

## Research Article

# Performance of Face Recognition System Using Gradient Laplacian Operators and New Features Extraction Method Based on Linear Regression Slope

Abdulbasit Alazzawi , Osman N. Ucan, and Oguz Bayat

Altinbas University, Engineering and Natural Science Faculty, Electrical and Computer Department, Istanbul, Turkey

Correspondence should be addressed to Abdulbasit Alazzawi; [abdulbasit.alazzawi@ogr.altinbas.edu.tr](mailto:abdulbasit.alazzawi@ogr.altinbas.edu.tr)

Received 2 May 2018; Revised 17 August 2018; Accepted 28 August 2018; Published 24 September 2018

Academic Editor: Piotr Jędrzejowicz

Copyright © 2018 Abdulbasit Alazzawi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recent research proves that face recognition systems can achieve high-quality results even in non-ideal environments. Edge detection techniques and feature extraction methods are popular mechanisms used in face recognition systems. Edge detection can be used to construct the face map in the image efficiently, in which feature extraction techniques generate the most suitable features that can identify human faces. In this study, we present a new and efficient face recognition system that uses various gradient-and Laplacian-based operators with a new feature extraction method. Different edge detection operators are exploited to obtain the best image edges. The new and robust method based on the slope of the linear regression, called SLP, uses the estimated face lines in its feature extraction step. Artificial neural network (ANN) is used as a classifier. To determine the best scheme that gives the best performance, we test combinations of various techniques such as (Sobel filter (SF), SLP/principal component analysis (PCA), ANN), (Prewitt filter (PF), SLP/PCA, ANN), (Roberts filter (RF), SLP/PCA, ANN), (zero cross filter (ZF), SLP/PCA, ANN), (Laplacian of Gaussian filter (LG), SLP/PCA, ANN), and (Canny filter (CF), SLP/PCA, ANN). The BIO ID dataset is used in the training and testing phases for the proposed face recognition system combinations. Experimental results indicate that the proposed schemes achieve satisfactory results with high-accuracy classification. Notably, the combinations of (SF, SLP, ANN) and (ZF, SLP, ANN) gain the best results and outperform all the other algorithm combinations.

## 1. Introduction

Face recognition system has been widely used in the biometric technology area. The increasing commercial use of this system has increased its robustness and efficiency in the recent years. Furthermore, the development of face recognition algorithms in hardware systems has led to the relatively easy application and wide distribution of these systems. Subsequently, such technologies have been implemented in many system applications, such as face verification, face identification, surveillance, and monitoring system.

Face recognition has been studied in many works and research projects due to its importance. Reference [1] proposed face recognition based on polar fast Fourier transform. In this work, the face was detected using the Viola-Jones algorithm [2], whereby the image was converted into a

greyscale image with a small size of 120120 pixel. Discrete Fourier transform was applied to the preprocessed image, and the dominant value of 2D-DFT quantities was calculated using polar Fourier transform method and characterized as 1D P-FF. A magnitude that denotes maximum value in 1D PFFT was considered a feature value. This extracted value was utilized to construct a symbolic object for epitomizing face images. The performance of the proposed scheme was tested using the AR database [3], and an average accuracy of 96.25% was achieved.

Reference [4] introduced another face recognition system using 1D transform domains. The authors benefitted from the combination of discrete wavelet transform and fast Fourier transform for feature extraction. 1D-DWT and 1D-FFT were compared using probe images and a database. The Euclidian distance was used to calculate the performance parameters,

and matching processes were conducted to obtain the best results.

Reference [5] put forward the local appearance-based face recognition by using adaptive directional wavelet transform on the edges of extracted faces. The authors reported the importance of this algorithm by comparing this technique with efficient feature extraction algorithms such as linear discriminate analysis and existing non-adaptive multi-resolution. The popular Yale database was used to reveal the effectiveness of the adaptive directional wavelet transform-based subspace feature. The algorithms showed acceptable results compared with other face recognition algorithms.

Reference [6] established a new algorithm by modifying the eigenphase method for face recognition based on support vector machine (SVM). The authors performed two phases to increase accuracy. The first one was the local spatial domain phase, in which face images were segmented into equal size blocks. These blocks were concatenated as preparation for phase spectrum estimation. In the second phase, which was local frequency domain eigenphase, a similar special procedure was performed but in the frequency domain. The output of the two phases was fed to the SVM for classification. Accordingly, this algorithm could work better than the other compared algorithms in time and performance.

In [7], the authors proposed a new algorithm for face recognition using two feature extractors PCA technique and Wavelet decomposition vectors. This algorithm uses also two classifiers which are nearest distance and support vector machine. The proposed system was tested using the ORL database [8] and the experimental results showed that the new system enhanced the performance significantly. In another work, the authors suggested using a neural network instead of SVM to more enhance the performance.

Reference [7] proposed a new algorithm for face recognition using two feature extractors, namely, principal component analysis (PCA) technique and wavelet decomposition vectors. This algorithm used two classifiers: nearest distance and SVM. The proposed system was tested using the ORL database [8]. The experimental results showed that the new system enhanced the performance considerably.

Other papers have discussed face recognition issues using several techniques. Some works have depended on the appearance of the face only and used different combinations with the PCA algorithm (e.g. [9–11]). [12] proposed the Fisher face methods, which depended on the appearance of the face and was based on Fishers linear discriminant analysis. Other studies have focused on the geometry of the face and the relation between parts of the face and its properties such as nose, mouth, and eyes. Reference [13] used 16 and 30 face geometry features, respectively, and fed them into a classifier algorithm for classification. Reference [14] extracted the geometry of human face in detail using a 3D representation model. In many fields, researchers have tried to make hybridisation models between the appearance and geometry of the face, e.g., [15, 16]. Most of these approaches and techniques have limitations in terms of time for extracting features, the response time for the training dataset, and an update of the training dataset and hardware requirements.

This paper proposes a robust face recognition approach regardless of face image quality and illumination conditions. In this study, two groups of edge detection methods are used with the slope of the linear regression (SLP) operator to enhance the performance of the face recognition system. Edge detection methods include gradient-based operators (GOs) and Laplacian-based operators (LOs). The GOs include Sobel filter (SF), Prewitt filter (PF), and Roberts filter (RF). The LOs include Laplacian of Gaussian (LoG) filter (LF), Zero cross filter (ZCF), and Canny filter (CF). For feature extraction, a new method called SLP is suggested. The proposed SLP uses a simple regression technique with the eigenfaces based on PCA. Artificial neural networks are utilized for classification. The BIO ID database [17] is used in our empirical experiments. This database includes 23 subjects with 1,502 face images. The results show that the proposed (SF, SLP, ANN) method can enhance the performance of a face recognition system and gains high-accuracy classification.

The rest of the paper is organized as follows. In Section 2, we briefly review the edge detection methods that will be used in this study, namely, PF, SF, RF, ZF, LF, and CF. Section 3 explains our proposed algorithm that includes the segmentation process, feature extraction methods, and ANN. Section 4 presents the experimental results and detailed discussion. Finally, Section 5 provides the conclusion and future work.

## 2. Edge Detection Algorithms

Edge detection is a fundamental step towards understanding image features because edges consist of feature data and contain important information related to face edges. Most images contain a certain amount of redundancy that can sometimes be deleted when edges are detected and replaced during reconstruction. This gap is where edge detection comes into play. The edge detection process aims to accurately detect the image edges, reduce the image size, and remove information that may be regarded as less relevant. In the proposed system, the goal of the edge detection operation is to construct a map of the face from the edges that can be used to extract features of face objects and prepare the features for the subsequent stages of the proposed system. We apply most of the edge detection filters (SF, PF, RF, CF, LF, and ZF) on the dataset images to perform this process and study the differences amongst the image edges [18, 19].

*2.1. Sobel Filter.* The SF is used to respond maximally to image edges that change in vertical and horizontal directions. Two masks are used in the Sobel operator. One mask works in the vertical direction and the other mask is in the horizontal direction of image pixels. These kernels of the SF can be used separately to obtain isolated estimations of the gradient in horizontal and vertical directions,  $G_x$  and  $G_y$ , respectively [20]. The two kernels can be joined together to find the value of the gradient at each pixel and the direction of the gradient as it can be seen in Figure 1. The following equations explain the magnitude and direction of the gradient.

$$|G| = \sqrt{(G_x^2 + G_y^2)} \quad (1)$$

Gx		
-1	0	+1
-2	0	+2
-1	0	+1

Gy		
+1	+2	+1
-2	0	+2
-1	0	+1

FIGURE 1: Sobel convolution kernels.

Gx		
-1	+1	+1
-1	-2	+1
-1	+1	+1

Gy		
+1	+1	+1
-1	-2	-1
-1	-1	+1

FIGURE 2: Prewitt compass edge detecting templates.

Typically, an approximate magnitude is computed using

$$|G| = |Gx| + |Gy| \quad (2)$$

which can be computed rapidly. The angle of orientation of the edge (relative to the pixel grid) giving rise to the spatial gradient is given by

$$X = \arctan \frac{Gy}{Gx} \quad (3)$$

**2.2. Prewitt Filter.** The PF works by convolving the image with a small, discrete, and integer-valued operator in horizontal and vertical directions. The algorithm ignores all edges that are not higher than the threshold. The threshold value is set automatically by the Prewitt algorithm. In the proposed algorithm, we use a range of threshold values (instead of standard value) to find the best edge pixels. In the Prewitt model, the image is treated using a kernel with eight values. All these values are sensitive to image edges in both directions. For each pixel, the local edge gradient magnitude is evaluated with the maximum reaction of all eight kernels at the considered pixel location, as shown in Figure 2. The gradient magnitude is given by

$$|G| = \max(|Gi| \dots i = 1 \dots n) \quad (4)$$

where  $Gi$  is the response of kernel ( $i$ ) at the specific pixel location and ( $n$ ) is the number of convolution kernels. The local edge orientation is estimated with the orientation of the kernel that yields the maximum reaction [20, 21].

**2.3. Roberts Filter.** The RF applies an easy and rapid calculation called 2D spatial filter on an image. Regularly, the RF estimates the values (magnitude) of the gradient for every pixel point. The operator consists of a pair of 22 convolution kernels, as shown in Figure 3. One kernel is simply the other rotated by  $90^\circ$  [21]. The kernels can be applied separately to the input image to produce separate measurements of the gradient component in each orientation (Gx and Gy).

Gx	
+1	0
0	-1

Gy	
0	+1
-1	0

FIGURE 3: Roberts cross convolution kernels.

Thereafter, both kernels can be combined to find the absolute magnitude of the gradient at each point and the orientation of that gradient. The gradient magnitude can be calculated by

$$|G| = |Gx| + |Gy| \quad (5)$$

The two components of the gradient are conveniently computed and added in a single pass over the input image by using the pseudoconvolution operator.

**2.4. Zero Cross Filter.** The zero cross edge detector determines zero crossing regions in images where the values of the Laplacian cross through zero regions. According to this operator, the edge point pixels are mostly represented by points that have a value of the Laplacian passing through zero. Hence, in these points, the Laplacian is changing its sign. For example, in face recognition algorithms, the points where the intensity of the face image edges changes suddenly are considered edge point candidates. Scientists have recommended dealing with this filter as a feature extractor operator because the output of the zero cross algorithm is usually a binary image with one-pixel thickness [22].

**2.5. LoG Filter.** The LoG algorithm is proposed to solve the noise sensitivity of the regular Laplacian filter. This algorithm removes noise by applying image smoothing using a Gaussian blurring technique to obtain the best performance. After

applying the Laplacian method, a change in sign from positive to negative or vice-versa for any point (changes in sign) represents an edge. This algorithm supposes that important cross points will nearly be at edges in the images, that is, where the intensity values of the image regions change sharply. However, they can occur at places that are not as easy to allocate with image edges because some of these regions are very similar to one another, especially in greyscale images. The lines are detected by the LoG procedure with one-pixel thickness. The Laplacian  $L(x, y)$  of an image with pixel intensity values  $I(x, y)$  is given by

$$L(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \quad (6)$$

This step is an important pre-processing operation that decreases unwanted frequency noise prior to the differentiation operation. The 2D LOG function is centered on zero and with Gaussian standard deviation in the following form:

$$\text{LoG}(x, y) = \frac{-1}{\pi\sigma^4} * \left( 1 - \frac{x^2}{y^2} * \exp\left(-\frac{X^2}{2\sigma^2}\right) \right) \quad (7)$$

where  $x$  is the distance from the origin in the horizontal axis,  $y$  is the distance from the origin in the vertical axis, and  $\sigma$  is the standard deviation of the Gaussian distribution. The Gaussian smoothing filter and the Laplacian filter then convolve into this output result (hybrid filter) with the digital image to reach the desired output [22].

**2.6. Canny Filter.** Canny edge detection uses five different and sequential procedures to detect the edges. The first step is to smoothen the image to remove possible noise using a Gaussian filter. The second step is to compute gradient values in vertical ( $G_x$ ) and horizontal ( $G_y$ ) directions, which is the same procedure used in the Sobel algorithm. CF eliminates pixels that are not part of image edges by using a non-maximum suppression process. In the third step, threshold values are used to determine the potential edges. In the last step of the Canny algorithm, all edge pixels that have weak values or isolated from other pixels are eliminated and rejected [23, 24]. For further explanation, the mathematical model of the CF includes the following equations:

$$g_m = f(x+1, y) - f(x, y) \quad (8)$$

$$g_n = f(x, y+1) - f(x, y) \quad (9)$$

where  $g_m$  and  $g_n$  represents the numbers of pixels between two points in horizontal and vertical directions.

$$g(x, y) = G_\sigma(x, y) * f(x, y) \quad (10)$$

where

$$G_\sigma = \frac{1}{\sqrt{2\pi\sigma^2}} * \exp\left(-\frac{(x^2 + y^2)}{2\sigma^2}\right) \quad (11)$$

Compute gradient of  $g(i, j)$  using any of the gradient operators to get

$$M(x, y) = \sqrt{(g_m^2(x, y) + g_n^2(x, y))} \quad (12)$$

and

$$\theta(m, n) = \arctan \frac{g_m(x, y)}{g_n(x, y)} \quad (13)$$

In the equations above, the threshold is chosen such that all edge elements are kept whilst most of the noise is suppressed.

All the previous edge detection filters are implemented and illustrated in Figures S1-S6.

### 3. Proposed Face Recognition Algorithm

The proposed algorithm consists of three main steps. In the first step, the face images are preprocessed with special filters, and one edge detection filter is selected and applied to the face image. This study uses six edge detection operators that will be tested with other combinations of our proposed algorithm. After that step, the resulting binary image from the edge detection method is segmented using the specified method, which will be explained later. In the second step, a feature extraction algorithm is applied to select the most important features of the segmented binary image. In this study, two feature extraction methods will be tested and used, namely, PCA and our new proposed SLP filter. In the last step, after extracting the features, all the gained features of faces will be used to train the artificial neural network for classification purposes. Figure 4 presents the steps of our proposed algorithm. This figure shows the steps of training the proposed method; the test steps are the same except that the ANN training step will not be conducted and the test dataset of face images will be used.

**3.1. Segmentation.** The idea behind dividing the face images into sub-images is inspired by the popular problem-solving technique Divide and Conquer presented in [25]. Dividing a problem into many parts to solve each of them separately can help in solving complex problems. Hence, the complexity of the problem is simplified. Complex problems, such as face recognition, can be solved with efficiency if we correctly adapt the problem-dividing technique. In our proposed algorithm, each training face image is segmented into 2D blocks with the same size. Each block is converted to the 1D feature vector, which will be the input of the subsequent algorithm step or feature extraction. Selecting the number of segments is an important and sensitive process because the choice can affect the overall face recognition system. In the next section, we will explain the steps of determining the optimal number of the segment for each face image experimentally.

#### 3.2. Features Extractors

**3.2.1. PCA.** The eigenface finding based on PCA technique (also known as Kohonen-Loeve filter) is specifically used for face image data. In the PCA method, all dataset images are recognized with feature vectors that are created from projections of the training image to the basis in image space. PCA classifies dataset images according to calculated distances amongst feature vectors. Typical classifiers comprise nearest distance measure, Euclidean distance and nearest mean classification [26–29]. When PCA is used for the

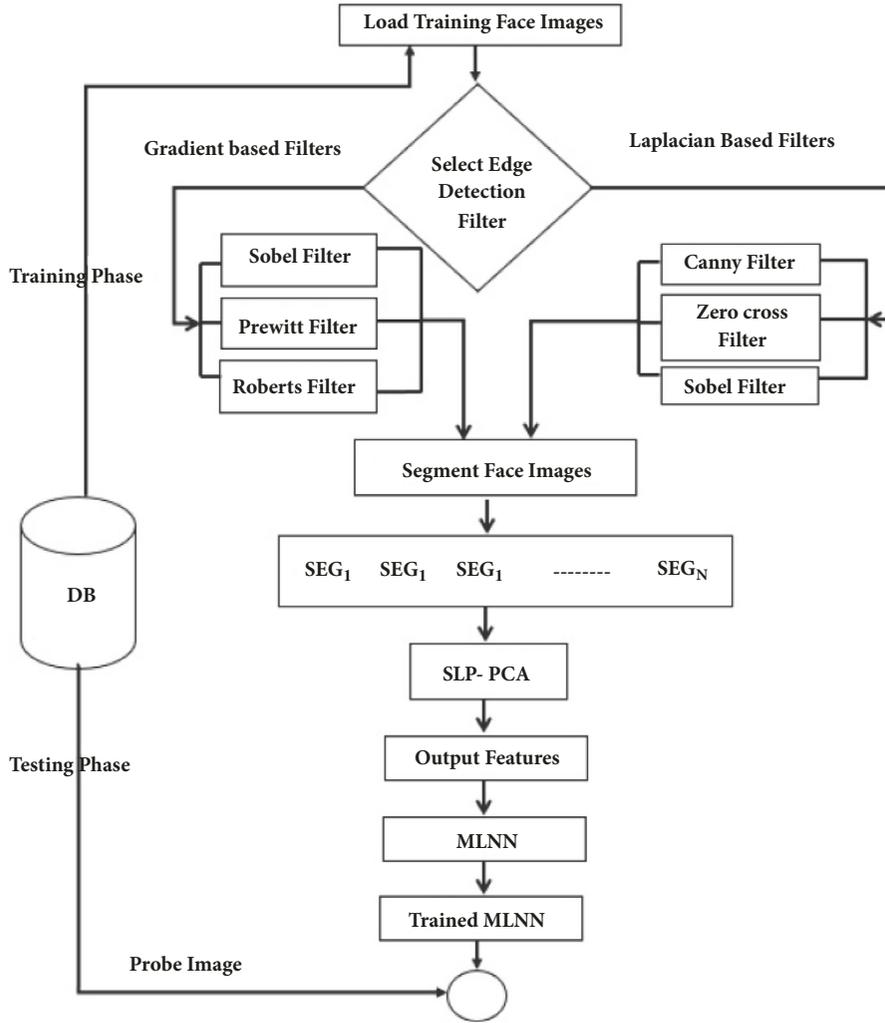


FIGURE 4: The main diagram of proposed system.

eigenface method, feature vectors identifying each image are shown in Figure 6. These vectors are obtained as follows:

- (1) Assume that we have  $N$  face images with  $(x)$  rows and  $(y)$  column.

$$S = S_1, S_2, S_3, \dots, S_N \quad (14)$$

representing those images in column vectors with  $(x^2 * 1)$  dimensions. We need to calculate a mean face image ( $Q$ ), from these vectors like,

$$Q = \frac{1}{N} \sum_{i=1}^N S_i \quad (15)$$

where “ $Q$ ” is the mean matrix (common properties of faces).

- (2) The main data matrix is subtracted from  $(S_i)$ , which can be represented as

$$L_i = S_i - Q \quad (16)$$

- (3)  $L_i$  column vectors are gathered in matrix:

$R = [L_1, L_2, \dots, L_N]$  with dimension  $(x^2 * N)$  and a covariance matrix  $C$  is formed as

$$C = A.A^T \quad (17)$$

Given high dimensionality and considering the linear algebra principle as well as great computational complexity we use  $A^T A$  instead.

$$C^* = A^T . A \quad (18)$$

- (4) We calculate  $N$  eigenvalues  $(\lambda_k)$  and  $N$  eigenvectors  $(V_k)$  of  $C^*$  to form the eigenface space.

$V = [v_1, v_2, \dots, v_N]$  represents a matrix including eigenvectors of  $(C)$  with a dimension of  $(x * x)$ . We can obtain eigenface space  $U = [u_1, u_2, \dots, u_N]^T$  by

$$U = V.A^T \quad (19)$$

All row vectors of ( $U$ ) are eigenfaces of face images in the training set. Face images with high eigenvalues have great contributions to the eigenface space. For this reason, systems with low computational capability sort eigenvectors of face images according to their corresponding eigenvalues in decreasing order and choose eigenvectors to form a small eigenface space.

(5)

$$W = w_1, w_2, w_3, \dots, w_N \quad (20)$$

The matrix with ( $x \times x$ ) dimension includes ( $N$ ) column vectors corresponding to each face image in the training set. These vectors are called feature vectors and they represent each images specific characteristics. ( $W$ ) can be obtained

$$W = U.A \quad (21)$$

After the eigenface space and feature vectors are obtained, we can compare a test image with the faces in the training set by projecting a test image into face space as follows:

- (6) " $S_T$ " is a column vector that represents our test image with ( $x^2 * 1$ ) dimension. At this stage, the distance of test image from mean face image should be calculated as  $L_T$  column - vector,

$$L_T = S_T - Q \quad (22)$$

- (7) After  $L_T$  calculated, we must project the result on our eigenface space to obtain  $w_T$  with ( $x * 1$ ) dimension.

$$w_T = U.L_T \quad (23)$$

- (8) (c) We need to find the similarity of  $w_T$  in each ( $w_i$ ) in the matrix ( $W$ ) to determine which image in the training set resembles our test image. Various classifiers can be used at this stage [26–29].

**3.2.2. Slope Based Method.** To extract the best features from edge images efficiently, a new feature extraction method based on the slope of the estimated curve of each small segment is used. The proposed SLP extraction method produces vectors of features that can identify each face successfully, as shown in Figure 5. The following steps explain in detail the process of extracting the features in the training phase:

- (1) We suppose we have  $N$  face images ( $im_1, im_2, im_3 \dots im_N$ ) with  $m$  rows and  $n$  columns, where  $N$  is the number of images in the training set.
- (2) We convert the face images ( $im_1, im_2, im_3 \dots im_N$ ) to binary images by using one of the six edge detection filters discussed in Section 2. The binary images are denoted as ( $B_1, B_2, B_3 \dots B_N$ ).
- (3) We divide each binary image  $B_i$ , into segments, where  $H$  is the number of rows and  $L$  is the number of columns in each segment. The segments of image face  $B_i$  are denoted by ( $S_{i1}, S_{i2}, \dots, S_{ik}$ ), where  $k$  is the number of segments with ( $H * L$ ) dimension.

- (4) For each segment  $S_{ij}$  of each face image  $B_i$ , the value of pixels with ones is considered as points and a linear regression estimation method is used to find the best line that fit these points. Assume that the equation of the line is as

$$Y = m * x + b \quad (24)$$

where  $b$  is the  $Y$ -intercept and  $m$  is the slope of the estimated line.

- (5) Compute the slop of the estimated line of each segment  $S_{ij}$  using the following equation which is derived from Pearson's correlation coefficient formula and linear regression:

$$m = \frac{(n \sum (xy) - \sum (x) \sum (y))}{(n \sum (x^2)) - (\sum x)^2} \quad (25)$$

where  $m$  is the slop of  $S_{ij}$  segment,  $x$  is the  $x$ -coordinates,  $y$  is the  $y$ -coordinates and  $n$  is the length of  $x, y$  vector in the binary image. In our work, we select the linear regression techniques because it is the most widely used statistical technique to model a relationship between two sets of variables and it is widely used to estimate the line equation from a set of points. Slopes of segments are collected in a matrix with dimension ( $H * L$ ) corresponding to each face image in the training set to create features vectors which will be the input to the Multi-layer Neural Network. The usage of the linear regression techniques in our steps makes the proposed algorithm works very well since it generates very robust features that are not affecting by outlier points or noise points that may exist in the binary edge images.

**3.3. Feed Forward Neural Network.** The last step in the proposed system is to train the neural network on the dataset. The input matrix of the training dataset consists of the feature vectors after we apply all the preprocessing operations, dimensionality reduction, and feature extraction on dataset images. ANN is a classifier that is used in the proposed face recognition system. The steps of this algorithm are summarized as follow:

- (1) The weighted sum of the input vectors (dot product between the input vector and weights that connect the input and the hidden layers) is calculated [30, 31].

$$NT_g = \sum_{k=1}^n (X_k * W_{gk}) \quad (26)$$

- (2) The results from the weighted sum calculation are passed to the activation function to normalize the output. In the proposed neural network, we use the adaptive sigmoid function.

$$H_g = \frac{1}{1 + \exp(-NT_g)} \quad (27)$$

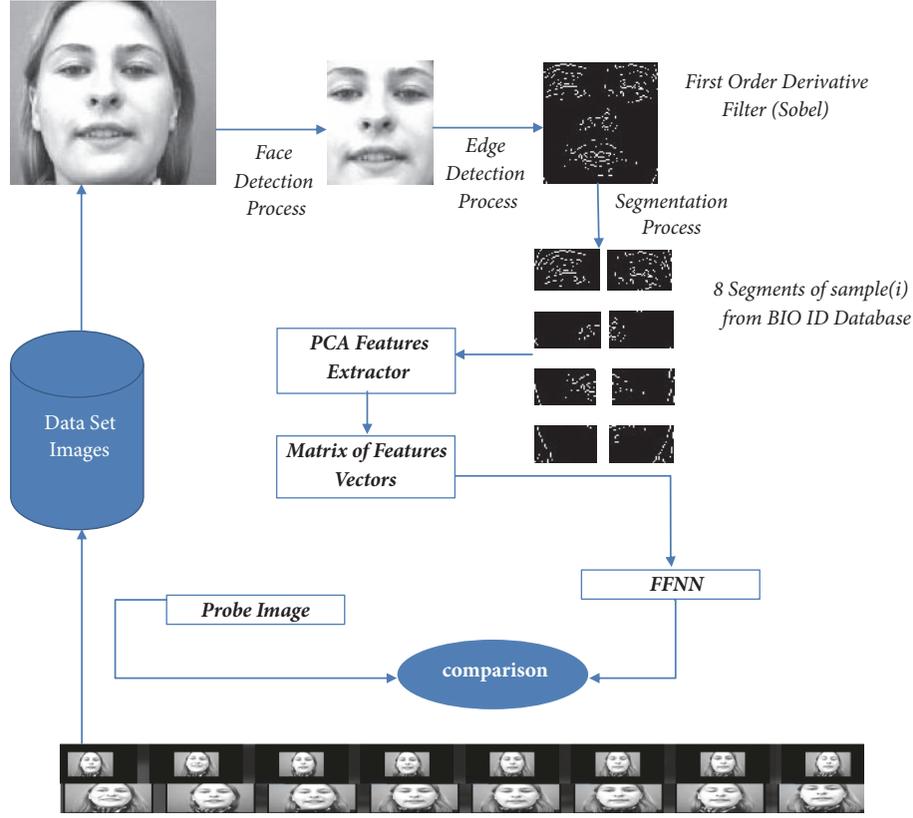


FIGURE 5: PCA steps diagram, FFNN refers to feedforward neural network.

- (3) In the same way as above, we calculate the output of the final layer.

$$Out_v = \sum_{l=1}^m (H_l * W_{lv}) \quad (28)$$

$$TrueNOut_i = \frac{1}{1 + \exp(-Out_v)} \quad (29)$$

- (4) After the output is calculated, error evolution between actual and the desired output is started by using the least square error function and updating of the weights of layers according to the rate tolerance in errors.

$$Error = \sum_{i=1}^v (TrueNOut_i - D_i) \quad (30)$$

$$\Delta W_{li} = \alpha \delta_i H_i \quad (31)$$

$$\delta_i = (D_i - TrueNOut_i) * TrueNOut_i * (H_i - TrueNOut_i) \quad (32)$$

#### 4. Results and Discussion

As described earlier, the combinations of different face recognition systems are developed and will be tested in this

section. The two steps in our proposed algorithm may use different schemes. The first step is change detection, which may be one of the six edge detection filters from the two groups. The first group is based on the gradient method and includes SF, PF, and RF. The second group includes LF, ZF, and CF that perform a Laplacian process before they begin edge detection. In the second step, which is feature extraction, we use an eigenvector-based method or the SLP method. ANN is the classifier that is used for the classification of the proposed algorithm combinations. The BIO ID database used for the training and the testing processes has 23 subjects (1,502 images). The face images are saved in a single file using the portable grey map (pgm) format. The pgm file contains a data header that has the image file information including image format, width and height and maximum allowed grey level value followed by the image data. Many experiments are carried out to determine the best parameters that we should use in each algorithm. One of the most important parameters is the number of segments that we should use in the first step of the proposed algorithm. Figure 7 shows the effects of using a different number of segments on the performance of our proposed algorithm. As noted from the results, the number of segments may considerably affect the performance of the algorithm. We also note that the best number of segments is 10. Hence, if we select values more or less than this value, then the results will not be acceptable. Therefore, in all our experiments, the face images are divided into 10 segments horizontally and vertically, thereby producing 100 segments.

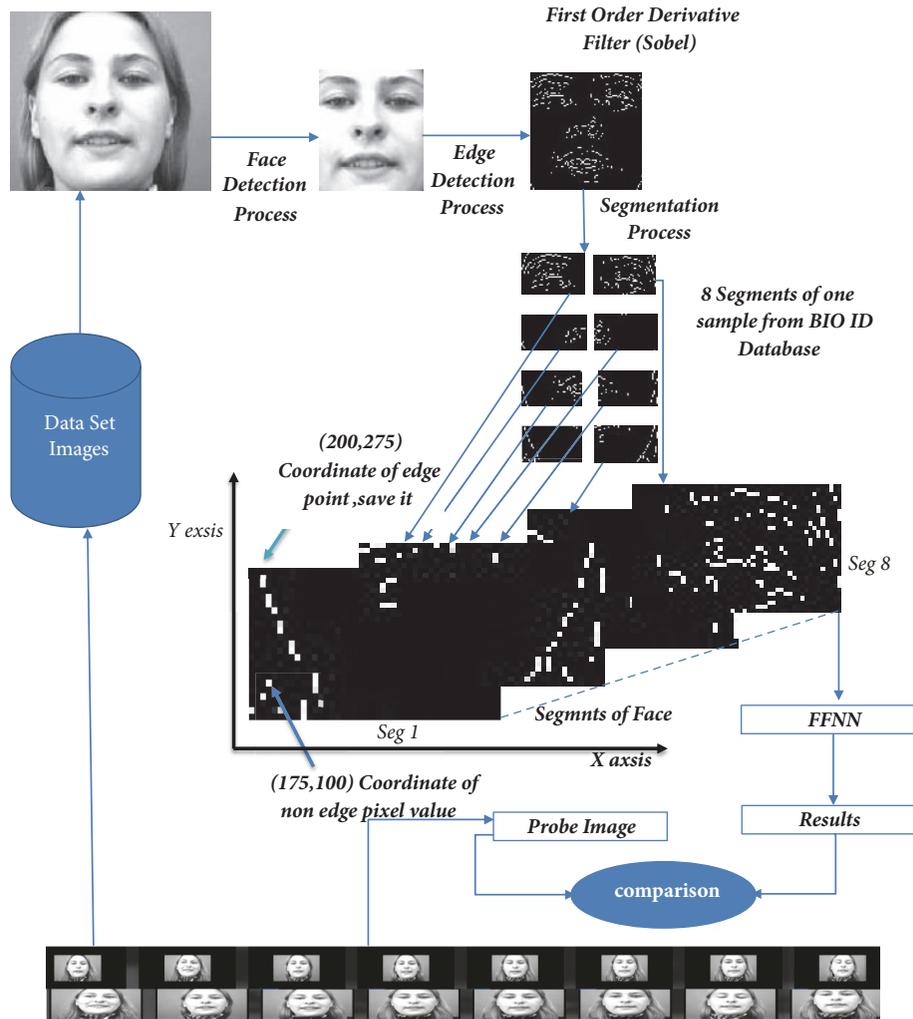


FIGURE 6: SLP steps diagram, FFNN refers to feedforward neural network classifier.

The second most important parameter is the best number of hidden layers that should be used to classify the complex scenarios efficiently. For determining this parameter, another experiment is executed on different algorithms to extract the best value. Figure 8 shows the results of this experiment. As shown in the figure, the best number of neurons that should be used is 30. However, increasing the number of neurons is not always as good as the results show.

In addition to the number of neurons and several segments, the threshold values of the edge detection filters are important and considerably affect the resulting binary edge image and the classification process. To select the thresholds accurately, a separate experiment is performed for each of the considered six edge detection filters. Figures 9–14 show the results of using different threshold values for each edge detection method. In our experiments, the best threshold value that maximizes the performance is selected for each edge filter. Tables 1 and 2 show the results of comparing different combinations of the proposed algorithm by using the BIO ID database. Table 1 presents the results of using SF, PF, and RF for SLP and PCA feature extraction methods.

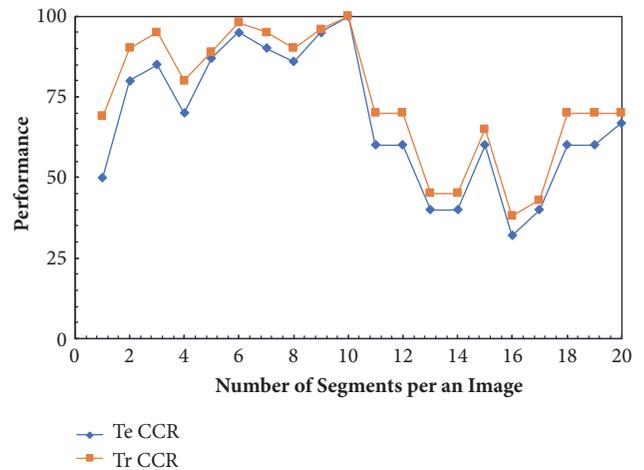


FIGURE 7: Number of segments per an Image.

For all algorithms, the multilayer perceptron neural network (MLPNN) is used. Table 2 presents the results of using LF, ZF,

TABLE 1: Sobel filter (SF), Prewitt filter (PF), Roberts filter (RF), SLP, PCA, and multilayer neural network (MLNN). Testing correct classification rate.

	Testing Rate	15%	20%	30%	40%	Weighted Mean
Technique	SF-SLP-ANN	100.0000	93.6508	90.7692	91.2371	93.1427
	SF-PCA-ANN	93.3133	92.0635	89.0635	85.5761	90.0041
	PF-SLP-ANN	96.6667	96.7742	89.2308	91.7526	93.6060
	PF-PCA-ANN	93.5484	93.6508	90.0000	89.6907	91.7224
	RF-SLP-ANN	96.6576	93.5485	92.5357	92.2681	93.7524
	RF-PCA-ANN	90.0000	92.0635	80.5970	86.5979	87.3146

TABLE 2: LoG (LF), zero cross filter (ZF), Canny filter (CF), SLP, PCA, and multilayer neural network (MLNN). Testing correct classification rate.

	Testing Rate	15%	20%	30%	40%	Weighted Mean
Technique	LF-SLP-ANN	96.0000	95.2381	93.0765	92.5632	95.21945
	LF-PCA-ANN	96.9759	95.8745	90.5467	90.7216	93.5296
	ZF-SLP-ANN	100.0000	95.2381	94.0299	93.8199	95.7719
	ZF-PCA-ANN	96.6567	95.3425	88.0548	92.2045	93.0646
	CF-SLP-ANN	93.5484	93.6509	92.6547	91.5614	92.8585
	CF-PCA-ANN	88.7086	92.7546	90.5187	91.5032	90.8711

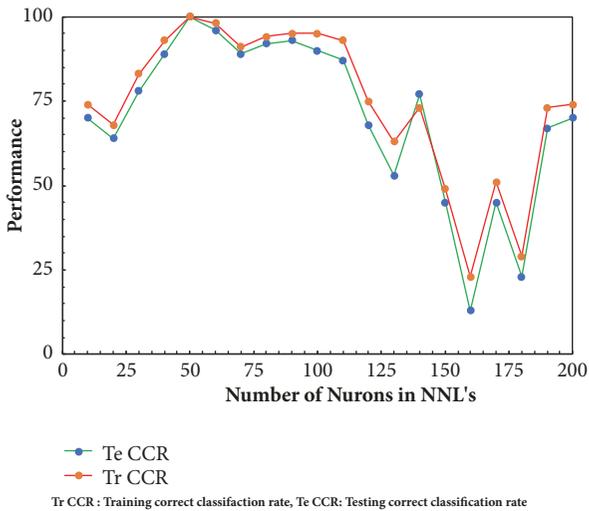


FIGURE 8: Number of neurons in NNs.

and CF for SLP and PCA feature extraction methods. For all algorithms, the MLPNN is also used.

The correct classification rate is used to measure the performance of the compared algorithms. This performance metric computes the average number of correctly classified faces as described in the following equation. To obtain the correct classification rate, the rates of BIO ID training and test data are set to four different values:

- (i) 85 % of data set for training and 15% of the data set for testing.
- (ii) 80 % of data set for training and 20% of the data set for testing.
- (iii) 70 % of data set for training and 30% of the data set for testing.

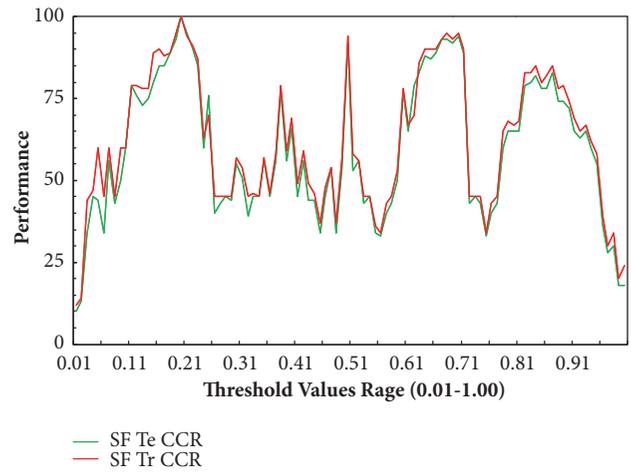


FIGURE 9: Performance of system based on Sobel filter.

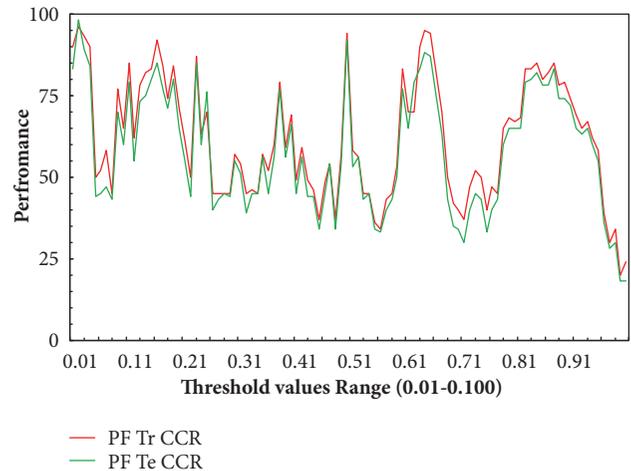


FIGURE 10: Performance of system based on Prewitt filter.

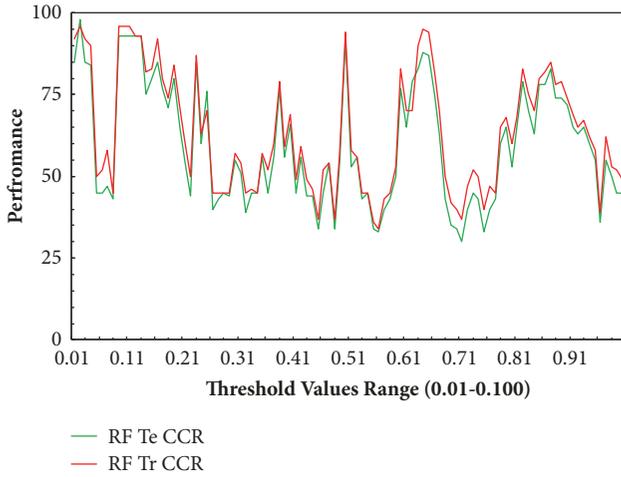


FIGURE 11: Performance of system based on Roberts filter.

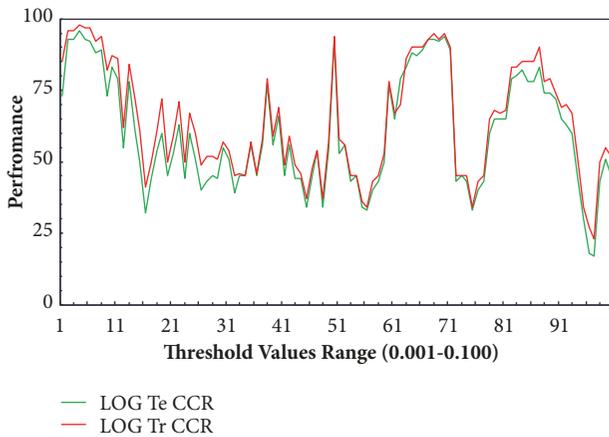


FIGURE 12: Performance of system based on LoG filter.

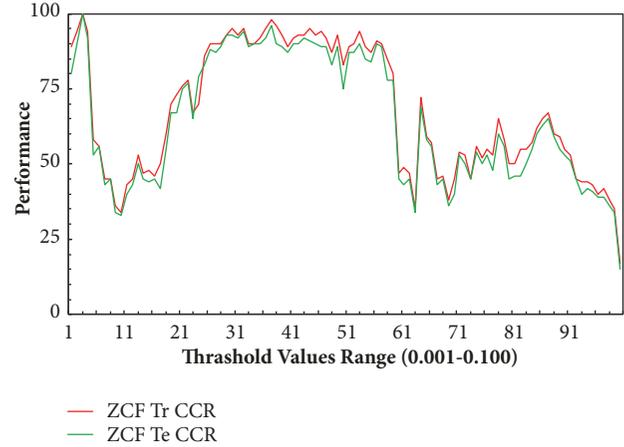


FIGURE 13: Performance of system based on zero cross filter.

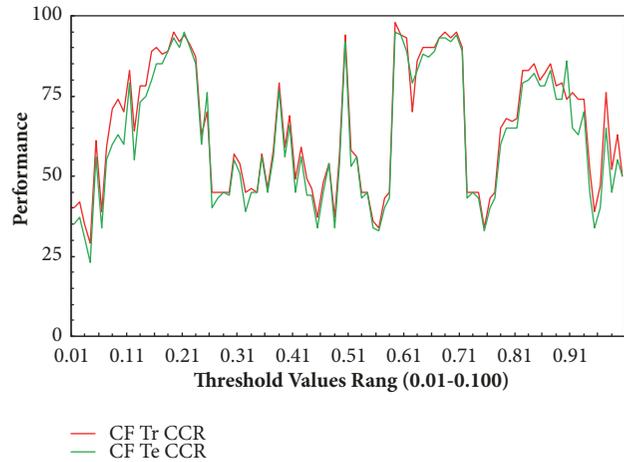


FIGURE 14: Performance of system based on Canny filter.

- (iv) 60 % of data set for training and 40% of the data set for testing.

As shown in Table 1, the SF, PF, and RF algorithms that use our new proposed SLP method outperform the other PCA versions. The best algorithm is the (RF, SLP, ANN) algorithm, which obtains the optimal mean value and most valuable results in two different percentages of training data. RF works effectively with binary images and is widely known as the best supporter kernel. The algorithms that use the PCA method obtain good results. For example, the (PF, PCA, ANN) algorithm obtains an average performance of 93%.

The results of the second-order filters in Table 2 confirm the performance of our new proposed combination with the new proposed SLP feature extraction method. As shown in Table 2, the LF, ZF and CF algorithms that work with our new proposed SLP method outperform the other PCA versions. In Table 2, the best algorithm is (ZF, SLP, ANN) with a correct classification of 95.7%.

The proposed SLP method can considerably enhance the performance of the face recognition system and performs better than the PCA mechanism. The proposed SLP method

creates position matrixes of pixel values to calculate the slope of each point. Such a feature makes SLP robust against image environmental factors, such as light conditions and complex backgrounds. Moreover, SLP does not need complex normalization processes, such as dimensionality reduction, and additional execution time. Second-order filters such as LF, ZF, and CF generate a more suitable binary image for face recognition than the first-order filters such as SF, PF, and RF.

### 5. Conclusion

This paper presents a new and efficient face recognition system that uses a new and efficient feature extractor called the SLP method. The proposed algorithm uses an edge detection algorithm followed by the SLP mechanism to extract the important features of the face. We use combinations of various techniques such as (SF, SLP/PCA, ANN), (PF, SLP/PCA, ANN), (RF, SLP/PCA, ANN), (ZF, SLP/PCA, ANN), (LF, SLP/PCA, ANN), and (CF, SLP/PCA, ANN) to achieve the best combination. The BIO ID dataset is used in the training and testing phases for the proposed face recognition system combinations. The experimental results

show that the proposed schemes achieve satisfactory results with high-accuracy classification. Notably, the combination of (ZF, SLP, ANN) gains the best results and outperforms all the other algorithm combinations.

### Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

### Conflicts of Interest

The authors declare that they have no conflicts of interest.

### Supplementary Materials

Figures 1s-6s: effect of different edge operators (SF, PF, RF, ZF, LF, and CF) which were utilized during the edge detection process. Edge derived from BIO ID dataset images according to the chosen threshold values are present clearly in the main images of the main manuscript. Supplementary Figure 7s: example of the segmentation and SLP feature extraction method. Supplementary Figure 8s: example segmentation and PCA feature extraction method. Figures 9s-27s represent examples of some test results. (*Supplementary Materials*)

### References

- [1] S. A. Angadi and V. C. Kagawade, "A robust face recognition approach through symbolic modeling of Polar FFT features," *Pattern Recognition*, vol. 71, pp. 235–248, 2017.
- [2] P. Viola, M. J. Jones, and R. R. Face, "The Viola Jones algorithm for face detection".
- [3] M. Martinez a and R. Benavente, "The AR face database," Tech. Rep., CVC, 1998, <https://doi.org/10.1023/B:VISI.0000029666.37597>.
- [4] S. Halvi, N. Ramapur, K. B. Raja, and S. Prasad, "Fusion Based Face Recognition System using ID Transform Domains," *Procedia Computer Science*, vol. 115, pp. 383–390, 2017.
- [5] M. A. Muqet and R. S. Holambe, "Local appearance-based face recognition using adaptive directional wavelet transform," *Journal of King Saud University - Computer and Information Sciences*, 2016.
- [6] J. Olivares-Mercado, K. Toscano-Medina, G. Sánchez Pérez, M. Nakano-Miyatake, and H. Pórez-Meana, "Modifications to the Eigenphases Method for Face Recognition Based on SVM. Ingeniería," *Investigación Y Tecnología*, vol. 17, no. 1, pp. 119–129, 2016.
- [7] E. Gumus, N. Kilic, A. Sertbas, and O. N. Ucan, "Evaluation of face recognition techniques using PCA, wavelets and SVM," *Expert Systems with Applications*, vol. 37, no. 9, pp. 6404–6408, 2010.
- [8] "Facedatabase www.Cl.Cam.Ac.Uk," <http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>.
- [9] B. Moghaddam, "Principal manifolds and Bayesian subspaces for visual recognition," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, pp. 1131–1136 vol.2, Kerkyra, Greece, September 1999.
- [10] K. Chung, S. C. Kee, and S. R. Kim, "Face recognition using principal component analysis of Gabor filter responses," *Ratfg-Rts*, vol. 1, Article ID 799223, 1999.
- [11] N. Sun, H. X. Wang, Z. H. Ji, C. R. Zou, and L. Zhao, "An efficient algorithm for Kernel two-dimensional principal component analysis," *Neural Computing and Applications*, vol. 17, no. 1, pp. 59–64, 2008.
- [12] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs., Fisherfaces, Recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.
- [13] T. Kanade, *Computer recognition of human faces*, Birkhäuser Basel, Basel, 1977.
- [14] A. Lanitis, C. J. Taylor, and T. F. Cootes, "Automatic face identification system using flexible appearance models," *Image and Vision Computing*, vol. 13, no. 5, pp. 393–401, 1995.
- [15] B. Weyrauch, B. Heisele, J. Huang, and V. Blanz, "Component-based face recognition with 3D morphable models," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPRW 2004, USA*, July 2004.
- [16] X. Lu, X. Lu, and E. Lansing, "Image analysis for face recognition. Personal Notes," <http://citeseerx.ist.psu.edu/viewdoc/download?>
- [17] BioID-Face-Database, <https://www.bioid.com/About/BioID-Face-Database>.
- [18] R. Gonzalez and R. Woods, *Digital image processing*, Prentice Hall, 2002.
- [19] O. R. Vincent and O. Folorunso, "A Descriptive Algorithm for Sobel Image Edge Detection," in *Proceedings of Informing Science & IT Education Conference (InSITE)*, p. 11, 2009.
- [20] G. N. Chaple, R. D. Daruwala, and M. S. Gofane, "Comparisons of Robert, Prewitt, Sobel operator based edge detection methods for real time uses on FPGA," in *Proceedings of the 2015 International Conference on Technologies for Sustainable Development, ICTSD 2015, India*, February 2015.
- [21] R. Maini and H. Aggarwal, "Study and comparison of various image edge detection techniques," vol. 3, 2009, <https://doi.org/>.
- [22] S. Cuillon and M. Donias, "Directional second order derivatives : Application to edge and corner detection," in *Proceedings of the 9th European Signal Processing Conference, EUSIPCO 1998, Greece*, September 1998.
- [23] Y. Wei-Bo and D. Zun-Sheng, "An improved Kirsch human face image edge-detection method based on canny lgorithm," in *Proceedings of the 2011 International Conference on Consumer Electronics, Communications and Networks, CECNet 2011*, pp. 4740–4743, China, April 2011.
- [24] H. Spontón and J. Cardelino, "A Review of Classic Edge Detectors," *Image Processing On Line*, vol. 5, pp. 90–123, 2015.
- [25] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, "Introduction to Algorithms-3rd Edition. Contemporary Sociology (Vol. 25)," <https://doi.org/10.2307/2077150>.
- [26] A. A. Alorf, "Performance evaluation of the PCA versus improved PCA (IPCA) in image compression, and in face detection and recognition," in *Proceedings of the 2016 Future Technologies Conference, FTC 2016*, pp. 537–546, USA, December 2016.
- [27] J. Kang, X. Lin, and G. Yang, "Research of multi-scale PCA algorithm for face recognition," in *Proceedings of the 2015 International Conference on Information and Communications Technologies (ICT '15)*, 2015.
- [28] D. S. D. Kumar and P. V. Rao, "Analysis and Design of Principal Component Analysis and Hidden Markov Model for Face Recognition," *Procedia Materials Science*, pp. 615–625, 2015.

- [29] K. Liu and S. Moon, "Robust dual-stage face recognition method using PCA and high-dimensional-LBP" in *Proceedings of the IEEE International Conference on Information and Automation (ICIA)*, 2016.
- [30] T. G. Clarkson, "Introduction to neural networks. Neural Network World," [https://doi.org/10.1016/S0140-6736\(95\)91746-2](https://doi.org/10.1016/S0140-6736(95)91746-2).
- [31] M. T. Hagan, H. B. Demuth, and M. H. Beale, "Neural Network Desig".

