International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 15, No.1 (Apr-24)

http://dx.doi.org/10.12785/ijcds/1501108

Attendance System Optimization through Deep Learning Face Recognition

Mahmoud Ali[1], Anjali Diwan[1] and Dinesh Kumar[2]

[1]*Department of Computer Engineering AI and Big Data, Marwadi University, Rajkot, India*
[2]*School of Computer Science Engineering and Technology, Bennett University, Greater Noida, India*

Received 18 Jan. 2024, Revised 14 Mar. 2024, Accepted 18 Mar. 2024, Published 1 Apr. 2024

Abstract: The significance of face recognition technology spans across diverse domains due to its practical applications. This study introduces an innovative face recognition system that seamlessly integrates Multi-task Cascaded Convolutional Neural Networks (MTCNN) for precise face detection, VGGFace for feature extraction, and Support Vector Machine (SVM) for efficient classification. The system demonstrates exceptional real-time performance in tracking multiple faces within a single frame, particularly excelling in attendance monitoring. Notably, the "VGGFace" model emerges as a standout performer, showcasing remarkable accuracy and achieving an impressive F-score of 95% when coupled with SVM. This underscores the model's effectiveness in recognizing facial identities, attributing its success to robust training on extensive datasets. The research underscores the potency of the VGGFace model, especially in collaboration with various classifiers, with SVM yielding notably high accuracy rates.

Keywords: Attendance, Face Detection, Face Recognition, Feature Extraction, Transfer Learning

1. INTRODUCTION

Face recognition technology has piqued interest in various sectors due to its practical applications. The process begins with face detection, followed by recognition once the face is located. Extracting facial features is crucial for robust recognition, ensuring adaptability to environmental changes. This step is essential for enhancing recognition results [1].

Traditional attendance tracking methods like paper-based registers or barcode scanners are inefficient and prone to errors. The shift to online learning due to the Covid-19 pandemic has made tracking attendance a time-consuming task in both offline and online settings. The need for more efficient systems, especially in education, workplaces, and events, has led to the adoption of face recognition technology [2].

Face recognition-based attendance systems have gained attention for their accuracy and speed in identifying individuals. These systems use advanced computer vision and deep learning to detect faces in images or live videos. While single-face recognition systems are common, there's a rising demand for solutions handling multiple face detection and recognition scenarios.

Identifying and authenticating multiple faces simultaneously holds importance in various practical applications. For instance, in education, efficiently recording attendance for all students at once is crucial. Likewise, in workplaces or events, monitoring attendance for many individuals is essential for security and organization.

To meet these demands, a sophisticated face recognition-based attendance system is proposed. It can detect and recognize multiple faces in a single frame by utilizing technologies like Multi-Task Cascaded Convolutional Neural Network (MTCNN) for face detection, VGGFace for feature extraction, and Support Vector Machine (SVM) for precise identification. This system aims to streamline attendance tracking, enhance security, reduce manual workload, and provide real-time attendance data. Its design emphasizes scalability, adaptability, and user-friendliness for deployment in diverse applications such as education, corporations, and large events.

2. RELATED WORK

The domain of facial recognition has experienced notable progress, especially due to the effective utilization of deep learning techniques in the past few years. Researchers have delved into different deep learning structures to extract features from faces and classify them, frequently utilizing transfer learning methods to enhance their effectiveness. The subsequent paragraph will elaborate on a few of these methodologies.

E-mail address: mahmoud.ali106127@marwadiuniversity.ac.in, anjali.diwan@ieee.org, dineshkr@ieee.org

Prior research studies have shown that face recognition tasks benefit from the high accuracy achieved by deep learning models. Additionally, the choice of classifier significantly influences the overall system performance [3].

Kumar et al. [4] introduces a multimodal biometric system merging facial and gait recognition. They employed PCA alongside an S-DNN, integrating cross-entropy analysis into their approach.

Arora et al. [5] introduce the MVM-LBP descriptor as a solution to address the constraints within face recognition systems. Their proposition implies that inclusion of variance within the descriptor can effectively accomplish this goal.

Chinimilli et al. [6] concentrate on improving face recognition-based attendance accuracy by reducing false positives with the aid of a confidence threshold, specifically employing the Local Binary Pattern Histogram (LBPH) algorithm. They found that the face recognition rate for identified students reached 77%, while the false-positive rate was reported at 28%.

Kocacinar et al. [7] developed a mobile system in real-time for recognizing masked faces using deep learning models, achieving a 90.40% validation accuracy. This system identifies full faces as well as just the eyes.

Maharani et al. [8] enhanced face mask recognition by merging Haar-cascade and MobileNet for detection, and incorporating VGG16 and Triplet Loss FaceNet for recognition with an accuracy rate of 82.20%. Additionally, they utilized cosine distance and FIFO techniques to produce person names and ID numbers, even under mask-wearing conditions.

Razaq and Shukur [9] devised a facial feature extraction method encompassing four phases. Their approach involved implementing a Faster Region Convolutional Neural Network to identify landmarks. Additionally, they utilized wavelet scatter and CNN techniques for feature extraction, employing DNN and SVM classifiers for subsequent classification purposes.

Ali and Kumar [10] suggested a rapid and precise method for face recognition, merging deep learning and machine learning. Their approach involved utilizing Inception V3 to extract features and employing Logistic Regression with Ridge regularization for classification. They obtained an accuracy of 94.87%.

Geetha et al. [11] focus on improving the speed and suitability of monitoring online exams by employing SVM and Eigenface algorithms, achieving an accuracy rate of 61%.

Ali et al. [12] present a real-time hidden Markov model for face recognition that minimizes computational complexity using preprocessing techniques.

Dang [13] introduces a facial recognition approach employing an enhanced FaceNet model built upon MobileNetV2 with SSD, achieving 95% accuracy on a small dataset. This method is well-suited for low-capacity hardware and exhibits promise in applications like smart attendance systems and IoT security.

Shibu and Samuel [14] present a new clustering algorithm named Celestial PSO (RCPSO), which relies on centroids and enhances the accuracy of clustering vast facial image datasets. This method draws inspiration from particle physics to tackle the complexities of clustering large, high-dimensional datasets. They achieved 93.20% accuracy with 5000 images and 85.6% accuracy with 10000 images on the LFW dataset.

Rajyalakshmi and Lakshmanna [15] propose an intelligent face recognition system aimed at expediting parking searches for drivers, lessening traffic congestion, and enhancing vehicle security, achieving a 91% face verification accuracy for 100 vehicles maintained.

Deng et al. [16] introduced ArcFace, a novel loss function designed for face recognition. This function projects face features onto a hypersphere, aiming to enhance distinguishability.

Yuan et al. [17] propose a lightweight federated learning model designed for both edge servers and mobile terminals, achieving 90% accuracy in experimental results. Their work aims to address the challenges of reduced model accuracy and equipment fairness in edge federated learning.

Yang and Han [18] developed a real-time attendance system based on video for face recognition, achieving an accuracy rate of up to 82%. The system focusses on four key aspects: enhancing check-in accuracy, ensuring system reliability, and minimizing truancy.

Ahmed et al. [19] conducted a comparison among four face recognition models using a dataset that featured only ten celebrities. VGG19 achieved an accuracy of 56%, while MobileNet demonstrated better performance with an accuracy of 84%, showcasing remarkable resilience to data variations.

Khan et al. [20] proposed a CNN-based approach for monitoring student attendance in smart classrooms. Their system incorporates IoT and edge computing to process data efficiently, outperforming conventional methods in detecting and recognizing faces. They obtained a notable performance with 94.6% accuracy in face detection and 85.5% accuracy in face recognition.

Potdar et al. [21] designed an attendance system for educators addresses proxy attendance by utilizing a liveliness detection algorithm. This algorithm distinguishes between real and spoof images through blinking detection and CNN classification. Additionally, the system employs FaceNet for face recognition, achieving an accuracy rate of 93%.

Autade et al. [22] have developed an automatic attendance tracking system based on facial recognition. It utilizes a CNN for real-time face recognition, leveraging the FaceNet model for face embedding and matching. The system achieves an 75% accuracy in attendance recording.

In the following sections, we'll explore the intricate components and operational principles of face recognition-based attendance system. A special focus will be given to its ability to effectively detect and recognize multiple faces simultaneously. Additionally, we'll delve into the technical details involving data collection, extracting features, and recording attendance in the cloud. This exploration aims to offer a comprehensive understanding of how this system can transform attendance management across various contexts. Further sections are arranged as follows: Section 3 presents an overview of the Attendance System. In Section 4, the Methodology employed in our research is explained. Section 5 presents our Proposed Approach in depth, providing a comprehensive explanation of the chosen approach and its rationale. To enhance clarity, Section 6 includes an Algorithm Diagram to visually represent the key processes. Moving forward, Section 7 discusses the effectiveness of our approach using specific evaluation metrics. In Section 8, the Discussion of Results critically analyzes the findings and their implications. Finally, Section 9 offers a Conclusion that summarizes the study's key contributions and hints at potential avenues for future research. This structured approach ensures a clear and organized presentation of our research methodology, findings, and insights.

3. ATTENDANCE SYSTEM

Attendance systems employing face recognition rely on advanced computer vision algorithms and artificial intelligence to automatically verify individuals by assessing facial features. Widely adopted across education, corporate offices, and government institutions, these systems streamline attendance tracking due to their convenience and effectiveness [23].

The attendance system using face recognition follows a sequence of stages to capture, process, and compare facial features for attendance purposes. The Figure 1 below illustrates the basic pipeline.

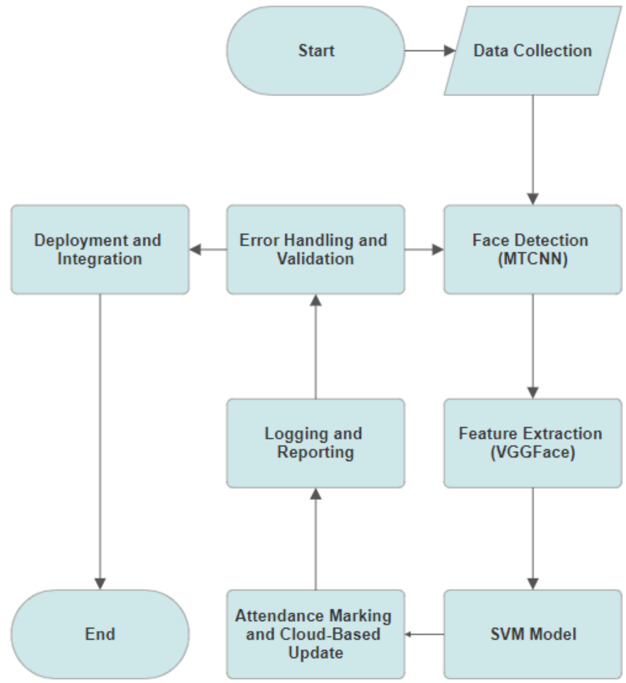

Figure 1. Working pipeline of face recognition-based attendance system

4. METHODOLOGY

We followed the following methodology to implement a face recognition attendance system:

A. Face Detection

Detecting faces holds paramount importance in face recognition systems, as it identifies and precisely localizes the existence of faces within images or videos. This process serves as a cornerstone for numerous other face-related tasks and various applications in the computer vision.

Various approaches exist for face detection, with one common method revolving around machine learning algorithms used to train models. During training, a dataset containing images with both facial and non-facial elements is typically used. As a result, the trained model becomes adept at recognizing facial features like eyes, nose, and mouth, enabling it to identify faces in new images [24]. Alternatively, another method involves template matching, where an image is compared to a predefined face template. If the image aligns with the template, the algorithm concludes the presence of a face within the image [25].

MTCNN utilizes multiple CNNs for face detection in a similar manner as [26], [27], employing three sequential convolutional neural networks. It conducts face detection, identifies facial landmarks, and aligns faces. We used the MTCNN algorithm to detect faces in input images before applying feature extraction methods and classifiers. Integrating MTCNN improved face detection accuracy and localization, significantly enhancing the performance of subsequent feature extraction methods and classifiers. Implementing MTCNN as a preprocessing step successfully eliminated false positives, ensuring the use of only authentic faces in the following stages.

MTCNN, a widely used technology in various fields such as security, human-computer interaction, and entertainment, is recognized for its accuracy in identifying facial features in images and videos, making it crucial in facial recognition systems [28]. Within this research, the utilization of MTCNN not only significantly improved the efficiency of feature extraction methods and classification algorithms but also emphasized the critical role of precise facial detection in achieving high accuracy in facial recognition procedures.

The utilization of MTCNN in the research showcased its capability to bolster the system's overall accuracy and dependability, highlighting its significance as a dependable face detection algorithm.

B. Feature Extraction

Feature extraction is a crucial process that derives significant features from raw data, enabling machine learning models to efficiently acquire knowledge from the data. This operation decreases the data volume the model deals with, facilitating the learning of relationships within the data. Convolutional neural networks (CNNs) are commonly employed for feature extraction in image tasks. However, training a CNN from the beginning demands substantial time and data. Transfer learning presents a solution by utilizing pre-trained models to extract image features, reducing the training burden and enhancing efficiency [29].

In this study, we utilized four well-known feature extraction models, including the Inception Model, SqueezeNet, VGG19, and VGGFace. These models exhibit unique architectures and training methods, making them suitable for different tasks. The evaluation involved analyzing the performance of these commonly used pre-trained deep learning models for feature extraction on our image dataset, with accuracy as the primary assessment metric.

C. Transfer Learning

We utilize transfer learning, a method that involves using a pre-trained model to address new tasks. This strategy involves capitalizing on the knowledge and features acquired by the pre-trained model to aid in training a new model tailored to our specific requirements. By doing so, the process of training machine learning models can be notably accelerated and simplified. Instead of starting the training process from scratch, we can utilize the knowledge and capabilities of a pre-trained model that has already mastered the extraction of valuable data features. This approach is especially beneficial in scenarios with limited data resources.

1) Inception Model

The Inception model stands out as a highly esteemed neural network architecture recognized for its exceptional performance and computational efficiency in computer vision tasks. Its notable capability resides in extracting intricate and valuable features from images, primarily utilizing a specific penultimate "bottleneck layer" to produce a concise 2048-dimensional feature vector for each input image. This achievement is facilitated by employing convolutional layer factorization, which reduces the network's parameters and connections, enhancing both efficiency and accuracy. Additionally, Inception integrates crucial techniques such as batch normalization, stabilizing and expediting network training while addressing gradient issues. To address overfitting, it implements dropout by randomly deactivating neurons during training, and pooling aids in preserving vital features while reducing spatial dimensions. These combined techniques significantly contribute to efficient feature extraction and model simplification, ultimately elevating the model's overall performance [30].

2) Squeeze Net

It is a deep neural network architecture that was introduced in 2016. Its unique design focuses on maintaining a small model size and low computational complexity while delivering high accuracy. It is a favored option for applications operating within constrained computational resources, including mobile devices and embedded systems. During its initial training, SqueezeNet utilized the ImageNet dataset, which comprises millions of classified images spanning diverse classes. This pre-training phase equips SqueezeNet with the ability to learn generalized features adaptable to various image classification tasks [31].

In the transfer learning process using SqueezeNet, we utilize a pre-trained model to extract features from the dataset. This involves inputting images into the pre-trained SqueezeNet model and gathering output features from one of its intermediate layers. These features extracted can be used to train a new classifier tailored to our specific test datasets.

3) VGG19

VGG19 is a deep convolutional neural network architecture that comprises 19 layers, encompassing 16 convolutional layers and 3 fully connected layers. The architecture utilizes 3x3 filters with a stride of 1 in its convolutional layers, while the pooling layers employ 2x2 filters with a stride of 2. This design enables VGG19 to extract distinct and detailed features from images, making it a popular choice for tasks such as identification, recognition, and classification [32].

We utilize the pre-trained weights of VGG19, initially trained on ImageNet, and adapt the model for our purposes. By using our own classification layer instead of the final softmax layer, we modify only the classifier's weights during training while maintaining the other layers unaltered. This strategy combines the strength of feature extraction of VGG19 with our classification task.

D. Classification

Building a face recognition system involves a crucial stage of training classifiers using features extracted from pre-trained models. The fine-tuned model compares the obtained facial features with stored data of registered individuals in the attendance database. These classifiers differentiate and identify faces within the system. Our system integrates four distinct classifiers: SVM, Random Forest, Logistic Regression, and Multi-Layer Perceptron (MLP). The following section will delve into each classifier, offering detailed insights into their operations and their integration within our face recognition system.

1) Logistic Regression

Logistic regression stands out as a prevalent statistical model extensively employed in binary classification tasks [33]. The logistic function is usually denoted mathematically as follows:

$$y = \frac{1}{1 + e^{-(\beta_0+\beta_1 x_1)}} \tag{1}$$

Where β_0 is the intercept term, a constant that shifts the curve left or right, while β_1 is the Coefficient for the predictor variable x_1, determining the direction and strength of its impact on the predicted probability. This form of regression analysis focuses on understanding how independent variables relate to a binary outcome variable. Typically used as the final classification step, its aim is to predict the probability of each class label.

Logistic regression in face recognition acts as a predictive technique used to estimate the probability of an image belonging to a particular category. It achieves this by computing a linear combination of the input features and then applying a sigmoid activation function. Known for offering understandable outcomes and streamlined training, logistic regression proves effective for image classification tasks in high-dimensional feature spaces.

We've implemented logistic regression as the ultimate classification layer for each pre-trained model within our face recognition system. Each pre-trained model's output is fed into the logistic regression layer, which computes probabilities linked to each class label. The predicted label is then decided based on the class that demonstrates the highest probability.

A Logistic Regression classifier is trained to map the extracted features $\mathbf{F}_i$ to student labels y_i. The logistic function is used to model the probability of each student y given the features $\mathbf{F}_i$:

$$P(y|\mathbf{F}_i) = \frac{1}{1 + e^{-(\mathbf{W}\cdot\mathbf{F}_i+\mathbf{b})}} \tag{2}$$

Where $\mathbf{W}$ represents a vector of weights associated with each feature in $\mathbf{F}_i$. The weights determine the impact of each feature on the prediction. $\mathbf{F}_i$ is a vector represents the features of the input data for the i-th observation. Each feature is multiplied by its corresponding weight in $\mathbf{W}$. $\mathbf{b}$ represents the bias term which is a constant that shifts the decision boundary. It's added to the weighted sum of features before applying the logistic function.

2) Random Forest

Random Forest is a commonly used ensemble learning method rooted in decision trees, frequently applied in regression and classification tasks. Its core purpose is to improve the predictive accuracy of machine learning models by integrating numerous decision trees to model complex relationships between input features and target variables.

The process begins with random subset selection, which involves randomly selecting a subset of training samples of size N_{sample} and a subset of input features of size M_{subset}. Following this, decision tree construction takes place by recursively partitioning the feature space to minimize impurity at each node. Finally, ensemble aggregation combines the predictions of all trees using methods such as majority voting for classification tasks. This ensemble approach guarantees robustness and reduces overfitting compared to individual decision trees. In applications such as face recognition, Random Forest effectively learns complex patterns and relationships among facial features, enhancing classification accuracy by managing missing data and providing feature importance rankings.

Random Forest is recognized for its ability to address overfitting, handle missing data, and provide rankings for feature importance. It builds numerous decision trees during training, with each tree in the forest created using a random subset of training data and input features. The ensemble works by combining the predictions of these trees, utilizing a random aspect to ensure variety and lower the chance of overfitting, thereby enhancing overall performance [34].

3) Multi-Layer Perceptron

The employment of MLP in facial recognition systems aims to improve classification accuracy by learning data patterns. MLP consists of multiple layers of nodes, with each layer processing the previous layer's output. Its final layer provides the classification. This study explores the use of Multi-Layer Perceptrons (MLP) employing two optimization methods: Stochastic Gradient Descent (SGD) and Adaptive Moment Estimation (Adam) for facial recognition using transfer learning. The Cross-Entropy function serves as the loss function for training the MLP classifier, optimized either by Adam or SGD algorithms. The initial learning rate was set to 1-e3. Implementation of pre-trained models and MLP was conducted using the PyTorch library.

a) Stochastic Gradient Descent

The SGD optimizer is extensively employed in deep learning, adjusting neural network weights based on the loss function's gradient. This process continues iteratively, guiding weights in the opposite direction of the gradient until the loss function reaches its minimum. SGD's simplicity and effectiveness render it crucial in training MLPs. The update rule for the i-th weight at time t is given by:

$$w_{i,t+1} = w_{i,t} - \alpha \frac{\partial L}{\partial w_{i,t}} \tag{3}$$

where $w_(i,t)$ represents the weight of the i-th parameter at iteration t. α is the learning rate, α hyperparameter that controls the step size of the update. $\frac{\partial L}{\partial w_{i,t}}$ is the partial derivative of the loss function L with respect to the i-th weight at iteration t, It represents the slope or gradient of the loss function with respect to that specific weight.

b) Adaptive Moment Estimation

Adam is a popular optimization technique extensively used in training deep neural networks. It combines the Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp) with the fundamental concepts of the SGD algorithm. This dynamic method adapts the learning rate for each parameter during training. Adam is recognized for its robustness to noisy gradients, ability to handle diagonal rescaling of gradients, and effectiveness in dealing with sparse gradients.

The update rule for the i-th weight at time t in Adam is given by:

$$w_{i,t+1} = w_{i,t} - \alpha \cdot \frac{\hat{v}_t}{\sqrt{\hat{m}_t} + \epsilon} \tag{4}$$

Where $w_{i,t+1}$ represents the updated weight of the i-th parameter at the next iteration (t+1), while $w_{i,t}$ denotes the current weight of the same parameter at the current iteration t. The learning rate α determines the step size during each iteration, guiding the algorithm toward a minimum of the loss function. $\hat{v}_t$ is the estimate of the first moment (the mean) of the gradients, obtained through an exponentially decaying average of past gradients. Similarly, $\hat{m}_t$ represents the estimate of the second moment (the uncentered variance) of the gradients, also computed as an exponentially decaying average of past squared gradients. Finally, $\sqrt{\hat{m}_t}$ corresponds to the square root of the second moment estimate, and ϵ is a very small number to prevent any division by zero.

In simpler terms, this equation is used to update the weights in the neural network in an adaptive manner at each iteration during the training phase. The weights are adjusted in the direction that minimizes the loss function. The adjustment is made based on the estimates of the first and second moments of the gradients.

In both SGD and Adam, the weights are updated at the end of each iteration, so at the end of an epoch, multiple weight updates have been applied. The difference between SGD and Adam is that with SGD the weight updates are simple steps in the direction of the (negative) gradient, while with Adam the gradient steps are scaled using running statistics of previous weight updates.

5. PROPOSED APPROACH

We propose a real-time attendance system using face recognition technology incorporating MTCNN, VGGFace, and SVM. Our model swiftly extracts features from input images. When an image is captured, the system processes and compares it with the existing database. If a match is found, the system registers the person as present and updates the attendance record accordingly. Further details regarding the operational steps of this system are explained in the following sections.

A. Data Collection

The first step is to gather a labeled dataset of face images $\{(X_i, y_i)\}$, where X_i represents the ith face image, and y_i is the corresponding label, indicating the student's identity.

Concerning the dataset, we combined two diverse sources: the ”Pins” dataset, sourced from Pinterest and containing facial images, and the ”SPU” dataset, comprised of images collected from students at the Syrian Private University. This combined dataset comprises a total of 18,036 faces representing 115 different individuals.

This dataset showcases a diverse range of images, demonstrating variations in pose, age, lighting conditions, facial expressions, and the presence of accessories like hats, eyeglasses, and sunglasses, as illustrated Figure 2.

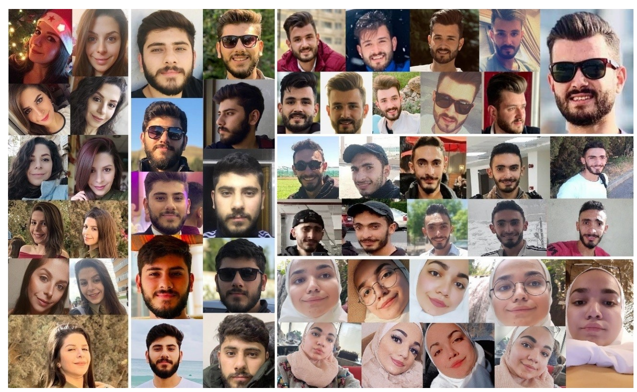

Figure 2. Sample Faces from Syrian Private University Dataset

To facilitate algorithm development and evaluation, the dataset was divided into three subsets: the training set (80% of the images), the test set (10%), and the validation set (remaining 10%). Researchers can utilize this dataset as a valuable resource to enhance face recognition algorithms and address real-world challenges.

B. Face Detection using MTCNN

We employ the MTCNN model to detect faces within an input image. It is a multi-step deep learning architecture specifically designed for face detection, encompassing three stages: face detection, bounding box regression, and facial landmark detection [28]. MTCNN is a face detection algorithm that consists of three cascaded CNNs:

P-Net (Proposal Net) is tasked with creating potential face bounding boxes by processing an input image, generating a series of bounding boxes with corresponding confidence scores.

R-Net (Refine Net) works to improve the initially proposed bounding boxes from P-Net. By taking the input image and P-Net's bounding boxes as input, R-Net produces a refined set of bounding boxes and confidence scores.

O-Net (Output Net) handles the conclusive stage of the MTCNN algorithm. It utilizes the input image and bounding boxes from R-Net to deliver a final set of bounding boxes, confidence scores, and facial landmarks.

The P-Net network is trained to reduce the following loss function:

$$L_P = \lambda_1 L_{cls} + \lambda_2 L_{reg} \tag{5}$$

where:

L_{cls} is the classification loss, which measures how well the network can distinguish between faces and non-faces. L_{reg} is the regression loss, which measures how well the network can predict the bounding box coordinates of faces. λ_1 and λ_2 are hyperparameters that control the relative importance of the classification and regression losses.

The classification loss is calculated using a softmax function:

$$L_{cls} = -\sum_{i=1}^{2} y_i \log(p_i) \tag{6}$$

where:

i is the index of the class (face or non-face). y_i is the ground truth label for class i. p_i is the predicted probability for class i.

The regression loss is calculated using a Euclidean distance metric:

$$L_{reg} = \sum_{i=1}^{4} (x_i^p - x_i^g)^2 + (y_i^p - y_i^g)^2 \tag{7}$$

where:

i is the index of the bounding box coordinate (x1, y1, x2, y2). x_i^p and y_i^p are the predicted bounding box coordinates. x_i^g and y_i^g are the ground truth bounding box coordinates.

The R-Net network is trained to minimize the following loss function:

$$L_R = \lambda_1 L_{cls} + \lambda_2 L_{reg} + \lambda_3 L_{lm} \tag{8}$$

where:

L_{lm} is the landmark loss, which measures how well the network can predict the facial landmarks of faces. λ_3 is a hyperparameter that controls the relative importance of the landmark loss.

The landmark loss is calculated using a Euclidean distance metric:

$$L_{lm} = \sum_{i=1}^{5} (x_i^p - x_i^g)^2 + (y_i^p - y_i^g)^2 \tag{9}$$

where:

i is the index of the facial landmark (left eye, right eye, nose, mouth left corner, mouth right corner). x_i^p and y_i^p are the predicted facial landmark coordinates. x_i^g and y_i^g are the ground truth facial landmark coordinates.

The O-Net network is trained to minimize the following loss function:

$$L_O = \lambda_1 L_{cls} + \lambda_2 L_{reg} + \lambda_3 L_{lm} \tag{10}$$

where the loss functions are the same as for the R-Net network.

Upon completion of training, the MTCNN networks gain the ability to detect faces within fresh images. The initial step involves using the P-Net network on the input image to create preliminary face bounding boxes. Following this, the R-Net network works to refine the precision of these initial bounding boxes. Ultimately, the O-Net network is employed to generate the definitive assortment of bounding boxes, along with confidence scores and facial landmarks.

The MTCNN algorithm is highly preferred for face detection because of its combination of precision and speed, consistently showcasing superior performance in various face detection benchmarks.

C. *Feature Extraction using VGGFace*

VGGFace stands as a pre-trained Deep Convolutional Neural Network (DCNN) model specialized in facial recognition tasks. It has been trained extensively on an extensive dataset of faces, enabling it to extract critical features essential for tasks like face recognition, verification, and identification. It is essentially an adjusted rendition of the VGG-16 model, precisely calibrated for adept extraction of facial features. The training regimen incorporated a dataset comprising 2.6 million facial images portraying 2,622 individuals, with the explicit objective of pushing the boundaries of face recognition technology [35].

To perform transfer learning using VGGFace, the initial step involves substituting the model's output layer with a task-specific layer. Once we fine-tune the VGGFace model on our dataset, the features extracted from it serve as input for different classifiers. When utilizing a classifier, the primary procedure entails extracting features from our dataset images via the pre-trained VGGFace model. Consequently, we proceed to train the classifier by associating labels with these features. After the training phase, the classifier gains the capability to forecast labels for new images. Figure 3 illustrates the architecture of the VGGFace network.

The feature extraction process entails the passage of the input image I_i through the VGGFace model to acquire the feature map $\mathbf{F}_i^{(l)}$. This can be represented as:

$$\mathbf{F}_i^{(l)} = \mathrm{CNN}_{\mathrm{VGGFace}}(I_i) \tag{11}$$

Here, $\mathbf{F}_i^{(l)}$ represents the features extracted from the ith face image at the lth layer of the VGGFace model. These extracted features $\mathbf{F}_i^{(l)}$ are then used as inputs for subsequent stages in the face recognition process.

D. SVM Model

Support Vector Machines (SVMs) stand as a prominent classification method within machine learning, lauded for its proficiency in handling high-dimensional data and resilience against noise [36]. In face recognition systems, SVMs play a pivotal role in accurately classifying input images based on individual identities. SVM learns a decision boundary that segregates various classes in the feature space. This boundary aims to maximize the margin between the closest training data points and itself. Represented as:

Given a set of training data (X_i, y_i), where X_i represents the feature vectors and y_i denotes the class labels, the SVM aims to find the optimal hyperplane represented as:

$$w^T x + b = 0 \tag{12}$$

Here, w denotes the weight vector, x represents the input feature vector, and b is the bias term. The objective involves maximizing the margin between the hyperplane and the support vectors, which are the closest points to the hyperplane from both classes. This optimization problem can be formulated as:

$$\min_{w,b} \frac{1}{2}||w||^2 \tag{13}$$

subject to:

$$y_i(w^T x_i + b) \geq 1 \tag{14}$$

The key objective is to identify the hyperplane that best separates the classes while minimizing classification errors. This ability allows the SVM model to generalize effectively on unseen data, contributing significantly to the accuracy and reliability of face recognition systems.

E. Attendance Marking and Reporting

After accurately identifying each student, their attendance for the ongoing session is logged. The attendance details are promptly refreshed within the attendance management system hosted on the cloud, ensuring immediate and centralized record maintenance. This entire procedure can be mathematically represented as:

Let $\mathbf{X}_i$ denote the feature vector extracted from the ith student's facial image using our trained model, and $\mathbf{y}_i$ represent the corresponding attendance status, with values such as 1 denoting "present" and 0 indicating "absent." The recognition procedure can be formally represented as:

$$\mathbf{y}_i = \mathrm{Recognition}(\mathbf{X}_i) \tag{15}$$

Here, Recognition($\cdot$) symbolizes the recognition function, which assesses the attendance status based on the extracted features $\mathbf{X}_i$.

Subsequently, the attendance records are effortlessly synchronized within the cloud-based system, guaranteeing their alignment with the current attendance status and ensuring a precise and current attendance log.

We uphold detailed attendance logs, recording dates, times, and students' attendance statuses for comprehensive auditing. Following this, we create attendance reports and distribute them among relevant stakeholders. To handle inaccuracies in face detection or recognition, we have integrated error rectification measures. Routine validation and testing procedures are in place to guarantee the precision and dependability of the system. It is configured for settings such as classrooms and synchronized with a cloud-based attendance management system, streamlining the process of attendance recording and reporting.

6. ALGORITHM DIAGRAM

The algorithm for the "Face Recognition Based Attendance System" automates attendance monitoring through facial recognition technology. It operates by identifying faces within an input image via MTCNN, detecting facial features using VGGFace, and classifying them using Support Vector Machine. Upon classification as "Present," it updates the attendance records and generates a report. This system significantly improves attendance management, particularly in situations involving numerous concurrent face recognition instances.

7. EVALUATION METRICS

In this section, we'll explore the evaluation metrics utilized to measure the performance of our face recognition attendance system. we employed two fundamental evaluation metrics:

A. Accuracy

In tasks involving classification, like face recognition systems, accuracy plays a vital role. It quantifies the percentage of correctly classified samples out of the total sample count. The accuracy formula for our attendance system is represented as:

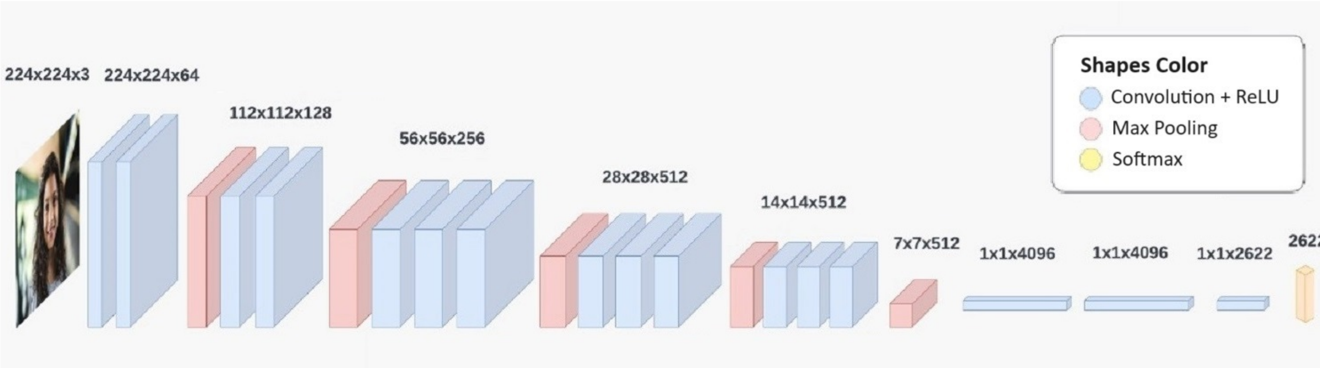

Figure 3. VGGFace Network Architecture

Algorithm 1 Face Recognition-Based Attendance System

Input: Database of labeled face images $\{(X_i, y_i)\}$, Input Image I
Output: Updated attendance records
procedure MAIN($\{(X_i, y_i)\}$, I)
  $B \leftarrow$ MTCNN_Detect(I)
  $F \leftarrow$ VGGFace_ExtractFeatures(B)
  for $i \leftarrow 1$ to length(F) **do**
    $y \leftarrow$ SVM_Classify($F[i]$)
    if y = "Present" **then**
      UpdateAttendanceRecord(y_i)
    end if
  end for
  GenerateAttendanceReport()
end procedure

$$\text{Accuracy} = \frac{\text{Correctly recognized faces}}{\text{Total faces in test dataset}} \times 100\% \quad (16)$$

Although accuracy provides a straightforward measure of the system's ability to correctly identify individuals within the dataset, it may not be sufficient, especially when dealing with imbalanced datasets or when the implications of false positives and false negatives vary.

B. F1 Score

This metric strikes a balance by considering both precision and recall. Precision signifies the proportion of true positive predictions relative to all positive predictions, while recall measures true positive predictions concerning all actual positive instances. The F1 score is calculated as the harmonic mean of precision and recall:

$$\text{F1 Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (17)$$

Precision is the ratio of true positive predictions to the sum of true positive and false positive predictions:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad (18)$$

Recall, also known as sensitivity or true positive rate, is calculated as the ratio of true positive predictions to the sum of true positive and false negative predictions:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (19)$$

The F1 score proves particularly useful when dataset instances are imbalanced or when different costs are associated with false positives and false negatives. It provides a comprehensive evaluation of the system's performance by accounting for both types of misclassifications.

8. DISCUSSION OF RESULTS

In our study, we explored various deep learning models - VGG19, VGGFACE, SqueezeNet, and the Inception model - for transfer learning. These models were paired with distinct classifiers: Support Vector Machine (SVM), Logistic Regression (LR), Random Forest, and Multi-Layer Perceptron (MLP). Through meticulously designed experiments, we aimed to assess their performance using our dataset. Our results unequivocally highlighted VGGFACE combined with SVM as the top performer, demonstrating the highest overall effectiveness. Additionally, fine-tuning the VGGFACE network on our specific dataset notably enhanced the model's performance. Overall, our findings indicate that VGGFACE with SVM presents a promising approach for transfer learning in image classification tasks.

The data in Table I displays the effectiveness of different embedding models combined with a SVM classifier. It outlines details such as parameter count per model, feature shape extracted, and the achieved accuracy and F-score measurements. Remarkably, the "VGGFace" model showcases superior accuracy and F-score at 95%, while alternative models like "Squeeze Net" exhibit comparatively lower accuracy and F-score outcomes.

TABLE I. THE IMPACT OF VARIOUS EMBEDDING MODELS ON THE PERFORMANCE OF THE SVM CLASSIFIER

Embedding Model	No of parameters	Feature shape	Accuracy	F1-Score
Inception Model	21,802,784	2048	39%	40%
Squeeze Net	1,235,496	1000	32%	33%
VGG19	122,788,928	4096	43%	45%
VGGFace	145,002,878	2622	95%	95%

The inclusion of SVM as a classifier in the face recognition system explained in this paper presents a dependable and accurate method for determining a person's identity through the analysis of their facial features.

In Table II the Logistic Regression classifier's performance is assessed across different embedding models, it's worth mentioning that although both the Logistic Regression and SVM classifiers achieved identical accuracy and F1-score (95%) using the "VGGFace" model, their performance differs noticeably when tested with other models. This discrepancy emphasizes the significance of selecting an appropriate classifier, considering how it can yield diverse impacts on distinct feature sets. It underscores the importance of matching the classifier choice with the unique traits of data and feature representations derived from various models.

The data presented in Table III showcases how the Random Forest classifier performs across various embedding models. Notably, the "VGGFace" model attained notably high accuracy and F-score of 93%, yet the Random Forest classifier exhibits relatively poorer performance in comparison to both Logistic Regression and SVM classifiers. Across other models such as "Inception Model", "Squeeze Net" and "VGG19", the Random Forest also displays reduced accuracy and F-score values. These findings imply that the Random Forest classifier may not be as adept in handling the distinctive feature representations produced by these embedding models.

We have utilized the SGD optimizer across four instances with various embedding models. Assessing the performance of the Multi-Layer Perceptron (MLP) classifier in conjunction with the SGD optimizer involved evaluating distinct embedding models, outlined in the subsequent table.

The MLP, integrated with the SGD optimizer, serves as the classifier relying on the distinctive features extracted by pre-trained models. Each pre-trained model's output interfaces with the MLP to predict the identity associated with the input face.

The equations used to compute the output of an MLP are as follows:

$$h_i = \sum_{j=1}^{m} w_{i,j} x_j + b_i \tag{20}$$

$$y_i = f(h_i) \tag{21}$$

Where x_j is the input feature, $w_{i,j}$ is the weight connecting input x_j to hidden neuron i, b_i is the bias of hidden neuron i , h_i is the weighted sum of inputs to hidden neuron i, f is the activation function of the hidden layer, and y_i is the output of the hidden layer.

Table IV illustrates how the MLP classifier performs when using an SGD optimizer across diverse embedding models. Specifically, the "VGGFace" model stands out, exhibiting a notable accuracy and F1-score of 92%. Nevertheless, the performance of the MLP classifier displays considerable variation when applied to distinct embedding models, such as "Squeeze Net," resulting in notably lower accuracy and F1-score values. This emphasizes the sensitivity of the MLP with SGD concerning the distinctive features extracted by different models.

We employed an MLP with the Adam optimizer as the classifier to identify faces in provided images, utilizing learned features from pre-trained models. The output of these pre-trained models served as input features for the MLP during classification. Throughout the training phase, the Adam optimizer continuously adjusted the MLP's weights and biases based on the computed loss derived from the disparity between predicted and actual labels. The application of the MLP with the Adam optimizer in our face recognition system involved several crucial steps. Initially, we loaded the pre-trained models and maintained their weights unchanged to extract features from the facial images. Following this, the MLP was engaged for classification using these extracted features. Utilizing backpropagation alongside the Adam optimizer, we iteratively refined the MLP by updating weights and biases to minimize loss. Ultimately, the trained MLP was applied to classify new facial images by inputting their features into the MLP, thereby obtaining predicted labels.

The performance of the MLP Classifier with the Adam optimizer using various embedding models is presented in Table V It is worth noting that the "VGGFace" model demonstrates considerable effectiveness with a relatively high accuracy and F1-score of 91%. However, akin to the scenario with the SGD optimizer, the MLP classifier's performance significantly differs across diverse embedding models when using the Adam optimizer. Notably, the "Squeeze Net" model exhibits notably lower accuracy and

TABLE II. THE IMPACT OF VARIOUS EMBEDDING MODELS ON THE PERFORMANCE OF THE LOGISTIC REGRESSION CLASSIFIER

Embedding Model	No of parameters	Feature shape	Accuracy	F1-Score
Inception Model	21,802,784	2048	47%	48%
Squeeze Net	1,235,496	1000	41%	40%
VGG19	122,788,928	4096	55%	56%
VGGFace	145,002,878	2622	95%	95%

TABLE III. THE IMPACT OF VARIOUS EMBEDDING MODELS ON THE PERFORMANCE OF THE RANDOM FOREST CLASSIFIER

Embedding Model	No of parameters	Feature shape	Accuracy	F1-Score
Inception Model	21,802,784	2048	32%	31%
Squeeze Net	1,235,496	1000	28%	28%
VGG19	122,788,928	4096	35%	35%
VGGFace	145,002,878	2622	93%	93%

TABLE IV. THE IMPACT OF VARIOUS EMBEDDING MODELS ON THE PERFORMANCE OF THE MLP CLASSIFIER WITH SGD OPTIMIZER

Embedding Model	No of parameters	Feature shape	Accuracy	F1-Score
Inception Model	21,802,784	2048	45%	46%
Squeeze Net	1,235,496	1000	13%	9%
VGG19	122,788,928	4096	41%	43%
VGGFace	145,002,878	2622	92%	92%

F1-score values, reaching only 7% and 4%, respectively. This emphasizes how the MLP classifier's performance is highly sensitive to both the chosen feature representations and optimizer, highlighting the crucial role of meticulous model selection and optimization for achieving optimal outcomes.

Table VI presents a quantitative comparison of face recognition performance achieved by various classifiers using the VGGFace model. Each classifier is evaluated based on the number of faces correctly recognized faces (TP) and the number of misidentifications made (FP).

From the table, it can be observed that the SVM classifier has the highest number of recognized faces (1712) and the lowest number of misidentifications (90), making it the most effective classifier among those tested. This suggests that SVM, when used in conjunction with the VGGFace model, is highly effective for face recognition tasks.

On the other hand, the MLP (Adam) classifier has the lowest number of recognized faces (1640) and the highest number of misidentifications (162), indicating that it may not be as effective as the other classifiers when used with the VGGFace model for face recognition.

The results highlight the importance of choosing the right classifier when designing a face recognition system. While the VGGFace model provides a robust feature extraction mechanism, the choice of classifier can significantly impact the system's performance.

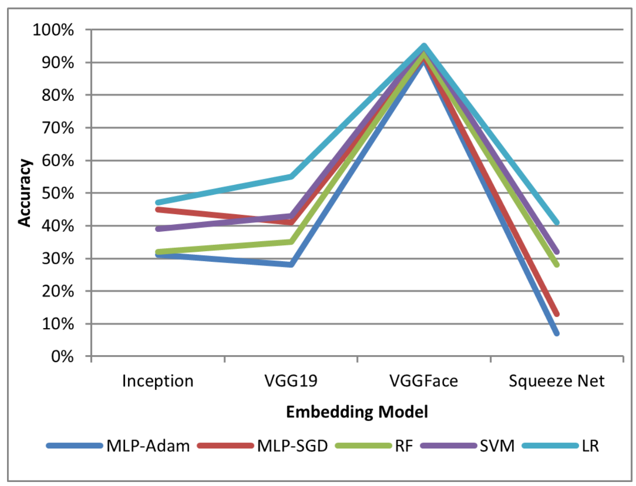

Figure 4. Performance of various classifiers depending on the results of various embedding models

The findings demonstrate that the accuracy achieved using the VGGFACE model is superior to that of other models, as illustrated in Figure 4.

During the evaluation of various pre-trained models, the superior performance of the VGGFACE feature extraction model in terms of accuracy and F1 score is evident. VGGFACE's advantage lies in its specialized training on a dataset specifically dedicated to face identity recognition, encompassing over 2.6 million face images from 2,622

TABLE V. THE IMPACT OF VARIOUS EMBEDDING MODELS ON THE PERFORMANCE OF THE MLP CLASSIFIER WITH ADAM OPTIMIZER

Embedding Model	No of parameters	Feature shape	Accuracy	F1-Score
Inception Model	21,802,784	2048	31%	33%
Squeeze Net	1,235,496	1000	7%	4%
VGG19	122,788,928	4096	28%	29%
VGGFace	145,002,878	2622	91%	91%

TABLE VI. Quantitative Comparison of Face Recognition Performance Using VGGFace Model Across Various Classifiers

Classifier	Recognized Faces	Misidentifications
SVM	1712	90
LR	1712	90
Random Forest	1676	126
MLP (SGD)	1658	144
MLP (Adam)	1640	162

different identities. In contrast, other models were trained on the more general ”ImageNet” dataset, which includes images from 1,000 diverse object classes.

The smaller size (1.23 M) of SqueezeNet and the dataset it was trained on led to lower performance. The choice of a pre-trained model significantly impacted each classifier's performance. For instance, SVM and LR demonstrated the highest accuracy rates when combined with the VGGFACE pre-trained model.

Our research findings emphasize the critical considerations needed when selecting a pre-trained model for a face recognition system. Despite Inception-v3 having 22M parameters and VGG-19 having 123M parameters, Inception-v3 outperformed VGG-19 when paired with MLP. Inception v3's incorporation of a residual block helps it memorize inputs, overcoming the vanishing problem. In terms of classifiers, RF and MLP demonstrated lower accuracy rates compared to SVM and LR. Among these classifiers, SVM emerged as an effective choice, achieving high accuracy rates when combined with pre-trained models. LR also provides a robust and accurate means of predicting a person's identity based on facial features.

Notably, LR consistently outperformed SVM in most scenarios due to SVM's sensitivity to the number of features—an increased number of features makes it harder for the algorithm to differentiate between classes. However, we propose SVM as the proposed method because it finds the "best" margin that separates the classes, reducing the risk of error on the data, while LR does not have this property. This leads to better generalization and potentially lower overfitting risk with SVM. Furthermore, SVM can effectively handle high-dimensional data and model complex decision boundaries, which are often required in face recognition due to the complex relationships among facial features.

Additionally, SVM can handle complex decision boundaries better than linear LR. It is more robust to class imbalance problems, which are common in face recognition tasks. SVM focuses on support vectors closest to the decision boundary, making them less sensitive to outliers in the data. This is beneficial in scenarios where faces can exhibit variations due to lighting, pose, and expression. On the other hand, LR assumes a linear relationship between features and the target variable, potentially limiting its performance on complex datasets. It can be susceptible to outliers, potentially affecting the decision boundary, and doesn't explicitly optimize for margin maximization, potentially leading to a higher overfitting risk. Therefore, considering these factors, we have chosen SVM as the proposed method for our face recognition system.

9. CONCLUSIONS

In this research, we extensively delved into face recognition systems, focusing on methods for extracting features and classifiers. Our investigation highlighted the exceptional effectiveness of the VGGFace model, trained on a vast dataset for face identity recognition. Utilizing its features led to achieving accuracy rates of up to 95%, with the SVM classifier demonstrating particularly outstanding performance by correctly recognizing 1712 faces.

Moreover, our research presented an advanced face recognition-based on attendance system adept at simultaneously detecting and identifying multiple faces. By combining MTCNN for face detection, VGGFace for feature extraction, and SVM for classification, we introduced a system streamlining attendance monitoring, improve security measures, reduces manual effort, and furnishing real-time attendance data. Its versatility across educational environments, workplaces, and substantial gatherings underscores its value as an effective attendance management tool.

Our research emphasizes the significance of choosing suitable pre-trained models and classifiers for face recognition assignments. It also underscores the opportunity for enhancements, such as integrating data augmentation and ensemble learning methods. We remain committed to progressing facial recognition technology, aiming to develop more accurate and reliable systems that meet the diverse needs of today's society.

References

[1] Y. Jing, X. Lu, and S. Gao, "3d face recognition: A comprehensive survey in 2022," *Comp. Visual Media*, vol. 9, pp. 657–685, 2023.

[2] A. Shukla, A. Shukla, and R. Singh, "Automatic attendance system based on cnn–lstm and face recognition," *International Journal of Information Technology*, 2023.

[3] A. Diwan, D. Kumar, R. Mahadeva, H. C. S. Perera, and J. Alawatugoda, "Unveiling copy-move forgeries: Enhancing detection with superpoint keypoint architecture," *IEEE Access*, vol. 11, pp. 86 132–86 148, 2023.

[4] A. Kumar, S. Jain, and M. Kumar, "Face and gait biometrics authentication system based on simplified deep neural networks," *International Journal of Information Technology*, vol. 15, pp. 1005–1014, 2023.

[5] N. Arora, G. Sucharitha, and S. Sharma, "Mvm-lbp: Mean-variance-median based lbp for face recognition," *International Journal of Information Technology*, vol. 15, pp. 1231–1242, 2023. [Online]. Available: https://doi.org/10.1007/s41870-023-01204-y

[6] B. Tej Chinimilli, A. T., A. Kotturi, V. Reddy Kaipu, and J. Varma Mandapati, "Face recognition based attendance system using haar cascade and local binary pattern histogram algorithm," in *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184)*. Tirunelveli, India: IEEE, 2020, pp. 701–704.

[7] B. Kocacinar, B. Tas, F. P. Akbulut, C. Catal, and D. Mishra, "A real-time cnn-based lightweight mobile masked face recognition system," *IEEE Access*, vol. 10, pp. 63 496–63 507, 2022.

[8] D. A. Maharani, C. Machbub, P. H. Rusmin, and L. Yulianti, "Improving the capability of real-time face masked recognition using cosine distance," in *2020 6th International Conference on Interactive Digital Media (ICIDM)*. Bandung, Indonesia: IEEE, 2020, pp. 1–6.

[9] I. Razaq and B. Shukur, "Combining wavelet transforms features and high-level features using cnn for face morphing attack detection," *International Journal of Information Technology*, vol. 15, pp. 3957–3966, 2023.

[10] M. Ali and D. Kumar, "A combination between deep learning for feature extraction and machine learning for recognition," in *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Kharagpur, India, 2021, pp. 01–05.

[11] M. Geetha, R. Latha, S. Nivetha, S. Hariprasath, S. Gowtham, and C. Deepak, "Design of face detection and recognition system to monitor students during online examinations using machine learning algorithms," in *2021 International Conference on Computer Communication and Informatics (ICCCI)*, Coimbatore, India, 2021, pp. 1–4.

[12] D. Ali, I. Touqir, A. M. Siddiqui, J. Malik, and M. Imran, "Face recognition system based on four state hidden markov model," *IEEE Access*, vol. 10, pp. 74 436–74 448, 2022.

[13] T.-V. Dang, "Smart attendance system based on improved facial recognition," *Journal of Robotics and Control (JRC)*, vol. 4, no. 1, 2023.

[14] K. B. Shibu Kumar and P. Samuel, "An efficient method to accurately cluster large number of high dimensional facial images," *IEEE Access*, vol. 11, pp. 39 934–39 949, 2023.

[15] V. Rajyalakshmi and K. Lakshmanna, "Intelligent face recognition based multi-location linked iot based car parking system," *IEEE Access*, vol. 11, pp. 84 258–84 269, 2023.

[16] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019, pp. 4685–4694.

[17] P. Yuan, R. Huang, J. Zhang, E. Zhang, and X. Zhao, "Accuracy rate maximization in edge federated learning with delay and energy constraints," *IEEE Systems Journal*, vol. 17, no. 2, pp. 2053–2064, 2023.

[18] H. Yang and X. Han, "Face recognition attendance system based on real-time video processing," *IEEE Access*, vol. 8, pp. 159 143–159 150, 2020.

[19] T. Ahmed, P. Das, M. F. Ali, and M.-F. Mahmud, "A comparative study on convolutional neural network based face recognition," in *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Kharagpur, India, 2020, pp. 1–5.

[20] M. Z. Khan, S. Harous, S. U. Hassan, M. U. Ghani Khan, R. Iqbal, and S. Mumtaz, "Deep unified model for face recognition based on convolution neural network and edge computing," *IEEE Access*, vol. 7, pp. 72 622–72 633, 2019.

[21] A. Potdar, P. Barbhaya, and S. Nagpure, "Face recognition for attendance system using cnn based liveliness detection," in *2022 International Conference on Advances in Computing, Communication and Materials (ICACCM)*, Dehradun, India, 2022, pp. 1–6.

[22] A. Autade, P. Adhav, A. BabarPatil, A. Dhumal, S. Vispute, K. Rajeswari, M. Ahirrao, and S. Rathi, "Automated multi face recognition and identification using facenet and vgg-16 on real-world dataset for attendance monitoring system," in *2023 7th International Conference On Computing, Communication, Control And Automation (ICCUBEA)*, Pune, India, 2023, pp. 1–5.

[23] D. Y. Perwej, A. Trivedi, C. Tripathi, A. Srivastava, and N. Kulshrestha, "Face recognition based automated attendance management system," *International Journal of Scientific Research in Science and Technology*, vol. 9, pp. 261–268, 02 2022.

[24] A. Kumar, A. Kaur, and M. Kumar, "Face detection techniques: a review," *Artificial Intelligence Review*, vol. 52, pp. 927–948, 2019.

[25] R. Mishra, "Persuasive boundary point based face detection using normalized edge detection in regular expression face morphing," in *2023 International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*, Ballar, India, 2023, pp. 1–4.

[26] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, "A convolutional neural network cascade for face detection," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, 2015, pp. 5325–5334.

[27] S. S. Farfade, M. Saberian, and L.-J. Li, "Multi-view face detection using deep convolutional neural networks," in *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*. Shanghai, China: ACM, 2015.

[28] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE*

Signal Processing Letters, vol. 23, no. 10, pp. 1499–1503, October 2016.

[29] M. Ali and D. Kumar, "The impact of optimization algorithms on the performance of face recognition neural networks," *Journal of Advanced Engineering and Computation*, vol. 6, no. 4, pp. 248–259, 2022.

[30] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826.

[31] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and ¡0.5mb model size," in *Computer Vision and Pattern Recognition*, 2016.

[32] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations (ICLR)*, 2015.

[33] X. Zou, Y. Hu, Z. Tian, and K. Shen, "Logistic regression model optimization and case analysis," in *2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT)*, 2019, pp. 135–139.

[34] A. S. More and D. P. Rana, "Review of random forest classification techniques to resolve data imbalance," in *2017 1st International Conference on Intelligent Systems and Information Management (ICISIM)*, Aurangabad, India, 2017, pp. 72–78.

[35] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proceedings of the British Machine Vision Conference (BMVC)*, X. Xie, M. W. Jones, and G. K. L. Tam, Eds. BMVA Press, September 2015, pp. 41.1–41.12.

[36] G. Guo, S. Li, and K. Chan, "Face recognition by support vector machines," in *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*, 2000, pp. 196–201.

Mahmoud Ali is a Ph.D. Scholar in the Computer Engineering - Artificial Intelligence and Big Data department at Marwadi University, Rajkot, India. He obtained his B.E. degree in Information Technology from the Syrian Virtual University, Damascus, Syria, in 2013, and an M.Sc. degree in Web Science from the same institution in 2018. His research interests include Semantic Web, Linked Open Data, Computer Vision, Machine Learning, and Deep Learning. He is dedicated to advancing the fields of web science and artificial intelligence through his academic pursuits and contributions.

Anjali Diwan possesses extensive expertise as an academic and professional in the software industry, amassing over two decades of experience. Her diverse areas of academic and research focus include Machine Learning, Image Processing, Artificial Intelligence, Deep Learning, Data Security, Multimedia Forensics, and the application of technology to address humanitarian challenges. As a distinguished professional, Dr. Diwan holds the status of a senior member of the IEEE. Currently, she serves as Secretary IEEE Gujarat Section, she was an integral part of the IEEE R10 SAC team as a senior advisor of the subcommittee (2023-2024) and served as the Section Chair of the IEEE Young Professionals affinity group for the Gujarat section (2023-2024). Notably, she previously held the position of Section Chair of Student Activity for IEEE Gujarat from 2016 to 2019 and served as the IEEE WIE affinity group co-chair for the Gujarat section from 2014 to 2018. In addition to her leadership roles, Dr. Diwan actively contributes to the academic community by participating in the Technical Program Committee (TPC) of various IEEE conferences. Her commitment to scholarly endeavors is further evident through her role as a reviewer for international journals and IEEE transactions. Currently, Dr. Anjali Diwan holds a faculty position in the CE-AI/Big Data department at Marwadi University in Rajkot, Gujarat, India.

Dinesh Kumar is a Senior IEEE member and has a faculty position at the Marwadi University, Rajkot, in Computer Engineering – Artificial Intelligence. His education includes an under-graduation degree (Bachelor of Technology) in Electrical Engineering Indian Institute of Technology Kanpur in 2002 and a Ph.D. in Information Science and Technology (in the research domain of Biomedical signal processing and Machine Learning) from the University of Coimbra, Portugal in 2015. His main areas of interest are Signal Processing and Control, Image processing, Soft computing, and Machine learning. He has been an associate professor in the school of computer science engineering and technology of the Bennett University, Greater Noida, India.