Iraqi Journal for Computer Science and Mathematics

Journal Homepage: http://journal.esj.edu.iq/index.php/IJCM
e-ISSN: 2788-7421 p-ISSN: 2958-0544

Facial Recognition Technology-Based Attendance Management System Application in Smart Classroom

Ahmad S. Lateef [1]*, Mohammed Y. Kamil [1]

[1] College of Science, Mustansiriyah University, Baghdad, Iraq
*Corresponding Author: Ahmad S. Lateef

DOI: https://doi.org/10.52866/ijcsm.2023.02.03.012
Received may 2023; Accepted July 2023 ; Available online August 2023

ABSTRACT: Traditional attendance recording methods can be replaced with artificial intelligence (AI) methods. This paper aims to develop an automated attendance system with a user-friendly graphical interface in Arabic to enhance user interaction. The Python programming language is used to build the system, which employs object detection and feature extraction algorithms. This system processes a live broadcast from an IP camera installed in the classroom. The system employs object detection algorithms to detect the faces in the video. The face locations are then sent to the feature extraction algorithms to extract and compare features with the features stored in the database. Our paper concludes that automating the attendance registration process using facial recognition technology can achieve up to 100% accuracy. Comparing our proposed system with previous studies, we find that it outperforms them in several aspects. Our method handles real-time tests with a larger number of students. It accommodates students positioned at different distances from the camera within the classroom, demonstrating our method's effectiveness in automating the attendance registration process.

Keywords: Attendance system, Dlib, Facenet, HOG, LBPH, MySQL, real-time face recognition

1. INTRODUCTION

AI has come a long way, so individuals' aspirations have recently shifted toward automation in all fields. In contrast to humans, AI can efficiently perform repetitive tasks in our daily lives [1]. Among these tasks is registering attendance at universities and educational and governmental institutions. So many lecturers and administrative employees at these institutions anticipate the automation of this time-intensive task. For example, the one-hour lecture in universities must be shortened by at least ten minutes when marking attendance using traditional methods. One of these traditional systems is the paper-and-pen system, in which the lecturer records the time and date on a piece of paper and then passes it to the students, who write their names on it. This system has numerous flaws. One issue is a student's lack of attention in lectures due to his preoccupation with his attendance; additionally, fraud can occur when one of the present students registers the attendance of one of the absent students (impersonation) [2]. The same holds for the system of attendance lists, in which the administrative staff provides the lecturer with a list of student names. The lecturer reads each student's name and marks those present with one mark and those absent with another. This method of recording attendance consumes lecture time. It is a vexing process for both students and lecturers [3]. In addition, traditional attendance systems rely on paper, which can be lost or stolen, making them an unsafe method of attendance recording. Moreover, determining the attendance status of a specific student on a specific date requires a substantial amount of work because it requires searching through a vast number of documents and archives. All of these factors necessitated the development of an alternative method for completing the attendance registration process safely and efficiently without human intervention. Some of these methods use biometric systems, which use a person's fingerprints to record attendance.

This paper utilizes an attendance recording system based on facial recognition. Face recognition is the ability of a computer to identify a person using AI [4]. Due to the widespread availability of cameras, facial recognition can now be performed at the lowest possible cost. This paper identifies faces through the camera. The procedure for recognizing faces consists of two phases: The first is the process of detecting human faces in an image or video (which is a series of successive images), which involves extracting the human face (or faces) present in the image and differentiating it from the image's background, such as the sky, clouds, buildings, fields, or even differentiating it from the rest of the human body since the human face is the region of interest (RoI) in the process of recognizing faces [5]. In the second step, a person's face is recognized by comparing it with another face in the database. Face recognition is among the most challenging computer vision tasks. However, the advanced preprocessing procedures made it very easy and enabled the computer to recognize faces in ways humans could not. This study proposes an accurate automated attendance registration system that uses multiple face recognition approaches. The system was tested on thirty students in a classroom setting with varying lighting, seating distances, and pose variations. Despite these issues, the proposed system was 100% accurate. It used the MySQL database management system to reduce costs and had no physical components other than the computer and the camera.

The first section describes the topic of the study and its relationship to AI. The second section contains a literature survey of research involving automated attendance registration systems based on face recognition technology. The third section describes the minor details used to complete the methods of this study. The findings are summarized in the fourth section. Finally, the conclusion discusses the gaps in the study and existing models. It also tells researchers who come after them how to use face recognition technology to make an automatic electronic attendance system.

2. RELATED WORK

Automating the recording attendance process was necessary to eliminate the problems associated with traditional methods of that process. Several alternative methods have emerged for fully automating this process. Radio Frequency Identification-based (RFID) attendance registration was one of the first to emerge [6]. Nevertheless, this method cannot be used to automate registering attendees for various reasons. This method needs improvements in accuracy and in the time required to accomplish the attendance recording process.

Additionally, this method is considered risky as a person's card can be easily given to someone else to register attendance instead of him, or it can be stolen or lost. In addition, creating cards and purchasing the reader device are costly processes. For these reasons, Alternative automated attendance systems have been developed by researchers. Due to technological advancements and the development of AI, researchers were able to create new technologies to accomplish this challenging task by relying on the biometric system, which employs the vital fingerprints of the human body [7]. One of these systems is attendance registration based on fingerprints. Although this method is considered more accurate and secure than RFID, time and cost limitations remain. The same holds for the iris recognition-based attendance registration system [8]. Using face recognition algorithms, scientists have automated marking attendance because no other system could perform the task quickly and accurately [9]. Face recognition algorithms are considered a breakthrough in automating the attendance process, as they are the only way to record the presence of multiple individuals simultaneously. Face recognition algorithms perform two functions: detecting the human face in an image and trying to separate it from other faces in the database. Both are performed using algorithms that are distinct from one another. These algorithms have been utilized to automate the process of registering attendance.

Viola-Jones [10], which relies heavily on the Haar cascade classifier [11], is the most common face detection algorithm. Computer scientists have implemented several face recognition algorithms using this algorithm. In [12], an automated attendance system has been developed using Viola-Jones for face detection and three facial recognition algorithms (Eigenfaces, Fisherfaces, and LBPH). The proposed system creates its dataset. The first disadvantage of the system is that it was only tested on individual students, which means it may fail when applied to a group of students in the class. The second issue is that the registration process must be completed online, which may be difficult in areas where Internet access is limited (like in developing countries). While [13] used the Scale Invariant Feature Transform Technique (SIFT) for face authentication as a backup to the RFID system, a face-verification system built with microprocessors like the Raspberry Pi adds an extra layer of security. The achieved accuracy was 84%-the accuracy increases as the number of training images increases. The benefit of this work is that it could train the algorithm on a group of students rather than individual students. One of the problems with this work is that it becomes expensive due to using the Raspberry Pi. Another problem is that the rate of facial recognition goes down as the faces get smaller, which happens when the person sits farther away from the camera. So the distance between the face and the camera is one of the most critical factors in the system's performance. Another method for face recognition is implemented in [14], which uses the ORB algorithm for face recognition. ORB is an acronym for (Oriented FAST and Rotated BRIEF). It is composed of two algorithms, BRIEF and FAST. The descriptor is BRIEF, while the FAST is the detector, so the ORB acts as both the descriptor and detector. The proposed system provides a beneficial program for organizations. However, it also has one problem: it tests only students individually. Developing an attendance system depending on the computer and separate webcam is discussed in [15]. It used LBPH for face recognition. It built a system for taking attendance that worked well and had no problems. This system's graphical user interface is advantageous because it facilitates user interaction with the program. [16] intends to use facial recognition technology to develop an electronic attendance registration system based on Android phones using Android Studio. It communicates with the Android smartphone and the Raspberry Pi using PHP, and to perform face recognition and attendance processing tasks on the server, it develops a python-based application using the OpenCV and Scikit-learn libraries for image processing and face image classification, respectively, to carry out face recognition and attendance processing tasks in the server. Face recognition is accomplished by creating an exclusive database. An image is captured and compared to the rest of the images in the database using either KNN or LDA, or logistic regression classifier. The higher achieved accuracy was obtained by using logistic regression. It was nearly 95.21%. Because all this currency is done on a private server, the process needs an Internet connection. As a result, it has issues with the fact that some countries have limited Internet access and that using the Raspberry Pi costs money. Moreover, the system tested only the students individually. An automated attendance system can be made using Gabor feature extraction, as in [17]. This work detects faces from the image using the Viola-Jones algorithm. Then it extracts the detected face's features using Gabor wavelets. Finally, template matching compares the features with the features stored in the dataset. The advantage of this work is that

when creating the database, the person must show a variety of facial expressions while wearing glasses and hats, which helps to strengthen the work. The research shows 88% accuracy, a 75% recall rate, and a 97% precision rate.

All previous methods were specific to applying machine learning algorithms to recognize faces through their harmony with the Viola-Jones algorithm for identifying faces. However, many studies have combined deep learning with the Viola-Jones algorithm to create more robust systems for distinguishing human faces [18, 19]. For example, the paper [20] built an automated attendance system based on convolutional neural networks (CNNs). This work generated an ideal image database (in terms of lighting, occlusions, and facial expressions). CNN extracted features from images and compared them with the data stored in the database. An accuracy of 95.23% was obtained because all the data was perfect and tested on individuals individually. However, this method is less accurate when used simultaneously on multiple databases and groups of students. Another example of deep learning in action with Viola Jones is shown in [21]. Viola-Jones was used to identify the faces, and the database was created. The features are then extracted using Facenet. Finally, one of the KNN or SVM classifiers is used to classify the faces. The accuracy of this work was approximately 96%.

Many other ways to detect a face, such as multitask convolutional neural networks (MTCNN), are used [22-26]. [22] successfully designed and implemented a security surveillance system using MTCNN for face detection and FACENET for face recognition. Individuals' numbers increased from one to five, and the distance between the person and the camera gradually increased to a maximum of seven feet. The algorithm's performance was then measured. Up to 86% accuracy was obtained. In [23], the WIDER FACE dataset was used to train the bounding boxes, while the CelepA dataset was used to train the facial landmarks. It also uses the same algorithms [22] but with an overall accuracy of 95%. In [24], the face detection mechanism is the same, but for face recognition, the authors used VGG19 and obtained good accuracy. [25] It created its dataset with 120 students in five FPT Polytechnic College building classes and used Facenet and Arcface for face recognition. The achieved accuracy was 92%. [26] used fifteen classes of the LFW dataset, detected faces using MTCNN, and extracted features using Facenet. Then, a random image from the fifteen classes is classified using a support vector machine classifier (SVM). The accuracy obtained was up to 99%. There is only one drawback in this work, which is that it does not use real-time to distinguish faces. The only flaw in this work is that it does not recognize faces in real-time due to the significant decrease in accuracy that occurs when doing so because the researcher may be limited in these situations by things like bad lighting or the accuracy of the camera used.

As shown in [27], Microsoft offers the Azure Cognitive Face API, a face recognition and detection process tool. The principle of this API is principal component analysis (PCA). A photograph is taken, and the face in it is detected. Data augmentation is used to rotate and lighten the image of the face. The database was created y converting the image to grayscale and rescaling it to 200 × 200 pixels. The image is resized for two reasons: first, most images have different resolutions, and second, to speed up the algorithm. The accuracy corresponding to the number of people is determined, with 60 being the maximum number. This study utilized SQLite as a database to store information about students or cooperating employees. The camera was moved in every direction using classroom equipment such as a battery, Arduino, and motors.

In addition to the methods described above, we can use the HOG method described in [28]. The accuracy of HOG for feature extraction and KNN for classification is 97%, while the accuracy of PCA or LBPH is 91% and 89%, respectively. Using YOLOv5 for face detection and deep learning for face recognition results in a beautiful automatic attendance system with a 93% accuracy [29]. The face detection process is completed in [30] using a single state detector (SSD) and a ResNet-34-like architecture for feature extraction and classification. A student's information is entered into a MongoDB database. This method has one flaw: it did not simultaneously test the algorithm on many people.

3. FACE DETECTION ALGORITHMS

It is the procedure of identifying a human face in an image and enclosing it in a box to isolate it from the rest of the image. This process was completed using three distinct methods.

3.1 VIOLA-JONES ALGORITHM

It is among the most efficient methods for locating objects in a real-time video [31]. This algorithm includes the following steps.

3.1.1 Haar features

Alfred Haar discovered Haar features related to facial detection. These characteristics are computed by subtracting the sum of the pixel values of two rectangles. Several types of these features exist, but the edge, line, and four-rectangle Haar features are the most well-known. The naming of these features depends on the number of boxes used, as illustrated in FIGURE 1.

Haar features are a series of frames passing over a grayscale image array (after the image has been rendered to grayscale, the Haar features are calculated). Because Haar features are the difference in intensities between selected

areas (bright or dark), they can be calculated at any scale except 1 × 1 pixel because its value is zero. Haar edge features define an object's edges, and their most miniature scale is 1 × 2 or 2 × 1. While linear Haar features detect lines within the body, 3 × 1 or 1 × 3 is the most miniature scale [32]. The diagonal lines and edges are determined using the four-rectangular Haar edges, and the most miniature scale is 2 × 2.

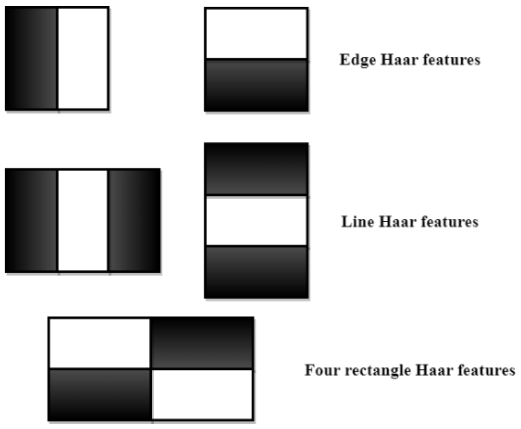


FIGURE 1. -Haar feature types

3.1.2 Integral image

The number of addition and subtraction operations required to calculate Haar features increases as the matrix size increases. Calculating the Haar features at various image sizes and positions takes considerable time. Consequently, a method was employed to facilitate the completion of this process. This method works by summing up the values of the array representing the pixels of the image and creating a new array with the sum values. This matrix makes it easy to calculate the sum of any subregion of the image. Each value in the integral image matrix is the total value in the top left corner of the matrix. Equation (1) is used to determine the remaining image elements.

$$I_{r,c} = \sum_{\substack{r' \le r \\ c' \le c}} i_{r',c'} \tag{1}$$

The pixel value at the r^{th} row and c^{th} column in the original image is represented by $I_{r,c}$, while the pixel value in the integral image is represented by $i_{r',c'}$. This method is essential for accelerating the calculation of an image's area sum. To calculate the 5 × 3 size of an image, for instance, we add two-pixel values and subtract two other factors instead of summing 15 elements, as depicted in FIGURE 2. For example, we must perform fifteen addition operations to determine the sum of the values within the red box.

30	39	48	1	10	19	28
38	47	7	9	18	27	29
46	6	8	17	26	35	37
5	14	16	25	34	36	45
13	15	24	33	42	44	4
21	23	32	41	43	3	12
22	31	40	49	2	11	20

a- Original image

30	69	117	118	128	147	175
68	154	209	219	247	293	350
114	206	269	296	350	431	525
119	225	304	356	444	561	700
132	253	356	441	571	732	875
153	297	432	558	731	895	1050
175	350	525	700	875	1050	1225

b- Integral image

FIGURE 2. -Implementation of integral image

Using the integral image, the sum of these values and those in the pink and blue dashed boxes equals 731. To find the sum of the red box's elements only, subtract the sum of the values in the pink box, which has a value of 128, and the sum of the values in the dashed blue box, which has a value of 297. However, because the pink box and blue dashed box share two-pixel values, we add the sum to avoid subtracting them twice. The sum value of the elements inside the red box is calculated using the fewest mathematical operations possible using the abovementioned method. 731 – 128 – 297 + 69 = 375; the sum of the red box's pixels equals 375. The value of the sum of a portion of the box is determined using the integral image, followed by the mean calculation. The sum and average of the second part are then calculated

and subtracted from the first value. The resulting value is compared to our threshold to determine whether or not this area represents the Haar feature.

3.1.3 Adaboost training algorithm

The features produced by calculating Haar features are numerous, exceeding 160,000 in number, and not all of these features impact classifier performance. As a result, the importance of the AdaBoost algorithm is emphasized. Freund and Schapire developed the Adaboost algorithm [33], which stands for Adaptive Boosting, a machine learning algorithm. It selects the features that significantly impact the algorithm and classifies each feature as a weak classifier. These weak classifiers come together to form what is known as a "strong classifier." The Adaboost algorithm's operation is summarized by providing the input data and output labels (i_1, o_1),…, (i_n, o_n). Where (i) is the input data and (o) is the output label. By specifying the size of the entered data (n), where labels represent a positive or negative value of whether or not the image contains a face, i.e., o = 0 or 1.

3.1.4 Cascade classifier

Even though the AdaBoost algorithm selected the features that had the most significant impact on face detection, it is still time-consuming to calculate all image subregions because the AdaBoost algorithm selected between 2500 and 3000 of a total of 16000 features. Consequently, using such an algorithm in real time face detection is not easy. Consequently, the cascade classifier is essential. AdaBoost calculates and passes the most significant image subregion features to cascade classifiers, each detecting whether the subregion contains a face. If the classifier predicts the presence of a face in that section, it progresses to the next classifier. Suppose no face is expected in the image's subregion. In that case, the current subregion is skipped, and the process is repeated on the remaining subregions, which reduces the time required to detect the face. In other words, the classification of these parameters is binary, so the subregion with a positive value concludes the series of classifications. In addition, the classifiers exclude the subregion with a negative value from the calculations, as illustrated in FIGURE 3.

Consequently, cascade classifiers drastically reduce the number of calculations performed on false windows. Two steps are required to implement the Viola-Jones algorithm. Input the images that will be used to train the algorithm before anything else. Next, the images should be converted to integral images. Calculate the Haar features utilizing these images. Various features are obtained with their output labels (face or not). The extracted Haar features can detect faces in a live video (or image) from a camera by training the algorithm with these features.

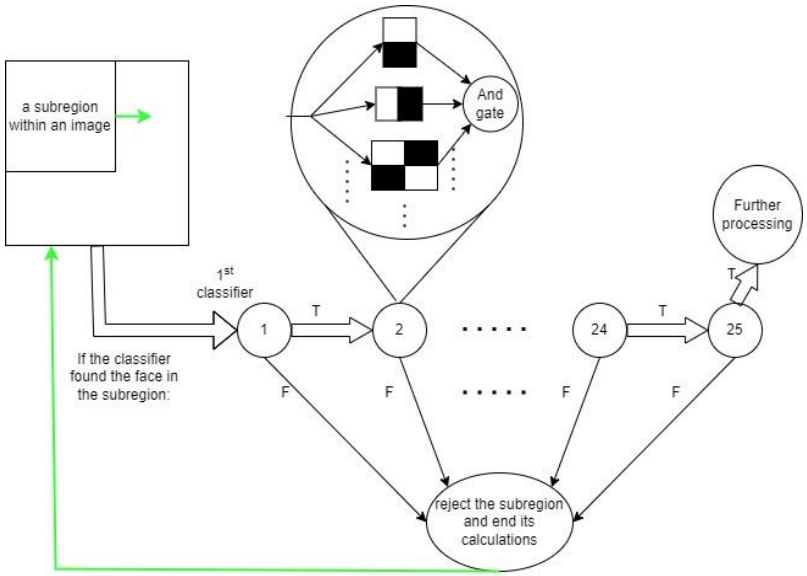
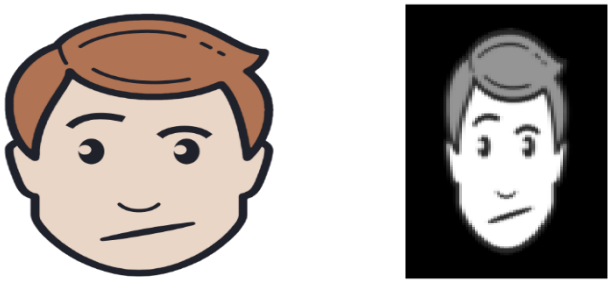


FIGURE 3. -Cascade classifiers

3.2 HISTOGRAM OF ORIENTED GRADIENTS (HOG)

The HOG algorithm is frequently applied as a feature descriptor in digital image processing and computer vision. It is considered one of the most effective edge-detection algorithms because it uses edge rotation and magnitude by dividing the image into multiple regions and generating histogram values for each region [34]. In contrast, other edge-detection algorithms determine whether or not a segment is an edge. This algorithm extracts features from image data in object detection systems. First, the image containing the object to be selected is inserted. It is then converted from RGB or BGR color space (depending on the image library type) to grayscale. Then the image is resized so that its width is double its height. The optimal size is 128 × 64 because, as described in the algorithm's original paper, for feature

extraction, the image is divided into 8 × 8 and 16 ×16 patches [35]. FIGURE 4a is a 513 × 513 pixel representation of a human face. The first step of the HOG algorithm is to resize and convert the image to grayscale, as depicted in FIGURE 4b. The gradients are calculated. The gradient is a slight change in pixel values that can be calculated using equations (2) and (3) [36].



***a*- Input image** ***b*- Resized and converted image**

FIGURE 4. Original and converted image

$$G_x(r,c) = I(r,c+1) - I(r,c-1) \tag{2}$$

$$G_y(r,c) = I(r-1,c) - I(r+1,c) \tag{3}$$

Using the same method to calculate the x- and y-gradients for all pixels yields two matrices, one for the x-gradients and one for the y-gradients. The magnitude and orientation of gradients to the centered pixel depicted in FIGURE 5 are then determined using (4) and (5), respectively [36]:

$$M = \sqrt{G_x^2 + G_y^2} = \sqrt{18^2 + (-16)^2} = 24.03 \tag{4}$$

$$\theta = \left| tan^{-1} \frac{G_y}{G_x} \right| = \left| tan^{-1} \frac{-16}{18} \right| = 41.63^{\circ} \tag{5}$$

30	39	48	1	10	19	28
38	47	7	9	18	27	29
46	6	8	17	26	35	37
5	14	16	25	34	36	45
13	15	24	33	42	44	4
21	23	32	41	43	3	12
22	31	40	49	2	11	20

FIGURE 5. -Centered pixel matrix.

The magnitude relation is obtained using the Pythagorean theorem, where the x- and y-gradients represent the base and height of a triangle. (M) is the magnitude, and (θ) is the angle of the gradient. The resulting matrices are divided into 8×8 cells to form a block. Each block receives a nine-point histogram [37]. This histogram is composed of nine bins. Performing the following calculations for each value in the 64 values in the block [38]:

$$n = 9 \tag{6}$$

$$\Delta\theta = 180 / n = 20^{\circ} \tag{7}$$

Where n is the number of bins ranging from 0 to 180 (it can be from 0-360, but the best results were with this range), its value is equal to 9 as the 9-point histogram was used, $\Delta\theta$ is the step size of the orientation of the gradient. Each i^{th} bin has the boundaries: $\left[\Delta\theta.i, \Delta\theta \cdot (i+1)\right]$. The value of the center of each bin is given by [38]:

$$C_i = \Delta\theta(i + 0.5) \tag{8}$$

then calculating the i^{th} and $(i+1)^{th}$ bin values. The following relations give these values [38]:

$$i = \left\lfloor \left(\frac{\theta}{\Delta\theta} - \frac{1}{2} \right) \right\rfloor \tag{9}$$

$$v_i = M \cdot \left(\frac{\theta}{d\theta} - \frac{1}{2} \right) \tag{10}$$

$$v_{i+1} = m \cdot \left(\frac{\theta - c_i}{\Delta\theta} \right) \tag{11}$$

After completing the histogram calculations for every block, as depicted in FIGURE 6, a $16 \times 8 \times 9$ matrix was produced. The unity of four blocks within the histogram matrix is then utilized to generate a new block. Concatenating the 9-point histogram of each cell in the four cells of each block into the four blocks yields a 36-feature vector. The resulting feature vector is highly sensitive to light. Suppose we attempt to adjust the lighting by multiplying or dividing the vector by a number; the magnitude of the vector changes relative to that number. In a process known as normalization, each value in the vector is therefore divided by the vector's norm. This type of normalization is known as L2 normalization.

The following relations are utilized to normalize the feature vector [38]:

$$N = \sqrt{f_1^2 + f_2^2 + \ldots + f_{36}^2} \tag{12}$$

$$F_{fj} = \left[\frac{f_1}{N}, \frac{f_2}{N}, \ldots, \frac{f_{36}}{N} \right] \tag{13}$$

Since there are seven blocks in the horizontal direction and 15 blocks in the vertical direction, the total number of features obtained in the HOG algorithm is 3780 ($36 \times 7 \times 15$), as illustrated in FIGURE 7.

After extracting HOG features, they must be classified according to whether they represent a human face. For this, a support vector machine algorithm is used.

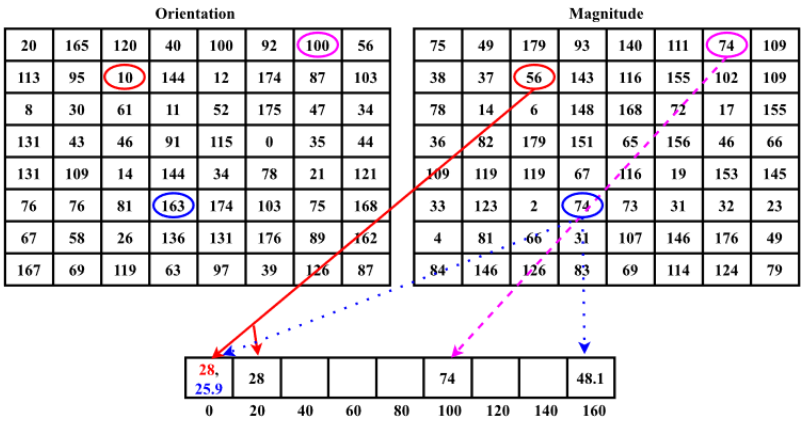


FIGURE 6. -HOG's calculation

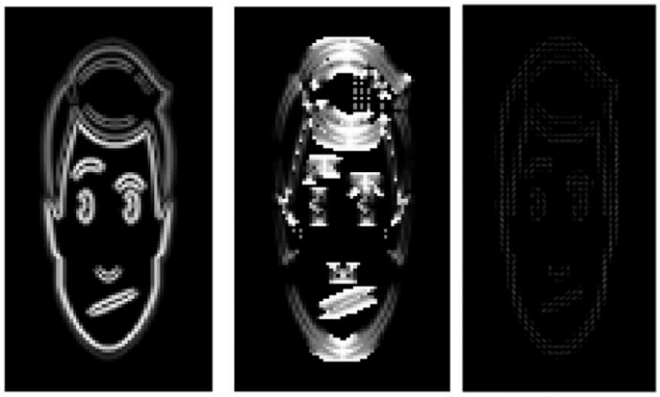


FIGURE 7. -Gradients magnitude, orientation, and HOG's visualization

3.3 MULTI CASCADE CONVOLUTIONAL NEURAL NETWORK (MTCNN)

It is among the most accurate face detection algorithms available. It is divided into three stages, which are as follows: The first stage entails identifying objects that are likely to be human faces by making multiple copies of the same image in different sizes. If the image contains a human face, each image has a different human face size. The generated image is known as the pyramid [39]. Each image is divided into 12 × 12 regions, and a kernel is run over the entire image to look for faces. The search starts at the top-left corner (0,0), where the kernel moves with a strain equal to 2 until it reaches the point (0+2a,0+2b) to find a face in the image. This process is repeated towards the image's right or bottom, and the face's presence is determined upon completion. Once the search is complete, it is determined whether or not a face is present in the image. If the image is confirmed to contain a face, it is sent to the second stage to determine the bounding box coordinates. Reduce the strain value to 2 to simplify the mathematical operations that must be performed while maintaining accuracy. The algorithm is unlikely to miss one of the human faces due to the strain value of 2 [40] because the human face is certain to be larger than two pixels in the image.

Moreover, the number of calculations the computer must perform is reduced by two, reducing RAM usage and accelerating program execution; thus, this value was chosen for stride. A small kernel is required to identify small faces within a large image. In addition, the kernel must be large in small images in order to recognize large faces. After passing the image to the algorithm, it must be enlarged and shrunk to produce images of different sizes, which are then passed to the proposal network, also known as the P-net, the initial stage of the algorithm. After training the network's weights and biases, each kernel's bounding box is obtained with reasonable precision. However, this network's confidence within the boxes varies, with some having a high level of confidence and others having a low level of confidence. Therefore, the bounding boxes must be refined to obtain a list of confidence values for each box, and boxes with low confidence must be eliminated (eliminating boxes with potentially no human face). It is necessary to standardize the coordinate systems by converting all coordinates to the original (unscaled) image coordinates after identifying the most reliable bounding boxes. Despite these efforts, the image still contains numerous bounding boxes

that overlap. Non-Maximum Suppression, or NMS for short, is employed to reduce the quantity of these boxes. NMS uses the most accurate boxes to identify faces, calculating the areas of the boxes and each kernel, calculating the overlap area between each kernel and the most accurate kernel, and removing boxes that overlap with the most accurate bounding box. It returns a list of the remaining bounding boxes. To constrain the boxes to a single, precise bounding box for each face, one NMS is applied to each scaled image and one to the kernels derived from all the scaled images. This technique must be used because taking the box with tremendous confidence is insufficient, as the images may contain more than one face.

Consequently, if only the most accurate box is accepted, all boxes surrounding the other faces will be removed, leaving only the most accurate box. Then, because the coordinates of the surrounding boxes are between zero and one, converting them to the actual image's coordinates, beginning with the point (0,0), which represents the kernel's upper-left corner, and ending with the point (1,1), which represents the kernel's lower-right corner. By multiplying the surrounding boxes' coordinates with the dimensions of the actual image, the coordinates of the bounding boxes are converted to the coordinates of the actual image. If the surrounding boxes are not square at the end of this process, they must be reshaped by lengthening the shorter part; if the box's width is less than its height, it is expanded horizontally [41]. Even as its height decreases, the box expands vertically. Finally, the surrounding box coordinates are saved and sent to the subsequent stage. Some images may have a human face projecting from one side.

In this instance, the generated P-Net's bounding box includes portions of the image side frame. As in the person's box on the right of FIGURE 8. An array is created for each bounding box to solve this problem. The pixel values within the bounding box (the image) are copied to the new array. If the box extends beyond the image's borders, the values within the frame are copied, and the remaining box values are set to zero. The process of adding zeros is called padding. After filling the bounding box matrices, they are scaled to 24 × 24 pixels and normalized between -1 and 1. In the current image format, the values range from 0 to 255, indicating that it is in RGB color mode. Each pixel is calculated by subtracting 127.5, half of 255, and dividing by 127.5, which produces values between negative and positive. This procedure generates a large number of 24 × 24 matrices (as the number of boxes left over from the P-Net), which can then be inserted into the R-Net to obtain its output. The outputs of the R network, also known as the refined network, are very similar to the outcomes of the P network. It contains the coordinates and confidence level for each of the new bounding boxes with a high confidence level. The NMS technique eliminates low-accuracy surplus boxes. After that, the coordinates must be reset to their initial values. After standardizing the coordinates, the bounding boxes are reshaped into squares, and the results are sent to the third stage (O-Net).

The third stage yields different outcomes than the previous two. Three outputs exist at this point. The first is the coordinates of the bounding boxes, and the second is the key points of the human face. This algorithm utilizes five points: eyes, nose, and mouth corners. The final output of this stage is each briefing box's confidence level [42]. Then, remove any redundant boxes and standardize the coordinates. At this point, each face should contain a single box. After completing this step, a dictionary will be received with three keys: 'Box,' 'Confidence,' and 'Points.'

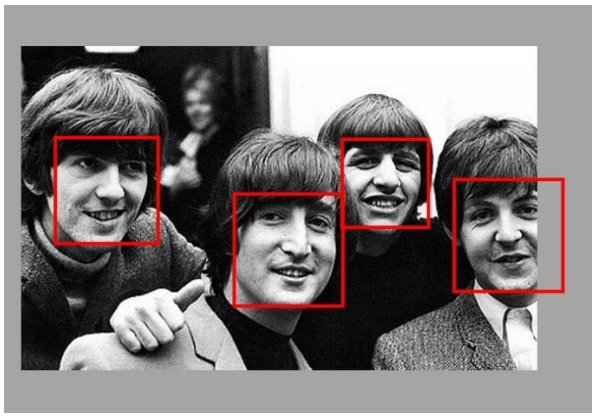


FIGURE 8. -The bounding box contains side frame portions [23]

4. FACE RECOGNITION

After face detection, pixel values representing human facial features are extracted. We used the following feature extraction algorithms.

4.1 LOCAL BINARY PATTERN (LBP)

LBP is a simple, high-precision texture operator that addresses pixels by thresholding the neighboring values and converting the result to a binary number. This algorithm is distinguished by its resistance to changes in image lighting and lack of implementation-required mathematical calculations [43]. Face recognition can utilize the LBP because it

describes visual features. The dataset must be prepared methodically before applying feature extraction using the LBP, with images of each individual collected in a specific folder and assigned a unique identifier. The same applies to the remainder of the people's images, which are organized in separate folders to give them distinct identities.

Before using this algorithm, the image is converted to grayscale. For each pixel value (gp), The number of pixels (N) that surround the central pixel is determined, and the pixel value coordinates are calculated using (15) [15].

$$\left(C_x - Rsin\left(2\pi p / N\right), C_y + Rcos\left(2\pi p / N\right)\right) \tag{14}$$

The value at the center (C) is the threshold value for the pixels around it. If the pixel's value is greater than the value of the central pixel, it is evaluated as one; otherwise, it is evaluated as zero. The LBP is then computed by encircling the central pixel with binary numbers and calculating the corresponding decimal number.

The uniform LBP is given by (16) [15]

$$LBP\left(gp_x, gp_y\right) = \sum_{p=0}^{N-1} F\left(gp - C\right) \times 2^p, \tag{15}$$

Where (gp) represents the current pixel value, N is the number of circular neighborhoods, R represents the spatial resolution of the operator, p = 0, 1, 2, …, (N), (C) is the central pixel value. F is a function, and it can be expressed as [15]:

$$F(x) = \begin{cases} 1 \text{ if } x \geq 0 \\ 0 \text{ if } x < 0 \end{cases} \tag{16}$$

In FIGURE 9, for instance, we can see that the center of the window is at the pixel with the value 16. The surrounding pixels have binary values (either zero or one). Where zero is assigned to values less than the central pixel and one is assigned to values greater than the central pixel, the LBP for that window is calculated from the image as follows:

$$(0\times 2^7)+(0\times 2^6)+(1\times 2^5)+(0\times 2^4)+(1\times 2^3)+(0\times 2^2)+(1\times 2^1)+(1\times 2^0)$$
$$0+0+32+0+8+0+2+1=43$$

We have a new image with distinguishable characteristics. A histogram is used to represent this image. One method is used to differentiate between two histograms for face recognition. Euclidian distances, chi-square, and absolute value are the most prevalent methods [44]. If Euclidean distance is used, it is determined using (18)

$$O = \sum_{i=0}^{n} (hist1_i - hist2_i)^2 \tag{17}$$

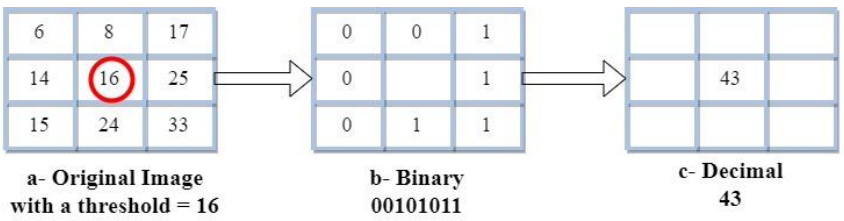


FIGURE 9. -LBP calculation

The algorithm's output is the identity of the person whose histogram value is closest to the histogram of the current image. Additionally, the algorithm produces a confidence value for its performance. The performance of this algorithm is more accurate as confidence decreases because the distance between the two histograms decreases (the similarity between them increases).

4.2 FACENET

Researchers from Google developed this facial recognition system which can generate high-resolution face maps using deep learning architectures such as the ZF Network and the Inception Network [45], training the structure with the triplet loss function. FIGURE 10 depicts the algorithm's construction.

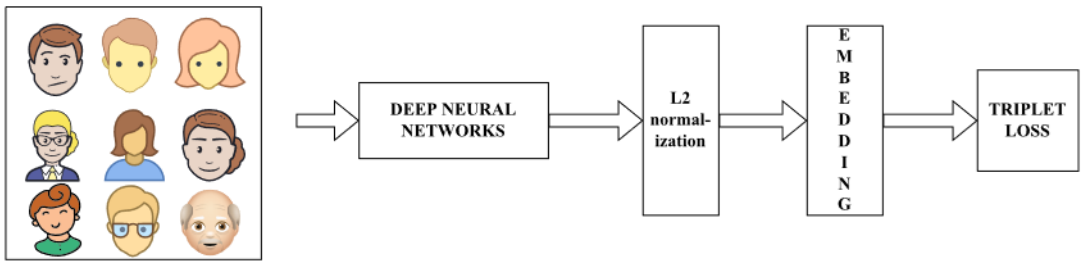


FIGURE 10. -Facenet steps

This system's primary backend is one of the two architectures (ZF or Inception). Several one-dimensional convolutional layers are added to these networks to reduce the number of parameters. These deep learning models generate an image embedding f(x) for which L2 normalization is performed. The loss is computed by passing these embeddings to the loss function [23]. The loss function's objective is to minimize the squared distance between two images of the same identity (images of the same student) while increasing the distance between images of different students. Consequently, this system uses the triplet loss as the loss function. The triple loss aids in separating the faces of various individuals.

Triplet loss: The image embedding in triplet loss is denoted by E(x), where $x \in$. A 128-dimensional vector represents this embedding [46]. This vector is normalized to satisfy the condition in (19)

$$\left\|E(x)\right\|_2^2 = 1 \tag{18}$$

The anchor image of a person $E(x_i^a)$ must be closer to the positive image (the image of that person) $E(x_i^p)$ than to the negative image (the images of other people) $E(x_i^N)$, as in (20) & (21) [46]

$$\left\|E\left(x_i^a\right)-E\left(x_i^P\right)\right\|_2^2+\alpha<\left\|E\left(x_i^a\right)-E\left(x_i^N\right)\right\|_2^2 \forall\left(E\left(x_i^a\right),E\left(x_i^P\right),E\left(x_i^N\right)\right)\in T \tag{19}$$

$$L=\sum_i^n\left[\left\|E\left(x_i^a\right)-E\left(x_i^p\right)\right\|_2^2-\left\|E\left(x_i^a\right)-E\left(x_i^N\right)\right\|_2^2+\alpha\right] \tag{20}$$

Where x is the image, E(x) is the image's embeddings.

If only triplets that satisfy the abovementioned equations are selected, these values are not very useful for training the model. To create a robust model, triplets that deviate from the abovementioned equations must be chosen.

Triplet selection: To expedite learning, triplets that violate the above equation are chosen. The highest possible $E(x_i^p)$ values and the lowest possible $E(x_i^N)$ values are selected for each $E(x_i^a)$ [47]. Because completing this procedure on all datasets is mathematically tricky, the triplets are chosen in one of two ways:

Similar to the training for linear regression, triplets are generated for each step based on the previous checkpoints. The maximum and minimum values are calculated on a subset of the data [48].

Fixed positive $E(x_i^N)$ and negative $E(x_i^p)$ values are determined using a small number of images.

4.3 DLIB'S RESNET-BASED FACE RECOGNITION MODEL

The ResNet-34 architecture serves as the foundation for this model. This model is trained with the support of three images. The first image is called the "anchor," and utilize it. The second image is the "positive" and depicts the same individual as the anchor image. The final image, known as the "negative," differs from the anchor image.

These networks generate a 128-dimensional embedding for each image. It also bases its training on the loss function, which aims to minimize the distance between the positive image and the anchor image (images of similar people) as much as possible while maximizing the distance between the anchor image and the negative image (photos of different people).

After training the network, the algorithm can generate embeddings for various images, which can then distinguish between the images provided. FIGURE 11 illustrates the principle of this model where the anchor is illustrated in the

middle image. The image on the right is the positive image, while the image on the left is the negative image. Since the negative image is further from the anchor image, the embeddings produced by the neural network of the positive image are closer to the anchor image.

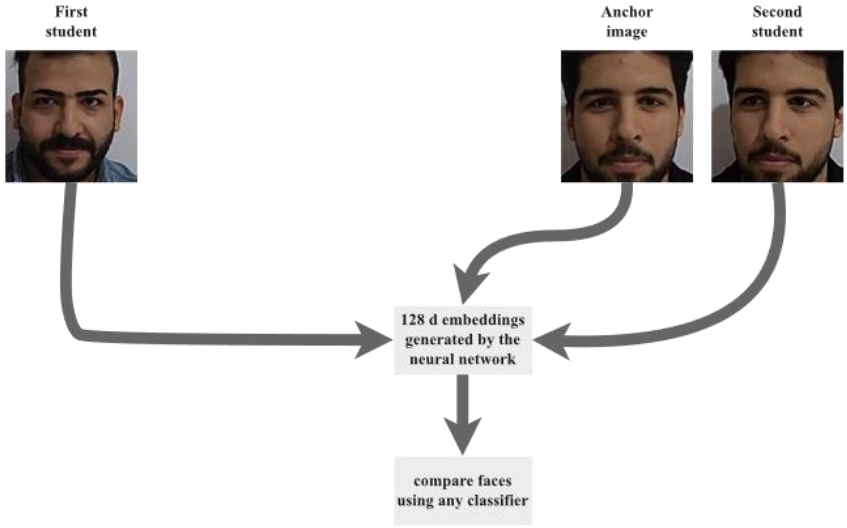


FIGURE 11. -Face recognition in Dlib

5. Materials and method

5.1 MATERIALS

A Hikvision camera and a computer are required to implement the proposed recording attendance system. The Hikvision bullet network camera, model DS-2CD2T60-I4, with a 2.8mm focal length, is installed in the students' lecture hall in front of the students and in the center of the hall to photograph all of the students present, without the use of microcontrollers, an internet connection, or any other requirements that raise the project's cost and complexity. An Intel(R) Core (TM) i7-7500U CPU @ 2.70GHz 2.90 GHz and 16 GB RAM computer with Windows 10 operating system were used to program the system.

A desktop application was created using the Tkinter library, which is used to design graphical user interfaces and is integrated with Python. This application handled images and videos using OpenCV, a computer vision library. Furthermore, this application uses the OpenCV, Scikit-learn, Keras, and TensorFlow libraries for reading and segmenting images and training algorithms to read and classify them. The first application window is the login window, depicted in FIGURE 12.

This window improves system security by preventing unauthorized users from accessing and modifying the system's information. By clicking the "add new user" button, you can grant permission to any of the unauthorized members. An admin registration form appears. This form contains all the necessary information about the member who requires administrative privileges. This form includes voice outputs to help explain any problems with the registration process. It also includes a password neither party knows, so the person in charge of this system must strengthen the security. After successfully logging in, the program's main window is displayed. FIGURE 13 depicts this window.

FIGURE 12. -The login window

FIGURE 13. -The main window of the program

5.2 DATASET COLLECTION

After the person responsible for collecting student data registers with a username and password, the program's main interface display to him; this window consists of six buttons, each leading to a window that performs a specific function. FIGURE 14 depicts the enrollment window when clicking the "Student Information" button. Clicking the "Take Pictures" button opens the window for taking pictures, and obtaining students' images commences.

The proposed system acquires images by logging into a camera using the Real Time Streaming Protocol (RTSP) and opening a separate window displaying a live camera image. The course students' information is then entered so that images can be taken of them. Instead of using a counter to take a certain number of photos, pressing a key on a wireless keyboard to take photos has been adopted because it provides greater control over the process. In the counter method, a different individual may enter the camera's field of view and be photographed in place of the required individual. Moreover, the photos must include a variety of facial expressions, such as happiness, sadness, laughter, nervousness, and surprise, necessitating the direction of the individual prior to taking the photographs. In addition, images were collected in batches. Each batch is distinguished from the next by a unique distance. The initial group was positioned one meter away, followed by two meters, four meters, eight meters, and finally, ten meters.

The Q button on a wireless keyboard is pressed after several photos have been taken, sending them to the Viola-Jones algorithm, which searches for the face in the images and crops it. The cropped faces are saved in the dataset folder within the work folder in JPG format with a different folder name associated with the student enrollment number. Students' pictures must be taken in the same hall so that the lighting and background match the images of the other students. This process is repeated for all students to obtain the data set. FIGURE 15 shows an example of the faces stored in the faces dataset used in this paper. The final faces dataset contains 2170 images, with 70 images for each student for 31 students of the faculty of science at Mustansiriyah university (the dataset is available for researchers

on Kaggle, using this link: https://doi.org/10.34740/kaggle/dsv/4925762). The authors obtained permission from the students whose images appear in this dataset.

FIGURE 14. -Registration window

FIGURE 15. -Portion of the face images in the dataset

5.3 ATTENDANCE PROCESS

Several steps were involved in the attendance procedure for the conducted attendance system, including the system's opening, the capture and recognition of faces, and the handling of attendance. FIGURE 16 depicts the flowchart for the conducted attendance system, and the specifics are described below.

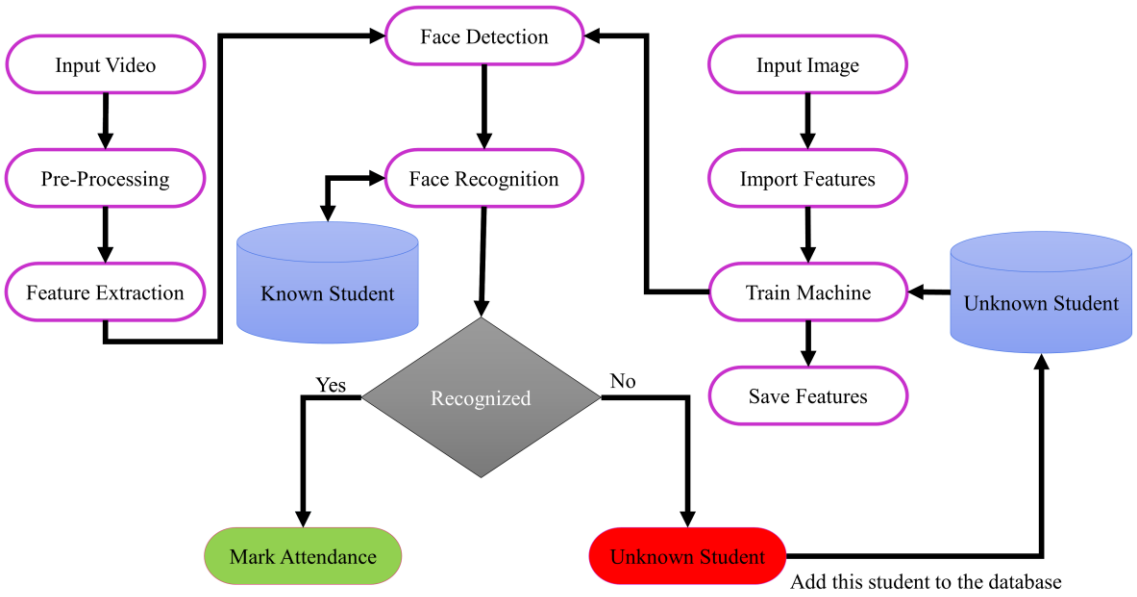


FIGURE 16. -The flowchart of the proposed system

5.3.1 ATTENDANCE RECORDING WINDOW

The window shown in FIGURE 17 opens when the user clicks the "Human Face Detector" button from the program's main window depicted in FIGURE 13. It displays two menu boxes, one for choosing the stage and one for choosing the subject. The person in charge of registering attendance must select them after logging into the system; otherwise, a message appears informing him/her that he/she must select the stage and the subject. When that person clicks the "Record Attendance" button, an audio commentary instructs the students to pay attention to the camera for their attendance to be recorded.

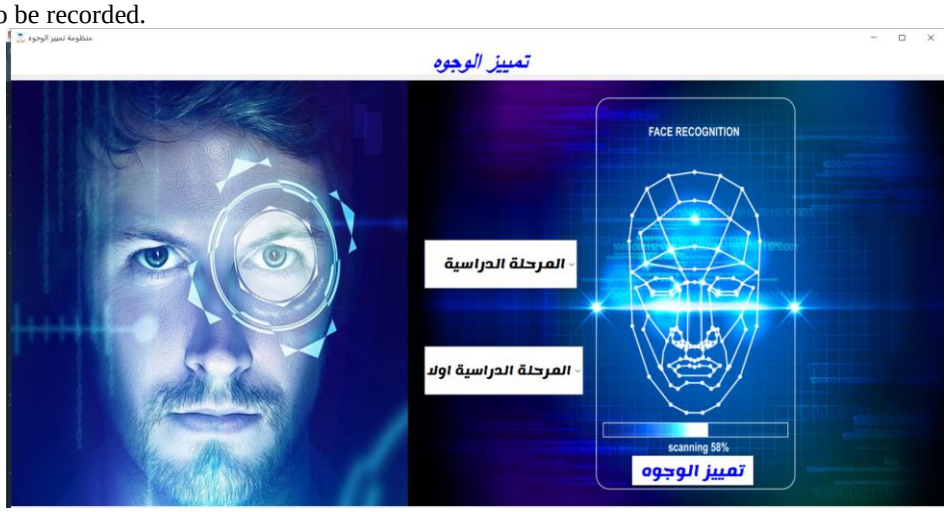


FIGURE 17. -Attendance registration window

5.3.2 CSV FILE GENERATION

When the system receives the stage and subject data, it generates a comma-separated file (CSV) with the following columns labeled: registration number (the number assigned to the student by the college registration staff), student name, department, time of attendance, date of attendance, attendance status and the subject. This file is ready to be filled out following the face recognition process.

5.3.3 ATTENDANCE MARKING

When the face recognition process is finished, the program searches the MySQL database for the recognized face and its entire set of data, which is then filled in using the previously created CSV file. As a result, all students who

attended that lecture or seminar are listed. And a green rectangle around these students' faces, with their registration numbers above it. For students not registered in the database, such as professors, the face detection algorithm draws a red rectangle around their faces and displays the phrase "unknown" above each. FIGURE 18 shows an example of the generated CSV file.

One of the essential features of the proposed system is that it saves the CSV file with the following title: [phase] [article] date@time.csv, as depicted in FIGURE 19. Then it saves it in a folder called "attendance," which makes it easier to find out which students were in class for a particular subject on a specific date.

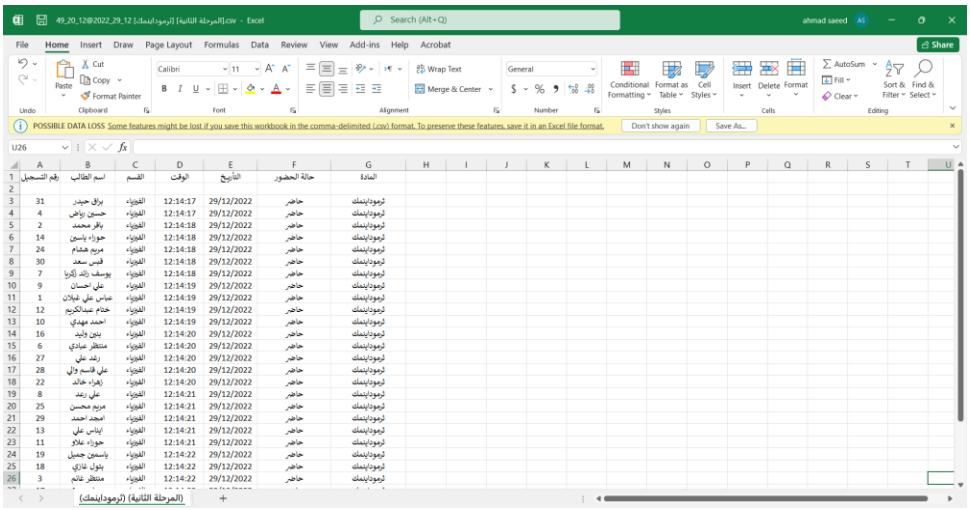


FIGURE 18. -Attendance report

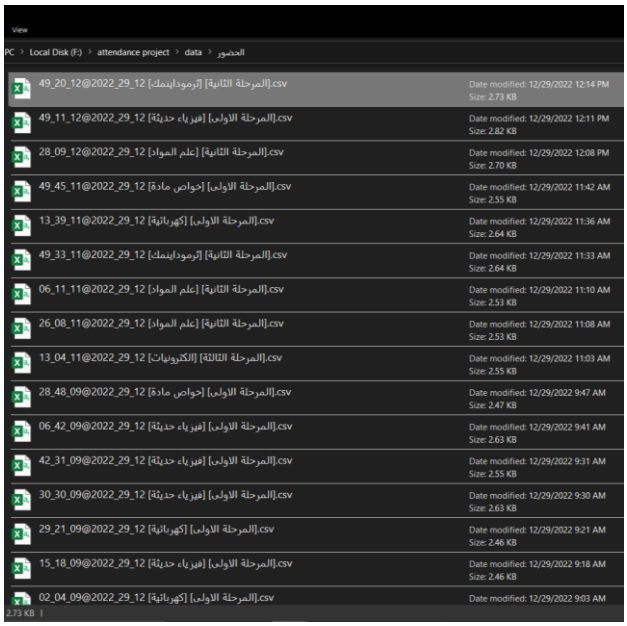


FIGURE 19. -The titles of the generated CSV files

5.4 PERFORMANCE EVALUATION

The Mustansiriyah University College of Science sent an official invitation to students, and their information was collected. Their images were obtained using a camera installed in one of the college's classrooms. The test was then administered to the students using three techniques. The first detects faces using the Viola-Jones algorithm and recognizes them using LBPH; the second detects faces using the HOG algorithm and recognizes them using Dlib

encodings; and the last technique detects faces using MTCNN and recognizes them using Facenet. The following terms must be defined to comprehend the results obtained:

True positive (TP) is a student whom the algorithm correctly identifies.

A false positive (FP) is a student the algorithm incorrectly identifies as someone else.

True negative (TN) is a student not in the database whose identities can be identified by the algorithm as "unknown persons."

False negatives (FN) are people whose faces the algorithm cannot recognize or who are marked as unknown despite being in the database [49].

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \times 100\% \tag{21}$$

$$Recall = \frac{TP}{TP + FN} \times 100\% \tag{22}$$

$$Precision = \frac{TP}{TP + FP} \times 100\% \tag{23}$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \times 100\% \tag{24}$$

6. RESULTS AND DISCUSSION

The video is a series of successive images known as frames. Face recognition algorithms process each of these frames separately and return the results in real-time on the video.

The proposed method faces numerous challenges. For example, restrictions to face vision, such as masks and the distance between the camera and the person, increase the likelihood that the algorithm fails to produce good results. As a result, most previous studies relied on grouping students into one or two lines of no more than ten people. Because the distance between people and the camera is fixed and does not exceed 5 meters, they obtained high-quality results. The same thing happens when the face is turned away from the camera. This study tested 31 students using live video streaming from a classroom during a lecture to mimic classroom conditions. This video includes tilted faces and various facial expressions, as well as the various distances of the students from the camera. The study includes students positioned at a distance from the camera between 10-32 feet, which is believed to be the first research to simulate these conditions with this number of students.

6.1 MTCNN AND FACENET

This method's results can be obtained from FIGURE 20. The students were labeled with numbers to avoid confusion in the image due to their names. The proposed method achieved 100% accuracy. One can note that students 21, 6, and 8 sit at a distance of more than 10 meters apart, and the method could easily distinguish their faces. The algorithm recognized students 19 and 24's faces even with their eyes closed. In FIGURE 21, student 9 bends his head downward and places his hand on his nose, obscuring most of his face, but the algorithm still recognizes him. The proposed method's accuracy helps to solve traditional system problems. It outperforms the accuracy obtained from other studies, such as [25], which approved an accuracy of 92% by testing the same method on single individuals, and [45], which achieved an accuracy reached 99.85%.

FIGURE 20. -Facenet performance in a lecture

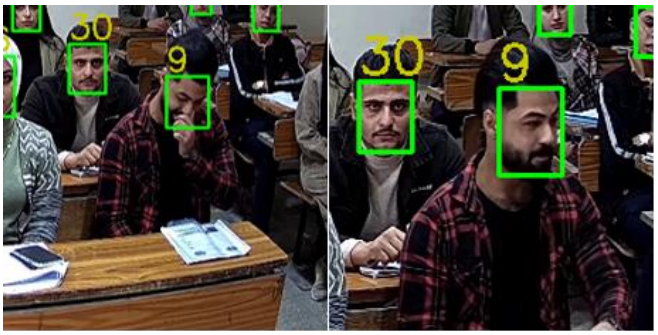


FIGURE 21. -Facenet performance on concealed and angled faces

6.2 HOG AND DLIB

The only disadvantage of this method is that it cannot detect faces far from the camera. FIGURE 22 depicts its inability to distinguish the faces of students 6 and 21, the last two students in the right column, due to the HOG algorithm's inability to detect their faces and send the location of the pixels that represent the face to the ResNet neural network to generate embeddings and compare them to the faces in the database. Therefore, the face verification technique was used as a second step to verify the person in [6] using the HOG algorithm to identify the face and one of the classifiers provided by the OpenCV library to distinguish faces, on the condition that the person is very close to the camera. However, the high ability of this method to generate correct embeddings for the faces used to generate the embeddings distinguishes it. When used in real-time to distinguish faces, it almost eliminates FPs, and as shown in Table 1, its accuracy reaches 94%. This accuracy is acceptable in real-time and educational environments compared to other studies that used this method, such as *[50]*, which faces a problem when there are 2 or 3 people in front of one camera and has a minimum distance of 2 meters to detect faces.

FIGURE 22. -Dlib performance in a lecture

6.3 VIOLA-JONES ALGORITHM AND LBPH

This method suffers from two problems. The Viola-Jones algorithm was used to locate faces, but it was unable to identify the face of a female student wearing black clothes and sitting in the second row on the right in FIGURE 23. This is because this algorithm cannot identify the faces of people who are not looking directly at the camera. The other problem is how it extracts the features used to classify the images. It can be seen that there are two false positive values that the algorithm has classified with identities that are not their real identities. One of them is the lecturer, who stands between the left and the column in the middle, and the other is the student, who sits in the left column in the second line and is classified as a student with identity number 12, but her real identity is number 23. Despite these problems, the results are significant compared to other studies that used LBPH in face recognition applications. For example, [9] and [12], which relied on a small data set of no more than ten students at very close distances from the camera, the obtained accuracy can be enhanced by increasing the quality and quantity of images for each class in the dataset or by using various image processing techniques such as histogram equalization [51].

FIGURE 23. -The performance of LBPH

Table 1. – Classification results

Technique	TP	TN	FP	FN	Accuracy	Recall	Precision	F1-Score
MTCNN + Facenet	31	1	0	0	100%	100%	100%	100%
HOG + Dlib	29	1	0	2	94%	94%	100%	97%
Viola-Jones +LBPH	29	0	2	1	91%	97%	94%	96%

Table 2. – Comparison of the proposed method with the results of some related work

Ref.	Method	Year	No. of examined faces	Accuracy
[9]	Viola-Jones + LBPH	2020	11 not in real time and just one person in real time	96%
[24]	MTCNN + VGG19	2021	25	80-90%
[50]	Viola-Jones + Dlib	2020	3	88%
[52]	PCA + SOM	2022	Lacks real-time implementation	94%
[53]	Viola-Jones + LBPH	2023	1	94%
The Proposed Methods	MTCNN + Facenet	2023	32	100%
	HOG + Dlib			94%
	Viola-Jones +LBPH			91%

Results in Table 1 were calculated using equations (22) - (25). According to Table 1, it can be noted that the first method obtained the ideal results because it relied on the transfer learning technique, where the results of a pre-trained model were used on a massive amount of data and used to extract the features of the images of people in the database of this study. Therefore, when face recognition algorithms are used in an application such as attendance registration, the algorithms must be trained on many images. The second study relied on identifying faces with the machine learning technique, which could not identify the faces of two students and, thus, extract their features or classify them. Moreover, the third method relied entirely on machine learning, which resulted in two false positives and one false negative, where the results of a pre-trained model were used on a massive amount of data to extract the features of the images of people in the database of this study. Therefore, when face recognition algorithms are used in an application such as attendance registration, the algorithms must be trained on many images. The second study relied on identifying faces with the machine learning technique, which could not identify the faces of two students and, thus, extract their features or classify them.

Furthermore, the third method relied entirely on machine learning, which resulted in two false positives and one false negative. Comparing the results of this study with those of similar studies, our study's results are promising, as depicted in Table 2. They can automate the registration process for students at universities and other institutions.

7. CONCLUSION

This paper proposes three facial recognition techniques for a completely automated attendance recording system. The first method used deep learning to recognize and differentiate faces. While the second technique utilized a combination of machine learning and deep learning, the first technique was used to detect faces, whereas the second technique was used to differentiate them. This paper identified and distinguished faces using machine learning in the third technique. The three approaches have achieved a high degree of accuracy and eliminated the problems associated with conventional methods of recording attendance. The primary objective of this paper is to devise reliable ways to automate the attendance registration procedure, which benefits educational and governmental institutions. The first technique had a precision of 100%, while the other two had more than 94% precision. The proposed method was tested on 31 students positioned at different distances from the camera and in classrooms with various conditions. Collecting a more extensive and precise database can significantly improve the accuracy obtained. Although the system achieves great accuracy, one of its limitations is the number of students used to conduct this study. Only 31 were examined. The accuracy improves when the distance separating the camera and the students is consistent and close to the camera, as the image becomes blurry when the distance increases. In some instances, the system produces false results for blurry faces, so different distances that differed from one student to the next were used. The proposed method employed relatively long distances. This study used a maximum distance of 10 meters (32.8 feet) to represent the dimensions of the classroom where the examination took place. Another factor to consider is the hall lighting, which should be medium. High light has the same negative effect as low light, affecting the features extracted from the human face. Lighting and pose variations significantly impact the performance of face recognition algorithms. Face recognition systems can have difficulty identifying individuals in low light conditions or when their face is not directly facing the camera. However, the results thus far have been very encouraging and promising.

In the future, we suggest enhancing these algorithms and linking them to the teachers' mobile phones instead of using cameras to eliminate the distance issue. To improve the algorithm, it is vital to consider oblique faces, changes in facial hair, and the use of glasses and accessories. In addition, the Internet of Things transmits attendance information to cloud storage rather than storing it on hard drives, utilizing these algorithms to determine students' emotions during lectures.

FUNDING

None.

ACKNOWLEDGEMENT

The authors would like to thank Mustansiriyah University, Baghdad, Iraq, for supporting the presented work. They also thank the reviewers for their valuable contribution in the publication of this paper.

CONFLICTS OF INTEREST

The author declares no conflict of interest.

REFERENCES

[1] S. B. Ahmed, S. F. Ali, J. Ahmad, M. Adnan, and M. M. Fraz, "On the frontiers of pose invariant face recognition: a review," *Artificial Intelligence Review*, vol. 53, no. 4, pp. 2571-2634, 2019, doi: https://doi.org/ 10.1007/s10462-019-09742-3.
[2] A. Aggarwal, M. Alshehri, M. Kumar, P. Sharma, O. Alfarraj, and V. Deep, "Principal component analysis, hidden Markov model, and artificial neural network inspired techniques to recognize faces," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 9, 2020, doi: https://doi.org/ 10.1002/cpe.6157.
[3] B. Xu, X. Li, H. Liang, and Y. Li, "Application research of personnel attendance technology based on multimedia video data processing," *Multimedia Tools and Applications*, 2019, doi: https://doi.org/ 10.1007/s11042-019-7227-y.
[4] A. Generosi, T. Agostinelli, and M. Mengoni, "Smart retrofitting for human factors: a face recognition-based system proposal," *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 2022, doi: https://doi.org/ 10.1007/s12008-022-01035-4.
[5] N. Nordin and N. H. Mohd Fauzi, "A Web-Based Mobile Attendance System with Facial Recognition Feature," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 14, no. 05, 2020, doi: https://doi.org/ 10.3991/ijim.v14i05.13311.
[6] E. Al Hajri, F. Hafeez, and A. A. N V, "Fully Automated Classroom Attendance System," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 13, no. 08, 2019, doi: https://doi.org/ 10.3991/ijim.v13i08.10100.
[7] A. Kumar, S. Jain, and M. Kumar, "Face and gait biometrics authentication system based on simplified deep neural networks," *International Journal of Information Technology*, 2022/09/11 2022, doi: https://doi.org/ 10.1007/s41870-022-01087-5.
[8] R. H. Farouk, H. Mohsen, and Y. M. A. El-Latif, "A Proposed Biometric Technique for Improving Iris Recognition," *International Journal of Computational Intelligence Systems*, vol. 15, no. 1, 2022, doi: https://doi.org/ 10.1007/s44196-022-00135-z.
[9] D. Bhavana *et al.*, "Computer vision based classroom attendance management system-with speech output using LBPH algorithm," *International Journal of Speech Technology*, vol. 23, no. 4, pp. 779-787, 2020, doi: https://doi.org/ 10.1007/s10772-020-09739-2.
[10] P. Viola and M. J. Jones, "Robust Real-Time Face Detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137-154, 2004/05/01 2004, doi: https://doi.org/ 10.1023/B:VISI.0000013087.49260.fb.
[11] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, 8-14 Dec. 2001 2001, vol. 1, pp. I-I, doi: https://doi.org/ 10.1109/CVPR.2001.990517.
[12] Y. W. M. Yusof, M. M. Nasir, K. A. Othman, S. I. Suliman, S. Shahbudin, and R. Mohamad, "Real-time internet based attendance using face recognition system," *International Journal of Engineering & Technology*, vol. 7, no. 3.15, pp. 174-178, 2018.
[13] Y. P. Jia, K. Y. Low, and H. L. Wong, "Development of a multi-client student attendance monitoring system," *International Journal of Recent Technology and Engineering*, vol. 8, no. 3S, pp. 6-11, 2019.
[14] M. Jahangir Alam, T. Chowdhury, and M. Shahzahan Ali, "A smart login system using face detection and recognition by ORB algorithm," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 20, no. 2, 2020, doi: https://doi.org/ 10.11591/ijeecs.v20.i2.pp1078-1087.
[15] Gnanaprakash, P. Harini, K. Harish, G. Suresh, and P. Purusothaman, "Smart Attendance System Using Cd-Lbp Algorithm," *International Journal of Scientific & Technology Research*, vol. 9, pp. 384-388, 2020.
[16] D. Sunaryono, J. Siswantoro, and R. Anggoro, "An android based course attendance system using face recognition," *Journal of King Saud University - Computer and Information Sciences*, vol. 33, no. 3, pp. 304-312, 2021, doi: https://doi.org/ 10.1016/j.jksuci.2019.01.006.
[17] V. Wati, K. Kusrini, H. Al Fatta, and N. Kapoor, "Security of facial biometric authentication for attendance system," *Multimedia Tools and Applications*, vol. 80, no. 15, pp. 23625-23646, 2021, doi: https://doi.org/ 10.1007/s11042-020-10246-4.
[18] P. R. Sarkar, D. Mishra, and G. R. K. S. Subhramanyam, "Automatic Attendance System Using Deep Learning Framework," in *Machine Intelligence and Signal Analysis*, (Advances in Intelligent Systems and Computing, 2019, ch. Chapter 29, pp. 335-346.
[19] R. Ullah *et al.*, "A Real-Time Framework for Human Face Detection and Recognition in CCTV Images," *Mathematical Problems in Engineering*, vol. 2022, pp. 1-12, 2022, doi: https://doi.org/ 10.1155/2022/3276704.
[20] G. R. Naufal, R. Kumala, R. Martin, I. T. A. Amani, and W. Budiharto, "Deep learning-based face recognition system for attendance system," *ICIC Express Lett. Part B Appl*, vol. 12, pp. 193-199, 2021.
[21] H. Pranoto and O. Kusumawardani, "Real-time Triplet Loss Embedding Face Recognition for Authentication Student Attendance Records System Framework," *JOIV: International Journal on Informatics Visualization*, vol. 5, no. 2, pp. 150-155, 2021.

[22] R. Alimuin, E. Dadios, J. Dayao, and S. Arenas, "Deep hypersphere embedding for real-time face recognition," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 18, no. 3, 2020, doi: https://doi.org/ 10.12928/telkomnika.v18i3.14787.
[23] F. M. Andiani and B. Soewito, "Face recognition for work attendance using multitask convolutional neural network (MTCNN) and pre-trained facenet," *ICIC Express Lett*, vol. 15, no. 1, 2021.
[24] T. Li, "Research on intelligent classroom attendance management based on feature recognition," *Journal of Ambient Intelligence and Humanized Computing*, 2021, doi: https://doi.org/ 10.1007/s12652-021-03042-x.
[25] N. T. Son *et al.*, "Implementing CCTV-Based Attendance Taking Support System Using Deep Face Recognition: A Case Study at FPT Polytechnic College," *Symmetry*, vol. 12, no. 2, 2020, doi: https://doi.org/ 10.3390/sym12020307.
[26] M. Varsha and S. Chitra Nair, "Automatic Attendance Management System Using Face Detection and Face Recognition," in *IoT and Analytics for Sensor Networks*, (Lecture Notes in Networks and Systems, 2022, ch. Chapter 9, pp. 97-106.
[27] M. V*, V. M. Vinod, and M. M, "Face Recognition Based Attendance System," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 12, pp. 520-525, 2019, doi: https://doi.org/ 10.35940/ijitee.L3406.1081219.
[28] R. Tamilkodi, "Automation System Software Assisting Educational Institutes for Attendance, Fee Dues, Report Generation Through Email and Mobile Phone Using Face Recognition," *Wireless Personal Communications*, vol. 119, no. 2, pp. 1093-1110, 2021, doi: https://doi.org/ 10.1007/s11277-021-08252-2.
[29] F. Majeed *et al.*, "Investigating the efficiency of deep learning based security system in a real-time environment using YOLOv5," *Sustainable Energy Technologies and Assessments*, vol. 53, 2022, doi: https://doi.org/ 10.1016/j.seta.2022.102603.
[30] N. A. Ismail *et al.*, "Web-based University Classroom Attendance System Based on Deep Learning Face Recognition," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 16, no. 2, pp. 503-523, 2022.
[31] P. Tavallali, M. Yazdi, and M. R. Khosravi, "A Systematic Training Procedure for Viola-Jones Face Detector in Heterogeneous Computing Architecture," *Journal of Grid Computing*, vol. 18, no. 4, pp. 847-862, 2020, doi: https://doi.org/ 10.1007/s10723-020-09517-z.
[32] I. Jasim Hussein *et al.*, "Fully Automatic Segmentation of Gynaecological Abnormality Using a New Viola-Jones Model," *Computers, Materials & Continua*, vol. 66, no. 3, pp. 3161-3182, 2021, doi: https://doi.org/ 10.32604/cmc.2021.012691.
[33] Y. Freund and R. E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119-139, 1997/08/01/ 1997, doi: https://doi.org/ https://doi.org/10.1006/jcss.1997.1504.
[34] M. Awais *et al.*, "Real-Time Surveillance Through Face Recognition Using HOG and Feedforward Neural Networks," *IEEE Access*, vol. 7, pp. 121236-121244, 2019, https://doi.org/ 10.1109/access.2019.2937810.
[35] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 20-25 June 2005 2005, vol. 1, pp. 886-893 vol. 1, doi: https://doi.org/ 10.1109/CVPR.2005.177.
[36] T. Watanabe, S. Ito, and K. Yokoi, "Co-occurrence Histograms of Oriented Gradients for Pedestrian Detection," Berlin, Heidelberg, 2009: Springer Berlin Heidelberg, in Advances in Image and Video Technology, pp. 37-47. [Online]. Available: https://doi.org/10.1007/978-3-540-92957-4_4.
[37] S. Sharma, L. Raja, V. Bhatnagar, D. Sharma, S. N. Bhagirath, and R. C. Poonia, "Hybrid HOG-SVM encrypted face detection and recognition model," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 25, no. 1, pp. 205-218, 2022, doi: https://doi.org/ 10.1080/09720529.2021.2014141.
[38] M. A. Hameed, M. Hassaballah, S. Aly, and A. I. Awad, "An Adaptive Image Steganography Method Based on Histogram of Oriented Gradient and PVD-LSB Techniques," *IEEE Access*, vol. 7, pp. 185189-185204, 2019, doi: https://doi.org/ 10.1109/access.2019.2960254.
[39] S. Afroze, M. R. Hossain, and M. M. Hoque, "DeepFocus: A visual focus of attention detection framework using deep learning in multi-object scenarios," *Journal of King Saud University - Computer and Information Sciences*, 2022, doi: https://doi.org/ 10.1016/j.jksuci.2022.10.009.
[40] E. C. Valverde, P. P. Oroceo, A. C. Caliwag, M. A. Kamali, W. Lim, and M. Maier, "Edge-Oriented Social Distance Monitoring System Based on MTCNN," *IEEE Transactions on Industrial Informatics*, pp. 1-13, 2022, doi: https://doi.org/ 10.1109/tii.2022.3217499.
[41] M. Gu, X. Liu, and J. Feng, "Classroom face detection algorithm based on improved MTCNN," *Signal, Image and Video Processing*, vol. 16, no. 5, pp. 1355-1362, 2022, doi: https://doi.org/ 10.1007/s11760-021-02087-x.
[42] Q. Guo, Z. Wang, D. Fan, and H. Wu, "Multi-face detection and alignment using multiple kernels," *Applied Soft Computing*, vol. 122, 2022, doi: https://doi.org/ 10.1016/j.asoc.2022.108808.
[43] A. B. Shetty, Bhoomika, Deeksha, J. Rebeiro, and Ramyashree, "Facial recognition using Haar cascade and LBP classifiers," *Global Transitions Proceedings*, vol. 2, no. 2, pp. 330-335, 2021, doi: https://doi.org/ 10.1016/j.gltp.2021.08.044.

[44] M. A. Abuzneid and A. Mahmood, "Enhanced Human Face Recognition Using LBPH Descriptor, Multi-KNN, and Back-Propagation Neural Network," *IEEE Access*, vol. 6, pp. 20641-20651, 2018, doi: https://doi.org/ 10.1109/access.2018.2825310.
[45] W. Chunming and Z. Ying, "MTCNN and FACENET Based Access Control System for Face Detection and Recognition," *Automatic Control and Computer Sciences*, vol. 55, no. 1, pp. 102-112, 2021, doi: https://doi.org/ 10.3103/s0146411621010090.
[46] F. Boutros, N. Damer, F. Kirchbuchner, and A. Kuijper, "Self-restrained triplet loss for accurate masked face recognition," *Pattern Recognition*, vol. 124, 2022, doi: https://doi.org/ 10.1016/j.patcog.2021.108473.
[47] H. Pranoto and O. Kusumawardani, "Real-time Triplet Loss Embedding Face Recognition for Authentication Student Attendance Records System Framework," 2021.
[48] X. Chen and H. Y. K. Lau, "The identity-level angular triplet loss for cross-age face recognition," *Applied Intelligence*, vol. 52, no. 6, pp. 6330-6339, 2021, doi: https://doi.org/ 10.1007/s10489-021-02742-3.
[49] M. Y. Kamil, "A deep learning framework to detect Covid-19 disease via chest X-ray and CT scan images," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 1, 2021, doi: https://doi.org/ 10.11591/ijece.v11i1.pp844-850.
[50] R. Muttaqin, S. Fuada, and E. Mulyana, "Attendance System using Machine Learning-based Face Detection for Meeting Room Application," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 8, 2020.
[51] S. M. Bah and F. Ming, "An improved face recognition algorithm and its application in attendance management system," *Array*, vol. 5, 2020, doi: https://doi.org/ 10.1016/j.array.2019.100014.
[52] J. Almotiri, "Face Recognition using Principal Component Analysis and Clustered Self-Organizing Map," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 3, 2022.
[53] V. Rohini, M. Sobhana, and C. S. Chowdary, "Attendance Monitoring System Design Based on Face Segmentation and Recognition," *Recent Patents on Engineering*, vol. 17, no. 2, 2023, doi: https://doi.org/ 10.2174/1872212116666220401154639.