. Teachers can interact with the system to manage their courses and view attendance data but do not have the privileges to alter the database, maintaining a clear separation of responsibilities and protecting against unauthorized modifications

3.5 Dataset

A dataset comprising 18 distinct classes was constructed by collecting images from 18 unique subjects. Then, Data augmentation techniques were applied to augment the dataset size. Subsequently, the dataset was partitioned into three mutually exclusive subsets: a training set ($n = 1080$), a validation set ($n = 540$), and a testing set ($n = 180$), constituting 60%, 30%, and 10% of the total dataset, respectively.

4. INVESTIGATION/EXPERIMENT, RESULT, ANALYSIS, AND DISCUSSION

4.1 Experiment Design

Facial detection and recognition experiments were conducted using the Multi-task Cascaded Convolutional Networks (MTCNN) for face detection and Keras FaceNet for face embedding. The dataset was divided into training, validation, and test sets. The faces were detected, extracted, and preprocessed for further analysis. For the CNN model, face detection and feature extraction were conducted by the CNN sequential Model. For EfficientNetB0, face detection, and feature extraction were conducted by EfficientNetB0 sequential model.

4.2 Results

The results of the experiments are visualized and summarized below for the CNN, SVM, KNN, EfficientNetB0, and RandomForest-based model training.

4.2.1 MTCNN+Facenet+SVM:

In the above image, A sample face was selected from the training dataset.

In the above image, Visualization of detected faces with key points. Blue rectangles indicate face boundaries and red points represent facial key points.

In the below image, a visualization of the detected face. Only the detected part was extracted from the image and then it was reshaped into 160*160 images before passing it to the feature extractor.

Face Embedding Extraction Using Facenet: Facial embeddings were extracted from the reshaped image using the Facenet model to pass it to the classifier for training. We have sent these facial embeddings to the SVM and KNN classifier.

We have tried to make a custom Facenet model. Where we have used globalaverpolling2D to reduce each feature map to a single value while effectively flattening the spatial dimensions and retaining the depth (features). We have used the ReLU activation function in the first dense layer and the softmax activation function in the last dense layer. This change gave us a slightly higher accuracy.

Fig. 6: Example of dataset

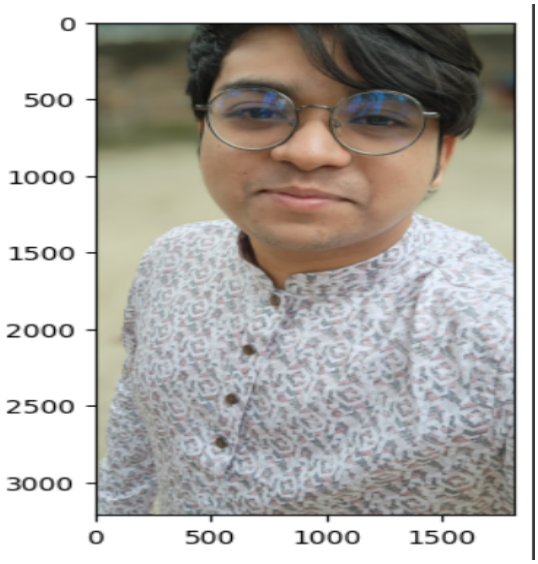


Fig. 7: Sample Face from the Training Set (1 Person)

SVM and KNN as classifier : Finally, we have used SVM and KNN for classification. We have fed these training feature embeddings to the SVM model for training. Then fed the validation and test embeddings to evaluate the performance of the model after training. We have tried using different regularization parameters and batch normalization for better performance. Finally, we got 100% accuracy in both train, validation, and test sets.

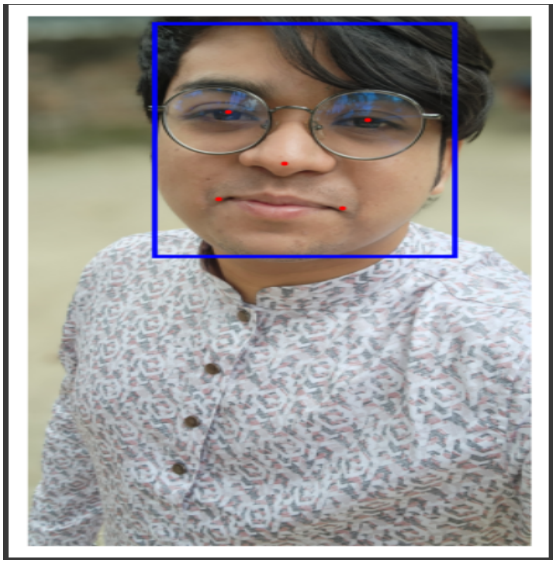


Fig. 8: Detected Faces with Keypoints Using MTCNN

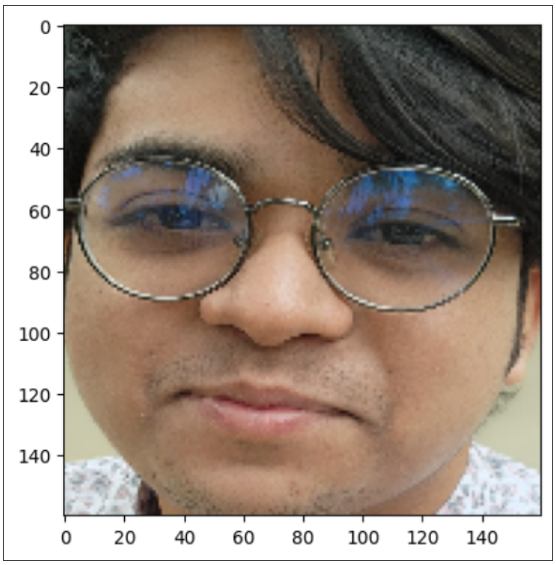


Fig. 9: Reshaped image of dimensions 160 x 160 x 3.

4.2.2 CNN-Sequential model: Firstly, we created a data frame from lists of file paths and labels. It visualized the images and labels in a grid format, scaled the data, and used class indices to annotate the images.
The batch size for the data generators is set to 32, meaning the data was processed in batches of 32 images. These generators preprocess the images (such as augmenting) and feed them into the model during training and evaluation.

A sequence model is developed. Three convolutional layers were added, each with 32, 64, or 128 filters, followed by a MaxPooling layer to minimize spatial dimensions. L2 regularization (with a factor of 0.001) was used to prevent overfitting. The output of the convolutional layers was flattened and sent via fully connected (Dense) layers with 128 units and L2 regularization. The final Dense layer had equal units as classes and utilized the softmax activation function for classification. We then trained the CNN model with early stopping to optimize training efficiency and performance.
The loaded image was turned into a NumPy array containing pixel values. The variety is enlarged along the first axis, resulting in a batch of one image necessary for model prediction. The preprocessed image was stacked as a batch and fed into the model for prediction. The model returned a prediction array. The predicted class is determined by locating the index of the greatest value in the prediction array.

Performance Evaluation:

```
f1_test = f1_score(Ytc, test_preds, average='weighted')

# Validation set
accuracy_val = accuracy_score(Yvalc, val_preds)
precision_val = precision_score(Yvalc, val_preds, average='weighted')
recall_val = recall_score(Yvalc, val_preds, average='weighted')
f1_val = f1_score(Yvalc, val_preds, average='weighted')

# Print the results
print(f'Training Accuracy: {accuracy_train:.2f}, Precision: {precision_trai
print(f'Test Accuracy: {accuracy_test:.2f}, Precision: {precision_test:.2f}
print(f'Validation Accuracy: {accuracy_val:.2f}, Precision: {precision_val:

Training Accuracy: 1.00, Precision: 1.00, Recall: 1.00, F1 Score: 1.00
Test Accuracy: 1.00, Precision: 1.00, Recall: 1.00, F1 Score: 1.00
Validation Accuracy: 1.00, Precision: 1.00, Recall: 1.00, F1 Score: 1.00
```

Fig. 10: Accuracy, Precision, Recall and F1-score of this model

Fig. 11: Creating Dataframes and Generating Data

```
Confusion Matrix, Without Normalization
[[ 2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  3  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  3  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  3  0  0  0  0  0  0  0  0  0  0  1  0  0  0]
 [ 0  1  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  1  0  3  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  1  0  0  0  0  0  3  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  3  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  4  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  1  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  3  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  1  0  0  0  0  3  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  3  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  3  0  0  0]
 [ 0  0  0  0  0  0  0  0  1  0  0  0  0  0  1  0  1  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 17  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  5]]
```

Fig. 12: Confusion matrix without normalization

4.2.3 EfficientNet B0: We first defined paths and created data frames to prepare data for image classification. For data generation with augmentation, we set the image size to (80, 80), choose the batch size for testing, and describe a scalar function

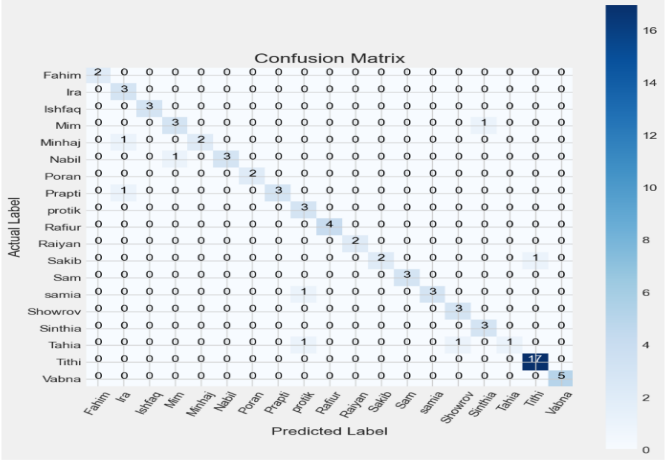


Fig. 13: Confusion matrix

	precision	recall	f1-score	support
Fahim	1.00	1.00	1.00	2
Ira	0.60	1.00	0.75	3
Ishfaq	1.00	1.00	1.00	3
Mim	0.75	0.75	0.75	4
Minhaj	1.00	0.67	0.80	3
Nabil	1.00	0.75	0.86	4
Poran	1.00	1.00	1.00	2
Prapti	1.00	0.75	0.86	4
protik	0.60	1.00	0.75	3
Rafiur	1.00	1.00	1.00	4
Raiyan	1.00	1.00	1.00	2
Sakib	1.00	0.67	0.80	3
Sam	1.00	1.00	1.00	3
samia	1.00	0.75	0.86	4
Showrov	0.75	1.00	0.86	3
Sinthia	0.75	1.00	0.86	3
Tahia	1.00	0.33	0.50	3
Tithi	0.94	1.00	0.97	17
Vabna	1.00	1.00	1.00	5
accuracy			0.89	75
macro avg	0.92	0.88	0.87	75
weighted avg	0.92	0.89	0.89	75

Fig. 14: Precision Recall F1-score

for preprocessing. ImageDataGenerator was used for data augmentation, with horizontal flipping applied to the training dataset. Data generators were created for training, validation, and testing, and then we input the data into the model in batches. We set a batch size of 32 for splitting images during training and validation. It then yields three generators: train_gen, valid_gen, and test_gen. These generators loaded and preprocessed images in batches in an efficient manner during model training and evaluation. After that, a batch of images from the training set was displayed, allowing for visual analysis of the data fed into the model. This phase was critical to ensure the images and labels were valid and the preprocessing steps were correctly followed.

First, we set the input shape to (80, 80, 3), which indicates 80x80 pixel images with three color channels (RGB). The class_count was defined by the number of classes in the training generator, as the final output layer must match the number of categories in the dataset.

The EfficientNetB0 architecture, a pre-trained model from the EfficientNet family noted for its efficiency and performance, serves as the model's core. This base model was loaded without the top classification layer, and the weights are pre-trained on the ImageNet dataset. To minimize the dimensionality of the base model's feature maps, we set the input shape to (80, 80,

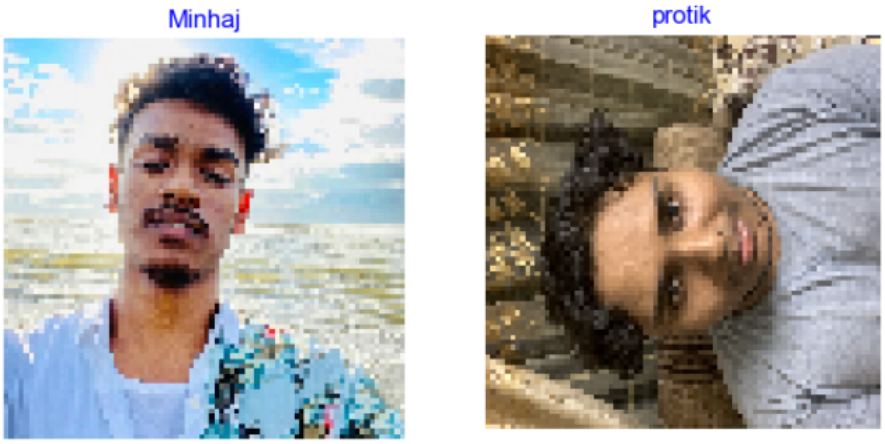


Fig. 15: Creating Dataframes and Generating Data

3) and pooling to max. The base model was then wrapped in a Sequential model, adding additional layers.

Batch Normalization: This layer normalized the previous layer's activations, which helped to accelerate training and achieve more stable convergence.

Dense Layer: A fully linked layer with 256 units that uses L2 and L1 regularization to avoid overfitting. The ReLU activation function was used to introduce nonlinearity.

Dropout Layer: Dropout was utilized at a rate of 0.45 to randomly discard 45% of the neurons during training to avoid overfitting. The seed guarantees consistency.

Output Dense Layer: This final layer has class_count units and a softmax activation function, making it appropriate for multi-class classification problems. The model is constructed using the Adamax optimizer, a version of the Adam optimizer that is less sensitive to significant gradient changes, with a learning rate of 0.001.

An early stopping callback was set up to monitor the validity loss. If the validation loss does not improve for five consecutive epochs (patience=5), training will be terminated early, and the best model weights will be restored. This helps to prevent overfitting and reduces wasteful computation. The loaded image was converted to a NumPy array. The dimensions of this array were expanded to include a batch dimension, making the shape (1, 80, 80, 3). This was necessary because the model expects a batch of images, even just one.
The preprocessed image was passed through the model's prediction function to get the prediction. The batch_size parameter was set to 32, though it processes only one image here. It then finds the predicted class by taking the class with the highest probability and maps it to the corresponding class name.

Performance Evaluation:

```
Confusion Matrix, Without Normalization
[[5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 5 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 5 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 1 0 0 4 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 5 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 5 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 4 1 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 5 0 0 0 0 0 0 0]
 [0 0 0 1 0 0 0 0 0 0 0 0 0 4 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 5 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 5 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 5 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 5 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 5 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 5]]
```

Fig. 16: Confusion matrix without normalization

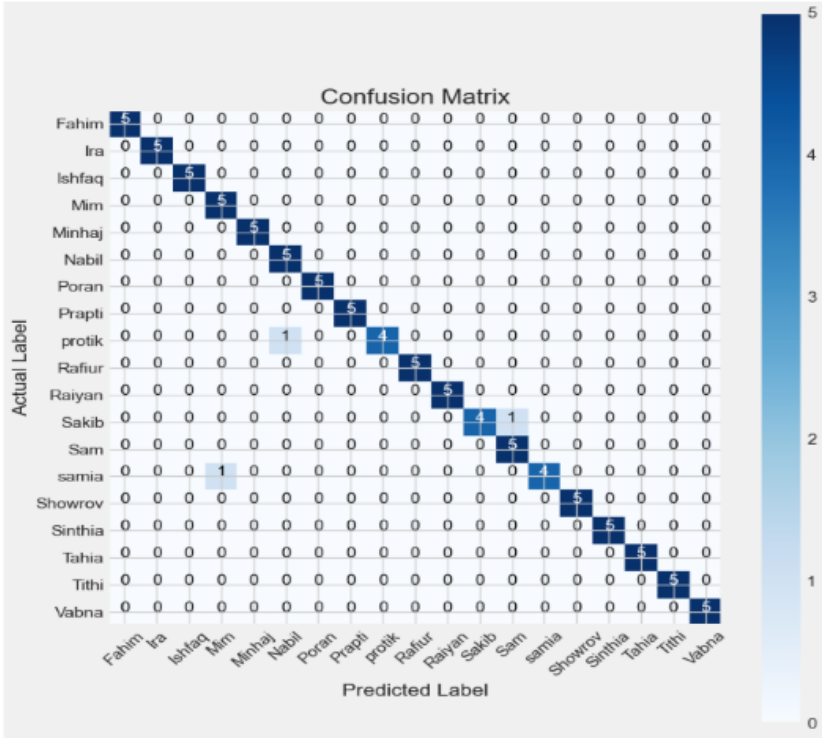


Fig. 17: Confusion matrix

	precision	recall	f1-score	support
Fahim	1.00	1.00	1.00	5
Ira	1.00	1.00	1.00	5
Ishfaq	1.00	1.00	1.00	5
Mim	0.83	1.00	0.91	5
Minhaj	1.00	1.00	1.00	5
Nabil	0.83	1.00	0.91	5
Poran	1.00	1.00	1.00	5
Prapti	1.00	1.00	1.00	5
protik	1.00	0.80	0.89	5
Rafiur	1.00	1.00	1.00	5
Raiyan	1.00	1.00	1.00	5
Sakib	1.00	0.80	0.89	5
Sam	0.83	1.00	0.91	5
samia	1.00	0.80	0.89	5
Showrov	1.00	1.00	1.00	5
Sinthia	1.00	1.00	1.00	5
Tahia	1.00	1.00	1.00	5
Tithi	1.00	1.00	1.00	5
Vabna	1.00	1.00	1.00	5
accuracy			0.97	95
macro avg	0.97	0.97	0.97	95
weighted avg	0.97	0.97	0.97	95

Fig. 18: Precision Recall F1-score

4.2.4 Website:

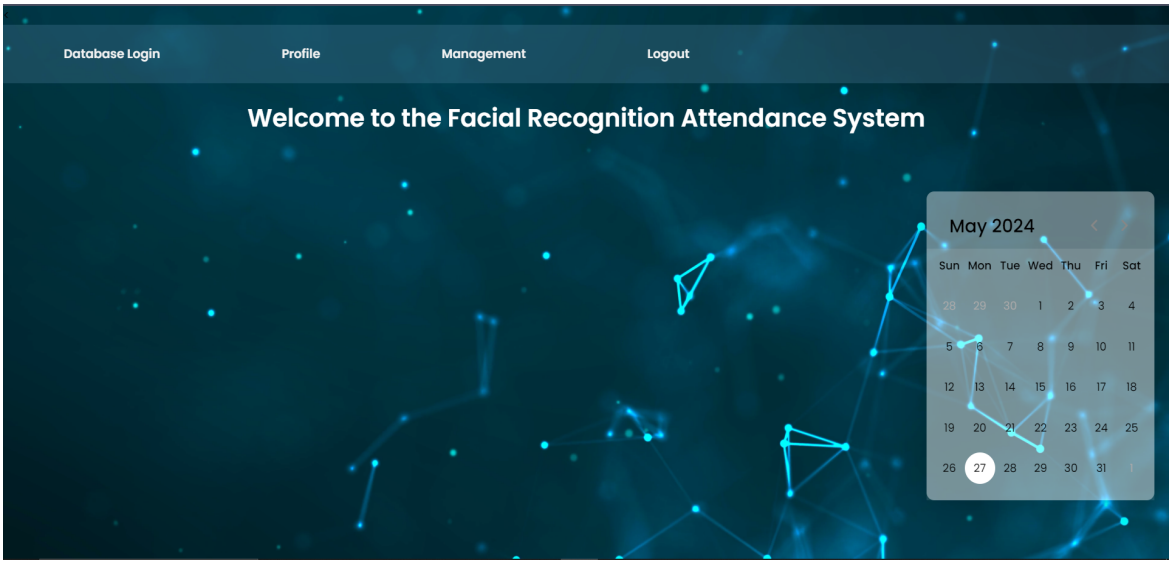


Fig. 19: Homepage of the website

In the above image, we can see the homepage. From here the admin can access the database and management page and the faculty members can access profiles where their sections are listed.

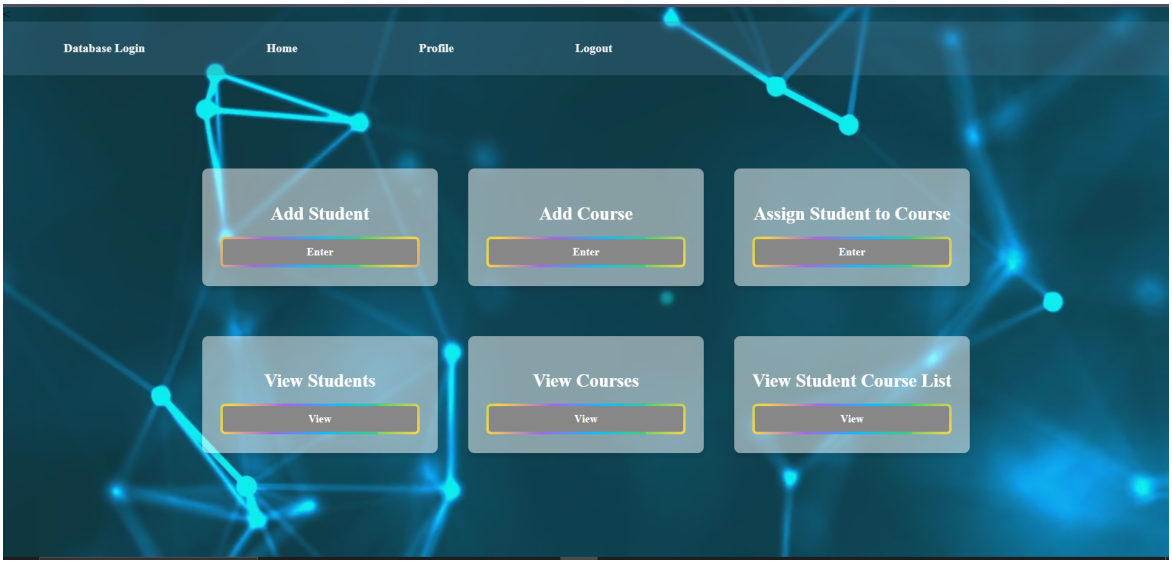


Fig. 20: Management of the website

This page is only accessible by the admin. The admin can add students, and courses and assign the student to different courses from this page.

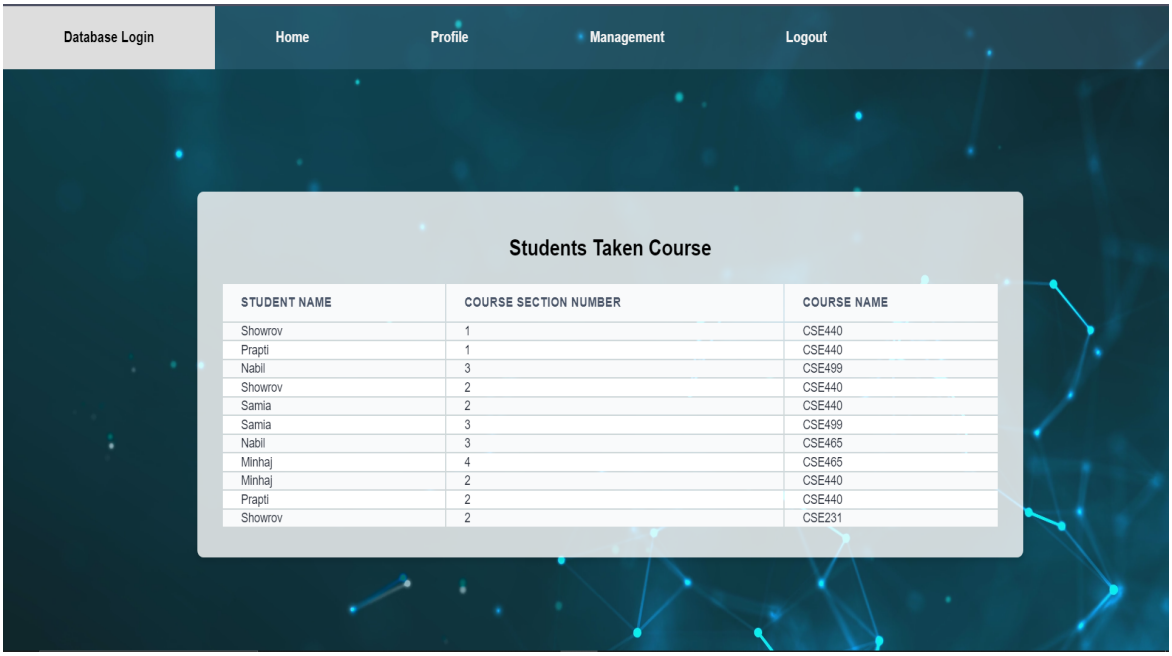


Fig. 21: List of the students enrolled in different sections

This page shows the student names, their courses, and the sections in which students are enrolled.

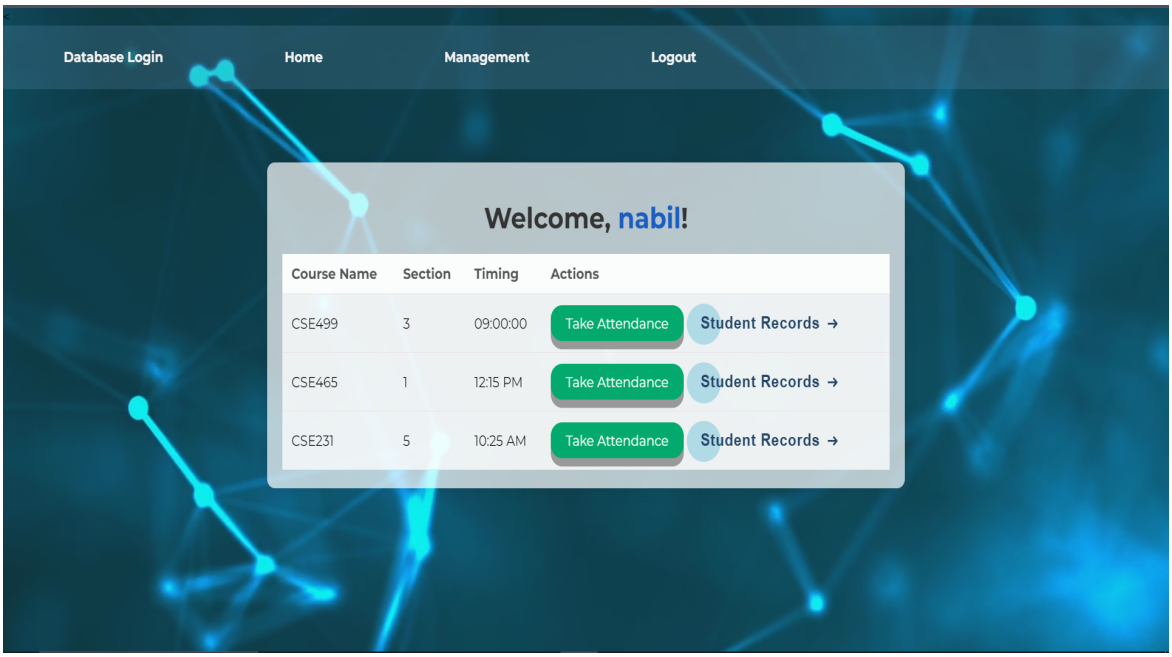


Fig. 22: Faculty profile view page

On this page, faculty members can see their courses and take attendance as well.

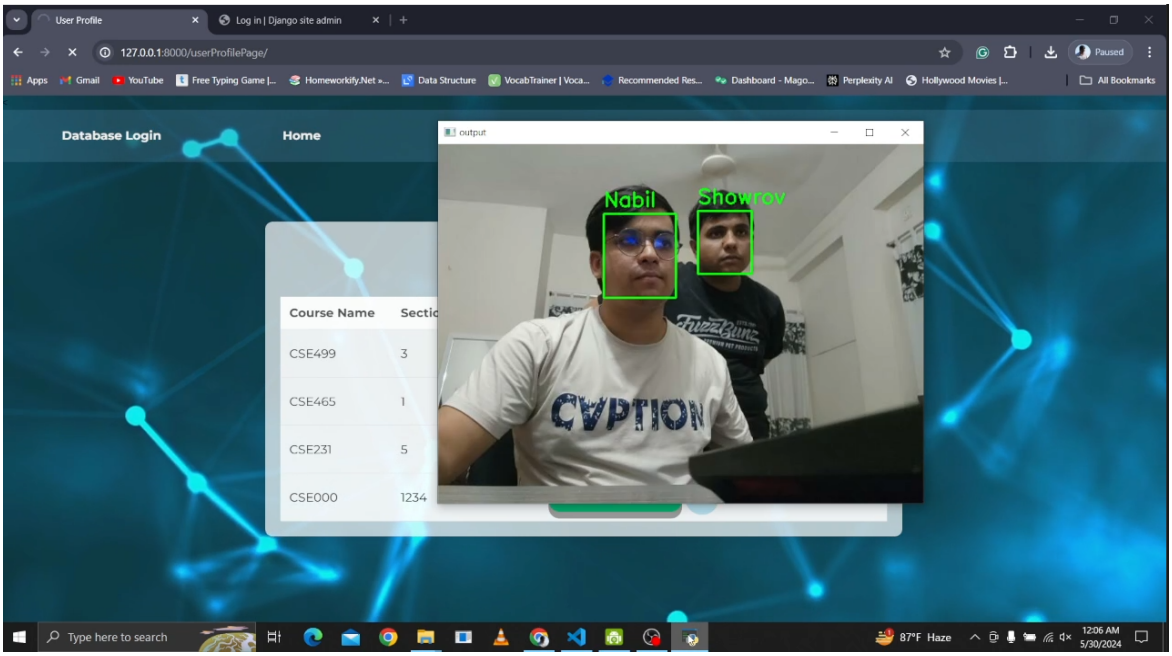


Fig. 23: View of taking multiple attendances at the same time

When the faculty member starts the attendance of a section it can continuously take attendance of the students. It can take multiple attendance at the same time. There are other functionalities where the admin can view the student list, and course list and can adjust them if they want. They can also assign faculty members to different sections and courses.

4.3 Analysis

4.3.1 Model Performance Metrics: :

Model Used	Train	Validation	Test
MTCNN, FaceNet, SVM	Accuracy: 100%. Precision: 100%. Recall : 100%. F1 Score : 100%.	Accuracy: 100% Precision: 100% Recall : 100% F1 Score : 100%	Accuracy: 100%. Precision: 100%. Recall : 100%. F1 Score : 100%.
MTCNN, FaceNet, KNN	Accuracy: 83%. Precision: 85%. Recall : 83%. F1 Score : 83%.	Accuracy : 73% Precision : 75% Recall : 73% F1 Score : 71%	Accuracy: 84%. Precision: 88%. Recall : 84%. F1 Score : 84%
CNN-Sequential Based	Accuracy : 100%	Accuracy: 78%	Accuracy: 89.3% Precision : 92% Recall : 88% F1 Score : 89%
EfficientNet B0	Accuracy: 100%	Accuracy : 94%	Accuracy : 97% Precision : 97% Recall : 97% F1 Score : 97%

Fig. 24: Model Performance Metrics

:

4.3.2 Model Performance Curves: **Performance of CNN-Sequential-based:** Representation of Training and validation loss curve and train and validation accuracy curve.

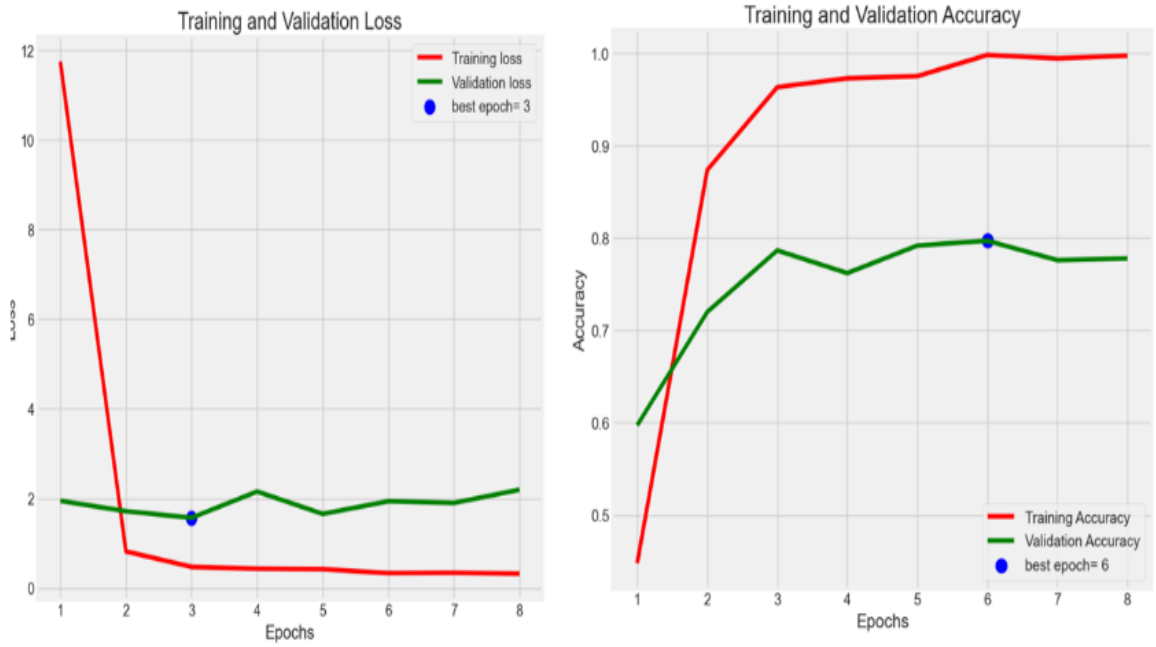


Fig. 25: Training and validation Error Curve (5a) and Accuracy Curve (5b)

Performance of EfficientNetB0: Representation of Training and validation loss curve and train and validation accuracy curve.

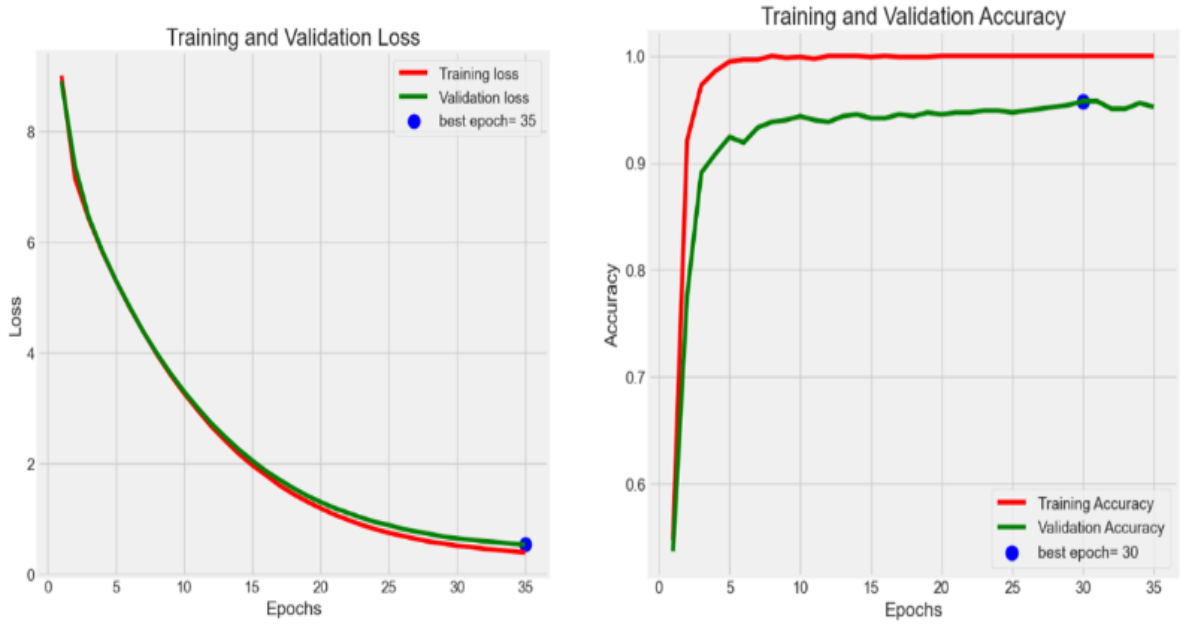


Fig. 26: Training and validation Loss Curve (6a) and accuracy curve (6b)

4.3.3 Model Evaluation: **Model Evaluation of MTCNN+Facenet+SVM:**
Training set size: 1080
Validation set size: 540
Testing set size: 180
Total Image was 1800 for this model.

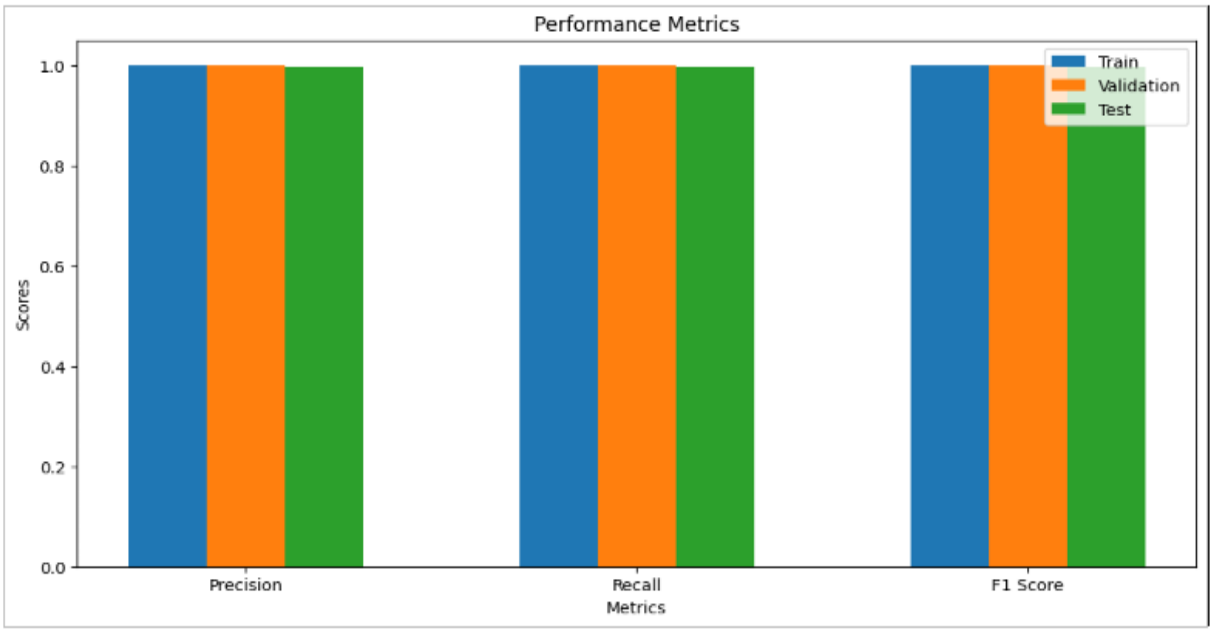


Fig. 27: Train, Validation and Test Performance Matrix (label were encoded to numerical Number)

The model demonstrated excellent generalization from training to the test set, with precision, recall, and F1 score of 1.0 across all sets. On the test set, it maintained high performance with precision of approximately 0.997, recall of 0.996, and F1 score of about 0.996.

Confusion matrix for MTCNN+facenet+SVM:

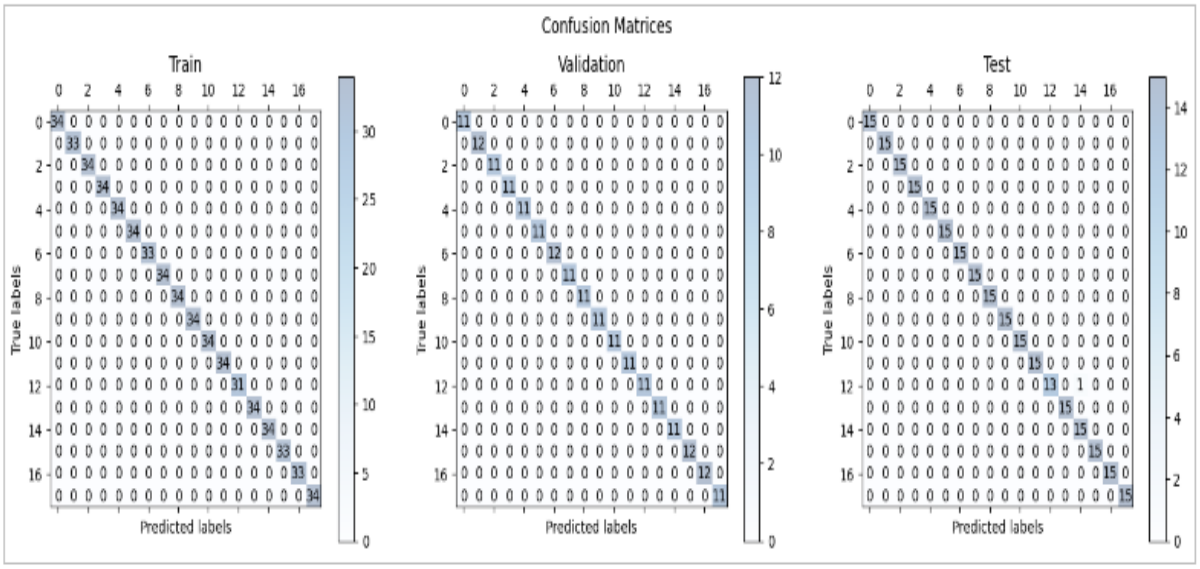
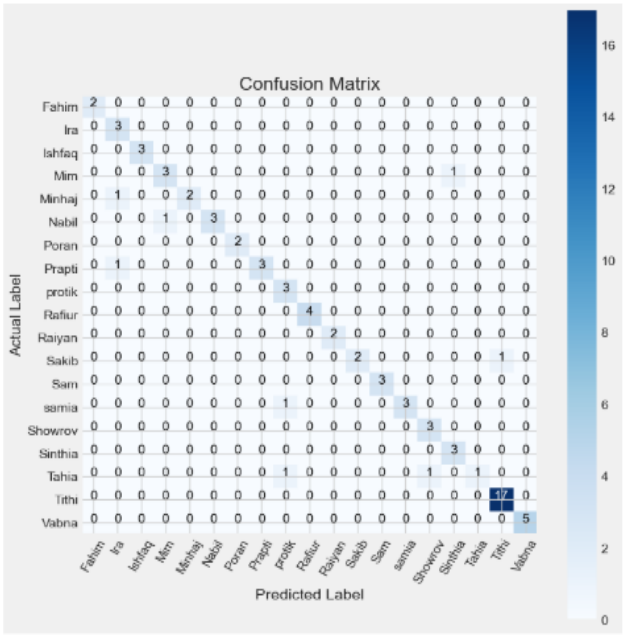


Fig. 28: Train, Validation and Test Confusion Matrix (label were encoded to numerical Number)

The training and validation matrices show high accuracy, with most classes correctly classified, while the test matrix confirms the model's robustness, with one misclassified instance.

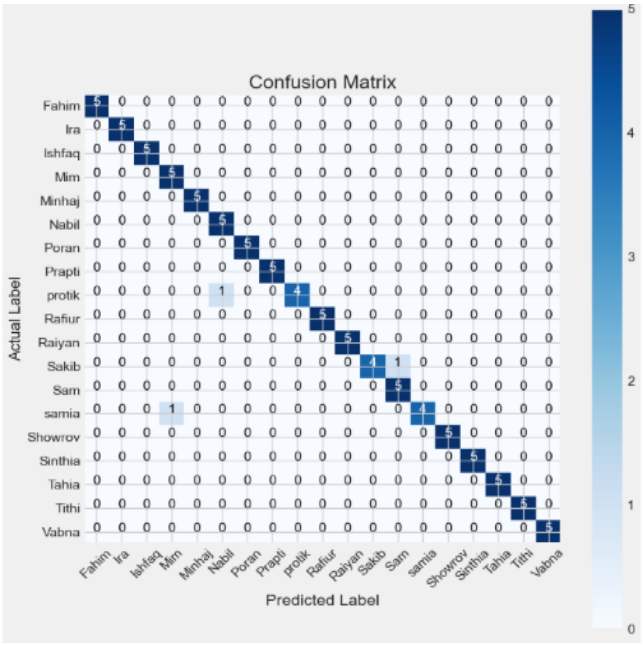
Model Evaluation of CNN-sequential-based:



	precision	recall	f1-score	support
Fahim	1.00	1.00	1.00	2
Ira	0.60	1.00	0.75	3
Ishfaq	1.00	1.00	1.00	3
Mim	0.75	0.75	0.75	4
Minhaj	1.00	0.67	0.80	3
Nabil	1.00	0.75	0.86	4
Poran	1.00	1.00	1.00	2
Prapti	1.00	0.75	0.86	4
protik	0.60	1.00	0.75	3
Rafiur	1.00	1.00	1.00	4
Raiyan	1.00	1.00	1.00	2
Sakib	1.00	0.67	0.80	3
Sam	1.00	1.00	1.00	3
samia	1.00	0.75	0.86	4
Showrov	0.75	1.00	0.86	3
Sinthia	0.75	1.00	0.86	3
Tahia	1.00	0.33	0.50	3
Tithi	0.94	1.00	0.97	17
Vabna	1.00	1.00	1.00	5
accuracy			0.89	75
macro avg	0.92	0.88	0.87	75
weighted avg	0.92	0.89	0.89	75

Fig. 29: Confusion Matrix (7a) and evaluation table (7b)

Model Evaluation of EfficientNetB0:



	precision	recall	f1-score	support
Fahim	1.00	1.00	1.00	5
Ira	1.00	1.00	1.00	5
Ishfaq	1.00	1.00	1.00	5
Mim	0.83	1.00	0.91	5
Minhaj	1.00	1.00	1.00	5
Nabil	0.83	1.00	0.91	5
Poran	1.00	1.00	1.00	5
Prapti	1.00	1.00	1.00	5
protik	1.00	0.80	0.89	5
Rafiur	1.00	1.00	1.00	5
Raiyan	1.00	1.00	1.00	5
Sakib	1.00	0.80	0.89	5
Sam	0.83	1.00	0.91	5
samia	1.00	0.80	0.89	5
Showrov	1.00	1.00	1.00	5
Sinthia	1.00	1.00	1.00	5
Tahia	1.00	1.00	1.00	5
Tithi	1.00	1.00	1.00	5
Vabna	1.00	1.00	1.00	5
accuracy			0.97	95
macro avg	0.97	0.97	0.97	95
weighted avg	0.97	0.97	0.97	95

Fig. 30: Confusion Matrix (8a) and evaluation table (8b)

5. CONCLUSION

5.1 Summary

A variety of machine learning and deep learning techniques, such as MTCNN+FaceNet+SVM, MTCNN+FaceNet+KNN, VGG16, CNN-Sequential Based, and EfficientNetB0, are used in this project's facial recognition attendance system. With training accuracy at 100%, validation accuracy at 100%, and testing accuracy at 100%, MTCNN+Facenet+SVM produced the best results. The system is a highly dependable solution because of its higher accuracy, attained through complex face feature analysis. This system contains three major steps that include (1) collection of facial images, (2) training various models on a train and a validation dataset containing pictures, (3) Results comparison for metrics like recall, accuracy, precision, F1-score, and precision. It can successfully recognize 18 different person faces with MTCNN+Facenet+SVM outperforming other models. So, by offering a productive, straightforward replacement for conventional techniques, this technology completely transforms attendance tracking. It makes use of cutting-edge algorithms and artificial intelligence to generate personalized face maps for each user, assuring precision and minimizing the need for manual interaction.
Django-based website featuring user authentication, dynamic profile pages with camera-based face extraction, management capabilities for students and courses, voice features, real-time attendance tracking, and calendar integration. Utilizing Sqlite3 for the back end and HTML, CSS, and JavaScript for the front end, our platform ensures robust performance, security, and a user-friendly experience, making it a complete solution for automated attendance management in educational institutions and workplaces.

5.2 Limitations

Our system performed very well with the MTCNN+Facenet+SVM model, with training accuracy at 100%, validation accuracy at 100%, and testing accuracy at 100%. All the other models also performed well, but with the VGG16 and Random Forest model, there was some issue with training on those datasets, because of which the test accuracy did not come out as decent. Privacy concerns are a major issue, as capturing and storing image data can be sensitive and subject to misuse if not properly secured. The accuracy of facial recognition can be affected by lighting and environmental conditions, such as low light, shadows, or background clutter. Occlusions like masks or sunglasses and changes in appearance such as facial hair or aging can also pose challenges. There are security risks associated with potential data breaches or hacking attempts, compromising data and user privacy. Ethical and legal issues also arise, as the deployment of this technology must navigate varying regulations and societal norms regarding image data usage. User acceptance can be hindered by concerns about privacy and security.

5.3 Future Improvement

This project will work on more confidentiality as this project includes working with pictures of people and will improve the user interface to enhance the system's adoption. The building software will keep updating the system with new facial recognition models and training datasets, which will improve its performance over time, keeping it aligned with evolving technological advancements to make it convenient for various organizations besides classroom attendance methods. Also by incorporating artificial intelligence, systems may be able to adjust and grow, leading to constant improvements in identification precision. So, the system will be scaled for larger operations with additional layers of authentication to avoid fraud.

REFERENCES

[1] Kapse, A. S., Kamble, T., Lohar, A., Chaudhari, S., Puri, D. (2022). Face recognition Attendance system using HOG and CNN algorithm. *ITM Web of Conferences*. doi:10.1051/itmconf/20224403028

[2] Zamir, M., Ali, N., Naseem, A., Ahmed Frasteen, A., Zafar, B., Assam, M., Othman, M., Attia, E.-A. (2022). Face Detection Recognition from Images Videos Based on CNN Raspberry Pi. *Computation*, 10(9), 148. doi: https://doi.org/10.3390/computation10090148.

[3] Nurkhamid, S., Setialana, P., Jati, H., Wardani, R., Indrihapsari, Y., Norwawi, N. M. (2021). Intelligent attendance system with face recognition using the deep convolutional neural network method. *Journal of Physics: Conference Series*, 1737(1), 012031. doi:10.1088/1742-6596/1737/1/012031

[4] Xu, W., Li, B., Du, Y., Dong, S. (2023). Study on facial recognition method based on YOLOV5. *Journal of Physics: Conference Series*, 2560(1), 012020. doi: 10.1088/1742-6596/2560/1/012020.

[5] Sanchez-Moreno, A. S., Olivares-Mercado, J., Hernandez-Suarez, A., Toscano-Medina, K., Sanchez-Perez, G., Benitez-Garcia, G. (2021). Efficient face recognition system for operating in unconstrained environments. *Journal of Imaging*, 7(9), 161. doi: 10.3390/jimaging7090161.

[6] Nandhini, R., Duraimurugan, N., Chokkalingam, S. P. (2019). Face recognition-based attendance system. *International Journal of Engineering and Advanced Technology (IJEAT)*, 8(3S), 616-620.

[7] Dar, S. A., Palanivel, S. (2021). Performance evaluation of convolutional neural networks (CNNs) and VGG on real time face recognition system. *Advances in Science, Technology and Engineering Systems Journal*, 6(2), 956-964. doi: 10.25046/aj0602109.

[8] Hammadi, O. I., Abas, A. D., Ayed, K. H., Abbas, A. D. (2018). Face recognition using deep learning methods: A review. *ResearchGate*. doi:10.14419/ijet.v7i4.22375

[9] Deore, M. (2019, April 4). FaceNet Architecture - Part 1: Architecture and running a basic example on Google Colab. Analytics Vidhya.

[10] FaceNet. (n.d.). Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/FaceNet.

[11] Bangar, S. (2019). ResNet Architecture Explained. Medium. [Online]. Available: https://siddheshbangar.medium.com/resnet-architecture-explained-934a6c5e1e3b.

[12] Rajput, S. (2020, September 1). Face Detection using MTCNN. Medium. [Online]. Available: https://medium.com/@saranshrajput/face-detection-using-mtcnn-f3948e5d1acb.

[13] Jazouli, M., Majda, A., Zarghili, A. (2017). A P recognizer for automatic facial emotion recognition using Kinect sensor. In 2017 Intelligent Systems and Computer Vision (ISCV). doi:10.1109/ISACV.2017.8054955

[14] Abdullah, M., Wazzan, M., Bo-Saeed, S. (2012). Optimizing face recognition using PCA. *International Journal of Artificial Intelligence and Applications*, 3(2), 33-44. doi:10.5121/ijaia.2012.3203

[15] Teoh, K. H., Ismail, R. C., Naziri, S. Z. M., Hussin, R., Isa, M. N. M., Basir, M. S. S. M. (2021). Face recognition and identification using deep learning approach. *Journal of Physics: Conference Series*, 1755(1), 012006. doi:10.1088/1742-6596/1755/1/012006

[16] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.

[17] 012006. [Online]. Available: https://doi.org/[DOI]

[18] Juneja, M., Moghe,S.(2023).Review paper on face recognition attendance system.https://doi.org/10.13140/RG.2.2.21028.24962

[19] Singh, S., Hazel, B., Asthana, P.(2023).Face Recognition Attendance System.https://doi.org/10.15680/IJIRCCE.2023.1105209

[20] Moz, S.H., Hosen. M.A., Khalid, M.M.H., Kabir, S.S.(2023).Face recognition-based attendance system with anti-spoofing, system alert, and email automation.https://doi.org/10.32620/reks.2023.2.10

[21] Kumar, H., Bhati, N., Bharadwaj, P., Chaudhary, P. (2023). Real-time face attendance system using face recognition.

[22] Sanchez-Moreno, A. S., Olivares Mercado, J., Hernandez-Suarez, A., Toscano, K. (2021). Efficient face recognition system for operating in unconstrained environments. Journal of Imaging, 7(9), Article 161. https://doi.org/10.3390/jimaging7090161

[23] Joshi, D., Patil, P., Singh, V., Vanjari, A., Shinde, T., Giri, H. (2023). Face recognition-based attendance system. In 2023 5th Biennial International Conference on Nascent Technologies in Engineering (ICNTE). IEEE. https://doi.org/10.1109/ICNTE56631.2023.10146718

[24] Aasthasingh. (2022, June 9). Face Emotion Recognition EfficientNet-B0 PyTorch. Kaggle. https://www.kaggle.com/code/aasthasingh21/face-emotion-recognition-efficientnet-b0-pytorch

[25] Liu, M., Cai, R., Li, L., Wang, J., Yang, Q. (2022). An efficient multiscale pyramid attention network for face detection in surveillance images. Security and Communication Networks, 2022, 1–11. https://doi.org/10.1155/2022/8080301

[26] Dubey, A. K., Jain, V. (2019). A review of face recognition methods using deep learning network. Journal of Information and Optimization Sciences, 40(2), 547–558. https://doi.org/10.1080/02522667.2019.1582875

[27] Design of automated smart attendance system using deep learning based face recognition. (2024, April 4). IEEE Conference Publication — IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/10568825

[28] Ali, M., Diwan, A., Kumar, D. (2024). Attendance System Optimization through Deep Learning Face Recognition. International Journal of Computing and Digital Systems, 15(1), 1527–1540. https://doi.org/10.12785/ijcds/1501108

[29] Al-Amoudi, I., Samad, R., Abdullah, N. R. H., Mustafa, M., Pebrianti, D. (2021). Automatic Attendance System Using Face Recognition with Deep Learning Algorithm. In Lecture notes in electrical engineering (pp. 573–588). https://doi.org/10.1007/978-981-16-2406-344

[30] Gomes, C., Chanchal, S., Desai, T., Jadhav, D. (2020). Class Attendance Management System using Facial Recognition. ITM Web of Conferences, 32, 02001. https://doi.org/10.1051/itmconf/20203202001

[31] Sanli, O., Ilgen, B. (2018). Face detection and recognition for automatic attendance system. In Advances in intelligent systems and computing (pp. 237–245). https://doi.org/10.1007/978-3-030-01054-6=17

[32] Smart Multiple Attendance System through Single Image. (2020, November 5). IEEE Conference Publication — IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/9318103

[33] Ali, M., Diwan, A., Kumar, D. (2024b). Attendance System Optimization through Deep Learning Face Recognition. International Journal of Computing and Digital Systems, 15(1), 1527–1540. https://doi.org/10.12785/ijcds/1501108

[34] Rastogi, R., Tyagi, A., Upadhyay, H., Singh, D. (2022). Algorithmic analysis of automatic attendance system using facial recognition. International Journal of Decision Support System Technology, 14(1), 1–19. https://doi.org/10.4018/ijdsst.286688

[35] Hong, Y. Z., Nordin, N. (2023, January 24). Web-based Attendance System with Face Recognition. https://penerbit.uthm.edu.my/periodicals/index.php/mari/article/view/9900

[36] Varsha, M., Nair, S. C. (2021). Automatic attendance Management system using face detection and face recognition. In Lecture notes in networks and systems (pp. 97–106). https://doi.org/10.1007/978-981-16-2919-8-9

[37] Sanivarapu, P. V. (2020). Multi-Face recognition using CNN for attendance system. In Lecture notes in networks and systems (pp. 313–320). https://doi.org/10.1007/978-981-15-7106-0-31

[38] Kala, V. S. S., Bhavyasri, V., Vigneshwari, S. (2021). Face recognition based attendance system and emotion classification. In Lecture notes in electrical engineering (pp. 625–634). https://doi.org/10.1007/978-981-15-7511-2-63

[39] K, S. H., Upreti, A. S., Shakya, S. N., Dadapeer, S., Panjiyar, P. (2022). Attendance monitoring system using face recognition. Zenodo (CERN European Organization for Nuclear Research). https://doi.org/10.5281/zenodo.7385439

[40] Vamsikrishna, J., Anudeep, K., Marcilin, L. J. A., Balamurugan, V. (2019). An advanced attendance marking system using facial recognition. IOP Conference Series Materials Science and Engineering, 590(1), 012049. https://doi.org/10.1088/1757-899x/590/1/012049