Journal of Information Systems Engineering and Management
2025, 10(8s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Robust Algorithm for Deep Face Recognition System Performance Enhancement Based on Hybrid Techniques

Abdulbasit Alazzawi1, Burhan Al-Bayati2 , Zainab Mohammed Ali3
1,2,3 Department of Computer Science, University of Diyala, 32001 Diyala, Iraq

ARTICLE INFO

Received: 19 Oct 2024
Revised: 16 Dec 2024
Accepted: 28 Dec 2024

ABSTRACT

The effectiveness of most face recognition systems would be greatly reduced if the images in the dataset had inconsistent lighting conditions in uncontrolled environments. To address this problem, a new algorithm was developed that combines Modified Difference of Gaussian (MDoG) with Discrete Wavelet Transformation (DWT) and various edge detection operators, including first-order derivative filters (such as Sobel, Prewitt, and Roberts filters) and second-order derivative filters (such as Zero cross, LoG, and Canny filters). A features extractor based on Linear Regression Slope (LRS) and Principal Components Analysis (PCA), was used to extract features from the images. The accuracy of the algorithm was conducted with the Optimized Artificial Neural Network (OANN) for classification purpose and with different datasets including CNF, CK, JAFFE, and CAS. The experiment proves that this algorithm has greater efficiency than other modern methods, the best results are obtained with MDoG-DWT-SF, LRS, and OANN pairing and also MDoG-DWT-ZF, LRS, and OANN combination.

Keywords: Face recognition, Deep face, Enhancement, Hybrid methods.

INTRODUCTION

In the last several years, face recognition systems have become even more significant in our life as well as in the area of our security. Border control systems are systemic and are used to authenticate people in diverse applications for example; airports, tracking systems, identification of criminals and checkpoints[1, 2]. There are two main functions of face recognition: identifying individuals and verifying their identity. Identification involves matching one face to many stored face images, while verification is more complex, requiring the system to match the tested face against a large database of saved face images[3]. There are various techniques and approaches that can be used in face recognition, which it is possible to categorize this into two main methods.

The initial strategy is data reduction and feature extraction by holistic methods, such as utilizing eigenvectors based on Principal Component Analysis (PCA) [4]. The utilized approaches are Independent Component Analysis (ICA)[5, 6], Linear Discriminant Analysis (LDA), and Kernel LDA[7].

The second approach is used as a classifier to identify facial features that are most relevant through using techniques such as neural network-based approaches [8], support vector machines [9], and nearest distance algorithms [10]. While these techniques and approaches offer numerous benefits, they also have some drawbacks. For example, feature extraction may be time-consuming, training the dataset may require a significant response time, the use of outdated training datasets can be a problem, and hardware requirements may vary depending on the dataset being used[11-13].

PCA is a technique for linear feature extraction, it might not be efficient in situations where it is necessary to handle non-linear relationships [14]. To address this limitation, a new feature extraction method called LRS was proposed. The LRS algorithm uses the linear regression slope, which is calculated after image segmentation and an optimized artificial neural network (OANN) is used for classification[15]. However, training datasets can affect the generalization of the algorithm during testing, presenting another challenge to achieving efficient performance [16]. To address this issue, an enhancement algorithm was added to the OANN to improve its performance [17]. In this paper proposes a framework that was tested using four different face datasets with varying quality, illumination conditions, and occlusions. The methodology includes a modified difference of Gaussian, discrete wavelet transforms, gradient-based and Laplacian-based operators, segmentation, and feature extraction methods such as LRS and PCA. The results and discussion of the proposed algorithm are presented in Section 3, while Section 4 provides conclusions.

2. METHODOLOGY

2.1. Modified Difference of Gaussian MDoG Method

The scale-invariant features transform descriptor, SIFT, relies on the DoG technique to identify and extract features in image processing and landmark recognition [18]. In our proposed algorithm, we also use the DoG technique for feature extraction. Our algorithm begins with pre-processing using a modified DoG operator to address variations in illumination and lighting conditions. The next step involves generating first and second-order filter images using

Sobel, Prewitt, Roberts, Log, Zero cross, and Canny filters [7, 19]. We then reconstruct the face edges map using the appropriate threshold.

Algorithm explains the DoG.

Algorithm 1:

- Inputs: I face image,
- Output: Enhanced image

σ: small value, Σ: large value

γ: Correction values in an image output

I0: pre-processing face image

a. Image Synthesis if λ > 0 i1 = I.λ else
i2 = I end if

b. Differences of Gaussian
F1 = DoG (Gauss, σ)
F2= DoG (gauss, Σ)

I3 = filter (F1, i1) - filter (F2, i2)

c. CLAHE for contrast equalization

i3 = clahe (i3)

d. Image data normalization
I4=equation (1)

e. Return I0 = I4

In algorithm 1, Following the application of the DoG technique and contrast enhancement using CLAHE, we performed image normalize to address lighting conditions. Image normalization involves expanding the dynamic range of an image to bring it into a more familiar or normalized range for feature extraction and classification purposes [20, 21]. This process is important for face recognition and reduces the computational burden in subsequent steps.

Normalization transforms a dimensional grayscale image as [22].

I: {X ⊆ R} −−− > {Min,...,Max}

A new image can be created from a grayscale digital image that has intensity values between a minimum and maximum range, by adjusting the intensity values to a new range between newMin and newMax [17]. The process of linear normalization can be carried out using the following formula [23].

$$I_N = (I - Min)\frac{newMax - newMin}{Max - Min} + newMin \qquad (1)$$

After the completion of algorithm 1, the output is used as input for the edge detection stage, during which we obtained a matrix of feature values that were second-to-last in the sequence.

2.2. Proposed Discrete Wavelet Transformation (DWT)

DWT is a widely used technique in image processing for pattern recognition systems, particularly for non-stationary signals. By translating the signal into shifted and scaled levels. DWT provides good frequency resolution for low-frequency components, resulting in sub-signals or sub-bands such as LL, LH, HL, and HH [24]. In the proposed system, a multi-level DWT was applied to the image, and a low-pass filter was used on the rows and columns to obtain the LL sub band. The LL sub band represents a smaller scale than the original image, providing a general trend of the pixel values and ignoring small details [3, 24, 25]. Additionally, a high-pass filter was applied to the image to obtain three other sub-bands, LH, HL, and HH, which capture image details. The LL sub band takes the most prominent role because it allows getting the original image back, while the other sub-bands represent the image details. The wavelet was used in the proposed system and it was Daubechies wavelet; it was of a specific level as shown in equation (2) [26].

$$n = (2^{o} - 2^{3}db) \qquad (2)$$

Decomposition is required for extraction of the best feature vectors. The mathematical representation of performing the multi-level DWT is illustrated in the following equations: The mathematical representation of performing the multi-level DWT is illustrated in the following equations [27].

$I(i,j) = (LL) + (LH + HL + HH)$ (3)

$I(i,j) = (I_{D1}^{n}) + (I_{D1}^{n} + I_{H1}^{n} + I_{V1}^{n})$ (4)

This formula is used for (n) times DWT decompositions. In our study, we used equation (2).

-db" refers to the Daubechies discrete wavelet transform (DWT).

The levels of the DWT decomposition utilized in this algorithm were denoted in the following manner [25]:

$I(i,j) = (IA11) + (ID11 + IH11 + IV11)$ (5)

$I(i,j) = (IA21) + (ID21 + IH21 + IV21)$ (6)

$I(i,j) = (IA41) + (ID41 + IH41 + IV41)$ (7)

$I(i,j) = (IA81) + (ID81 + IH81 + IV81)$ (8)

$I(i,j) = (I_{A1}^{8}) + (I_{D1}^{8} + I_{H1}^{8} + I_{V1}^{8})$ (9)

2.3. Segmentation Process

The system we propose includes the breaking down of the face image into the same size blocks, the first step in the model. This method involves dividing and conquering problem that is the main strategy of problem-solving where data is processed using the complex algorithm of pattern recognition [20]. In this paper, we focus on the two main goals of separating the image into sub-images, which were achieved by the method that we used. The first goal was to accurately identify the locations of edge pixels, which is a crucial step in utilizing the LRS feature extraction method [15, 28]. The second objective was to differentiate the noise pixels from the edge pixels, streamlining the feature extraction process. There is no universal standard for the number of segments, so it is a flexible parameter. We experimented with different scenarios to discover an optimal number of segments since this parameter is linked to other parameters [29].

2.4. Features Extractors Methods

2.4.1. Eigen faces

To generate feature vectors for image recognition, the principal component analys -is (PCA) method projects the training image onto the basis of the image space, as outlined in Algorithm 2. The PCA approach typically categorizes dataset images by determining the distance between feature vectors [30]. Popular classification methods include the nearest distance measure. Euclidean distance, and nearest mean classification [31, 32].

Algorithm 2

- Inputs: Dataset Images
- Output: Features Vector

a. To begin, the data set should be structured in the following manner. Suppose we have (n) faces with m rows and m columns, such as:

$S = S_1, S_2, S_3, \ldots S_T$ (10)

The images should be represented in a column vector space with (m2 x 1) dimensions. As per the PCA algorithm, a mean face image (W), which represents common features for all datasets, must be calculated from the vectors using the following equation [33]:

$Q = \frac{1}{T}\sum_{n=1}^{m} S_n$ (11)

Here, "Q" denotes the mean matrix, which represents the shared characteristics of the faces.

b. The second step involves subtracting the primary data matrix, which can be expressed as

$L_n = S_n - Q$ (12)

c. Next, the Ln column vectors are collected in a matrix denoted as

R = [L1, L2, L3..., LN], which has dimensions (m2 x N). A covariance matrix C is then constructed as follows

$C = \frac{1}{T}\sum_{n=1}^{m} L_n L_n^T = AA^T$ (13)

Due to the high dimensionality (taking into account linear algebra principles), and the signi-

cant computational complexity of AA^T multiplication, we opted to use A^TA instead

$$C = \frac{1}{T}\sum_{n=1}^{m} L_n^T L_n = A^T A \quad (14)$$

d. During this stage, we calculate N eigenvalues Λp and N eigenvectors (VP) of C in order to create an Eigen faces space. The matrix V = [v1, v2, vN] represents the eigenvectors of matrix C, which has a size of N * N. The Eigen faces space U, represented as [u1, u2, uN] T, may be obtained by multiplying V and A^T.

e. Each row vector in U represents a "Eigen face" that corresponds to a face picture in the training set. Face pictures with larger eigenvalues have a greater impact on the Eigen face space [32]. Due to limited processing capacity, systems prioritize the sorting of eigenvectors of face pictures based on their associated eigenvalues in a descending order. They then select the first Z eigenvectors to create a smaller Eigen face space [16].

f. Matrix vector W is a matrix with dimensions (N*N) that contains N column vectors, each corresponding to a face picture in the training set. The vectors in question are referred to as "feature vectors" because they serve to reflect the unique qualities of each image. W may be derived using the following formula.

$$W = U.L \quad (15)$$

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

g. In step f, "ST" is a column vector that represents the test image with (m2 x 1) dimensions. At this point, the distance between the test image and the mean face image should be computed as the LT column vector.

$$L_T = S_T - Q \quad (16)$$

h. Once L_T has been calculated in step g, it is projected onto the Eigen faces pace to obtain its feature vectors in the form of a column vector W_T with dimensions (N x 1).

$$W_T = U.L_T \quad (17)$$

i. Step h involves identifying which image in the training set is most similar to our test image. The similarity of W_T is required to each wi in matrix W. Various classifiers can be used at this step. The techniques we used are modified artificial neural network.

2.4.2. Linear Regression Slope LRS

To extract the most beneficial features from edge images, a novel feature extraction technique is employed that relies on the gradient of the estimated curve for each small segment. This method, called Linear Regression Slope-based (LRS) feature extraction, produces feature vectors that are capable of successfully identifying each face [36]. The LRS method is based on the positions of edges and face pixels, rather than their values. Specifically, the coordinate of each edge pixel (i, j) is used as an index to find the corresponding face edge (1s) values in each segment, rather than the edge pixel values (0, 1). The slopes of the line are then calculated and saved in a matrix, as outlined in Algorithm 3 for the LRS algorithm.

Algorithm 3

- Inputs: Dataset Images
- Output: Features Vectors

a. Suppose that the data set includes N faces images with m row and n columns such that η1,η2,...,ηN represent dataset images in the column vector with [(m∗n)∗1] dimension.

b. Convert the images to the binary image such that β1,β2,...,βk represent the binary images.

c. another step is to segment the image (βi) into N segments with m rows and m columns

d. Calculate the coordinates of (1os) values in each segment and save it as (Xj, Y j) where (Xj, Y j) represent the vectors of coordinates of points in segment ψj.

e. Fit a line for the given (Xj, Y j) points of ψj as :

$$y = \zeta_j * x + b \quad (18)$$

f. Find the slope of lines ι1,ι2,...,ιn in each segment to all faces such that

$$\zeta_j^i = \frac{\left(\left(\sum_{k=1}^{k_j^i}(y_k)(\sum_{k=1}^{k_j^i}(x_k^2))(\sum_{k=1}^{k_j^i}(x_k))(\sum_{k=1}^{k_j^i}(x_k^2 y_k^2)\right)\right)}{(\sum_{k=1}^{k_j^i}(x_k^2))(\sum_{k=1}^{k_j^i}(x_k))} \quad (19)$$

ζ_j^i: Slope for the j^{th} segment of i^{th} face sample at(i,j).

K_j^i: Total number of edge pixels for the j^{th} segment of i^{th} face sample.

g. In step f, the slopes of the segments are assembled in a matrix with dimensions (m x m), which comprises N column vectors corresponding to each face in the training set, to create feature vectors:

$$Z = [\zeta_{j1}^{i}, \zeta_{j2}^{i}, \dots, \zeta_{jN}^{i}, (\zeta_{j}^{i})_{N1}, (\zeta_{j}^{i})_{N2}, \dots, (\zeta_{j}^{i})_{Nk} \ ] \quad (20)$$

h. After obtaining the face features vectors in space, the comparison between a probe image to the training data images occurs by the following steps:
i. In step h, a testing image βT is segmented in the same manner as the training image, such that the number of segments in the testing image is identical to the number of segments in the training image βi.

j. Find the slopes of all the segments of the testing image.
k. Create a feature vector matrix of the testing face.
l. Step j involves determining which image in the training image set is most similar to our probe image. This is accomplished by computing the similarity of the image βT with each image in the training dataset.

THE PROPOSED CLASSIFIER

The final step of the proposed system entails the training dataset. The input matrix for the training dataset comprises database images that have undergone pre-processing operations, dimensionality reduction, and feature extraction. To overcome the limitations of existing NP-complete methods, an Optimized Artificial Neural Network (OANN), based classifier is proposed in this work. The classifier is designed to enhance accuracy, efficiency, and the time required for complex tasks. Evolutionary techniques are increasingly being employed in the training of artificial neural networks (ANNs). Although these approaches have demonstrated superior performance to traditional training methods, optimizing both network performance and architecture can result in slower training processes due to increased algorithmic complexity [37, 38]. To address this issue, emerging artificial neural networks (EANNs) have been developed in which adaptation is primarily achieved through evolution. EANNs offer numerous advantages over traditional MANN training methods. The design of an ANN necessitates balancing two conflicting objectives: increasing network capacity and minimizing neural design complexity. As a result, multi-objective evolutionary algorithms have shown significant success in simultaneously optimizing both architecture and connection weights . In this work, algorithms 4 and 5 are utilized for the training and testing procedures.

Algorithm 4

- Input: dataset images
- Output: trained OANN

a. The input feature matrix M from T is read at layer 1Fi.

b. The activation value for each neuron ANi is measured.

c. The neuron with the highest ANi value is located.
d. The results of step c are extracted, along with its input ID and the index of the maximum ANi.

e. If output = Ok is set to 1 for the k-th neuron with the highest ANi value.

f. Otherwise, the output is set to 0.

g. The input of the previous layer is fed to the next layer and eventually to the output layer.

h. The above steps are repeated for all input layers.

Algorithm 5

- Input: Testing Dataset Images
- Output: Classification Results

a. The test pattern to be recognized or classified is read.
b. The activation value ANi is calculated during layer 2.
c. The neuron with the highest ANi is selected.
d. The neuron with the maximum ANi index is extracted and saved as input. id and maxANi-index for matching purposes.
e. If a match is successful, the input-id of the max-ANi-index is returned as output.
f. The process is stopped.

RESULTS AND DISCUSSION

The suggested approach underwent investigation, which included the utilization of MDoG, DWT, and two sets of edge detection filters: gradient-based operators (Sobel, Prewitt, and Roberts) and Laplacian-based operators (LG, ZF, and CF). Feature extraction was conducted using both eigenvector-based and LRS approaches, and classification was carried out using the OANN classifier. Four training sets were created for the PCA and LRS

feature extraction algorithms. Each training set included changes in lighting, expression, and posture parameters for each subject. The experimental study was performed utilizing the MATLAB image processing toolkit on four independent face datasets. Each dataset was created under varied settings, including variations in picture quality, lighting conditions, occlusions, and other aspects. The suggested technique was assessed against state-of-the-art methodologies, namely LDN and LDN+DWT, and two modifications of the proposed method were examined in simulation results. In order to evaluate its efficacy, the suggested method's performance of the trained model has been evaluated and assessed using four scores

- True Positive (TP): is the quantity of samples that were accurately identified as face.
- False Positive (FP): represents the quantity of samples that incorrectly classified non-face as face.
- True Negative (TN): refer the quantity of samples that correctly predicts as non-face.
- False Negative (FN): indicates the quantity of samples that incorrectly identified face as non-face.

The proposed models was evaluated using five cutting-edge performance measures, namely accuracy, sensitivity, specificity, precision, and recall.
In order to assess the proposed method's efficiency, performance metrics are used. These are the metrics that can be calculated using many metrics [14, 39].

Metrics.

$$Accuracy = \frac{\mathrm{TP}+TN}{\mathrm{TP}+TN+FP+FN} \quad (21)$$

$$Sensitivity = \frac{TP}{TP+FN} \quad (22)$$

$$Specificity = \frac{TN}{TN+FP} \quad (23)$$

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP}+FP} \quad (24)$$

$$Recall = \frac{TN}{TN+FP} \quad (25)$$

This research compares the performance of all the approaches using four datasets. The findings are highlighted through the use of graphs and tables. Table 1 displays the recognition accuracy for different datasets, demonstrating that the proposed hybrid method versions surpass state-of-the-art approaches such as LDN and LDN+DWT for every dataset. The hybrid approach, when used with SLR, is highly efficient in enhancing recognition outcomes by minimizing extraneous characteristics.

Table 1: Recognition accuracy rate analysis for all datasets

	LFW	CK	CAS	JAFEE
Technique LDN [37]	94.21	83.33	86.85	59.81
LDN [37] +DWT	90.37	90.67	71.97	64.96
MDoG-DWT-SF-PCA	96.78	95.67	88.91	**92.07**
MDoG-DWT-SF-SLR	95.09	94.69	86.91	**98.81**

The use of the SLR results in significant variations across the various training courses, which ultimately enhances the performance of the suggested technique. The JAFEE dataset has a wide range of face expressions and changes, which leads to a decline in the recognition performance of all techniques. Nevertheless, the suggested approach still managed to get a satisfactory level of accuracy in recognizing the JAFEE dataset. Furthermore, the new approaches exhibited comparable outcomes to current methods across many performance parameters, in addition to accuracy. The precision and recall rates exhibited divergent outcomes for the LFW and JAFEE datasets, which were contrary to the accuracy performance. The proposed strategy consistently enhanced the accuracy and recall rates. The LFW dataset had inferior specificity performance, whilst the JAFEE dataset yielded unsatisfactory specificity findings. The figures

(1 to 4), exhibit the resilience of the suggested face recognition algorithms when applied to various facial research datasets. The findings are displayed in a tabular format below.

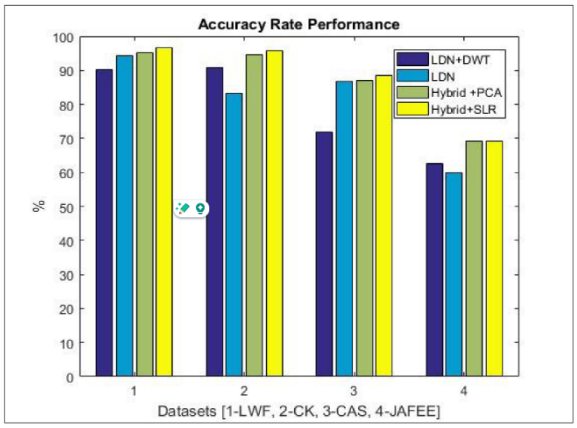

Figure 1: Recognition accuracy performance evaluation

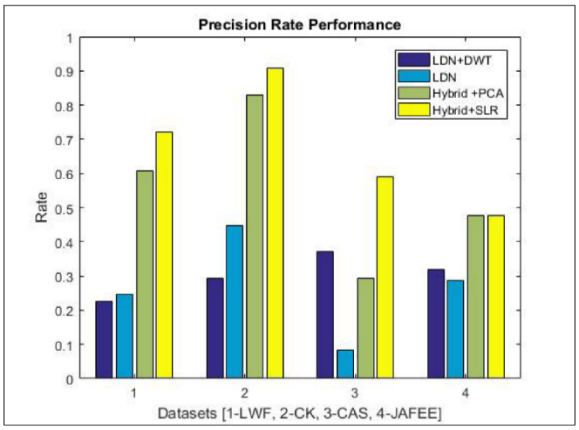

Figure 2: Precision rate performance evaluation

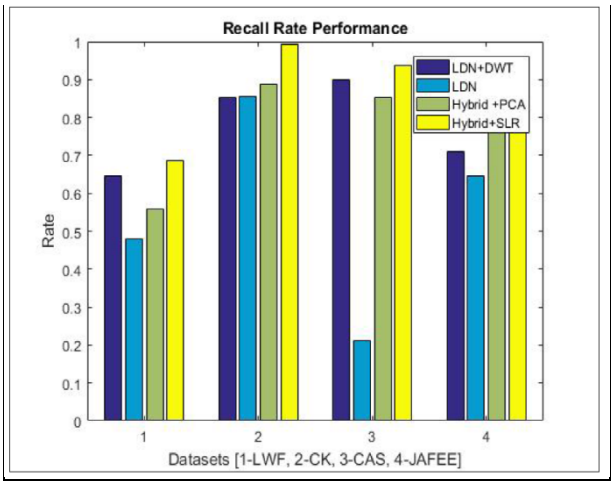

Figure 3: Recall rate performance evaluation

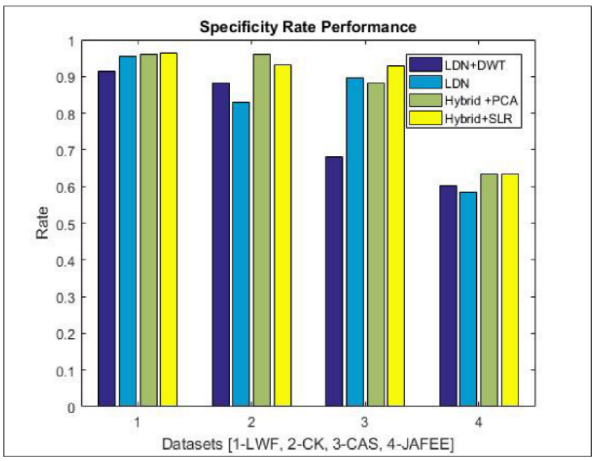

Figure 4: Specificity rate performance evaluation

As stated in reference [37], the LDN technique demonstrates an average performance outcome of 81.05% across all datasets. Nevertheless, employing the DWT feature extraction approach results in a decline in the recognition rate. The reason for this is that DWT characteristics are effective at extracting various changes in the facial image's position. In contrast, the suggested technique utilizes MDoG, DWT, and edge features in conjunction with the proposed strategy, resulting in a significant enhancement in the identification rate. This improvement amounts to an estimated increase of 8-9% across all datasets. This result aligns with the intended aims and goals of the suggested technique. Table 2 provides a study of accuracy performance in comparison to recently examined facial recognition techniques. The datasets included in this work encompass a variety of lighting conditions, facial expressions, unobstructed head poses, and occlusion scenarios. When viewed in this particular situation, the outcomes of the suggested approach are extremely meaningful and surpass all current approaches for recognizing faces.

To further explain, the correct classification rate is practically recorded in the experiments of the proposed algorithm, which were tested based on the following parameters. Firstly, six edge detection operators were used based on their sensitivity to noise, which was measured by the signal-to-noise ratio (SNR). Secondly, a range of threshold values from 0.01 to 0.9 was used since the standard values of the operators were not perfect in these experiments. Thirdly, different numbers of hidden units were used in the OANN layers, specifically 25, 50, 75, 100, 125, and 150, to prevent overfitting and improve generalization. Finally, several segment scenarios were performed, such as 4, 6, 8, 10, and 100, to precisely treat each part of the image according to the principle of divide and conquer in problem-solving.

The performance criteria used to measure the performance of the compared algorithms is the correct classification rate (CCR), which computes the average number of correctly classified faces. Tables 3 and 4 show the CCR for each experiment. The dataset was divided as follows (i): 40% for training, 30% for validation, and 30% for testing. (ii): 60% for training, 20% for validation, and 20% for testing. (iii): 50% for training, 30% for validation, and 20% for testing. and (vi): 50% for training, 20% for validation, and 30% for testing.

Table 2: State-of-art methods evaluation

Reference	Year and Dataset	MAX Accuracy
[33]	2015 (AR-illumination)	89.53
[34]	2016 (AR)	90.40
[35]	2016 (Yale)	82.61
[36]	2017 (LFW)	94.30
[37]	2017 (FRGC 2.0)	95.20
MDoG-SF-DWT-OANN	**LFW**	**98.87**

Table 3: Correct Classification Rate CCR results based on a hybrid of MDoG, DWT, First Order Derivative Operators, PCA, SLR, and OANN.

Testing Rate (Technique)	15%	20%	30%	40%	Weighted Mean of CCR
MDoG-DWT-SF-LRS-OANN	100.0000	998.5620	99.1030	97.5821	**98.92112**
MDoG-DWT-SF-PCA-OANN	94.7140	91.8130	88.9430	92.618	**98.8117**
MDoG-DWT-PF-LRS-OANN	93.4720	96.7320	91.8140	90.8420	92.0220
MDoG-DWT-PF-PCA-OANN	91.5040	91.7530	90.7810	88.9410	93.2145
MDoG-DWT-RF-LRS-OANN	91.6910	77.8410	50.8930	49.7840	90.7445
MDoG-DWT-RF-PCA-OANN	88.6030	83.9140	77.6170	53.3810	67.5522

Table 4: Correct Classification Rate CCR results based on a hybrid of MDoG, DWT, Second Order Derivative Operators, PCA, SLR, and OANN.

Testing Rate Technique	15%	20%	30%	40%	Weighted Mean
MDoG-DWT-ZF-PCA -OANN	100.0000	98.5420	95.751	93.2150	**96.8770**
MDoG-DWT-ZF-LRS-OANN	94.9420	86.7410	91.8140	92.9500	91.2717
MDoG-DWT-LF-PCA-OANN	98.6180	94.7320	88.5190	81.438	90.8267
MDoG-DWT-LF-LRS-OANN	90.4170	91.8430	88.2470	83.7290	88.5725
MDoG-DWT-CF-PCA-OANN	94.3180	94.7900	83.7640	76.6150	82.4330
MDoG-DWT-CF-PCA-OANN	96.5180	90.5910	70.981	71.6420	**96.8770**

As evidenced by the results, the number of segments used to divide the face images can significantly impact the algorithm's performance. The best number of segments is 10, and deviations from this value can critically affect the behaviour of the proposed system. Therefore, in all experiments, the face images are divided into 10 segments horizontally and vertically, resulting in 100 segments. The second most important parameter is the optimal number of hidden layers required to efficiently classify complex scenarios.

To determine this parameter, an experiment was conducted on different algorithms to extract the best value. Apart from the number of neurons and segments, the threshold values of the edge detection filters also play a significant role in the resulting binary edge image and classification process. To accurately select these thresholds, a separate experiment was performed for each of the six considered edge detection filters. Figures (5-10), show the results of using different threshold values for each edge detection method. The best threshold value that maximizes performance is selected for each edge filter. . Figure 11 shows the effects of using a different number of segments on

the performance of our proposed algorithm. Figure 12 shows the results of this experiment, indicating that the best number of neurons to use.

Figure 5: Performance of proposed System based on hybrid MDoG, Sobel, DWT, and LRS

Tables 3 and 4 present the results of comparing different combinations of the proposed algorithm using public databases. Table 3 shows the results of using the SF, PF, and RF for both SLP and PCA feature extraction methods, while Table 4 presents the results of using the LF, Zero CF, and CF for both SLP and PCA feature extraction methods.

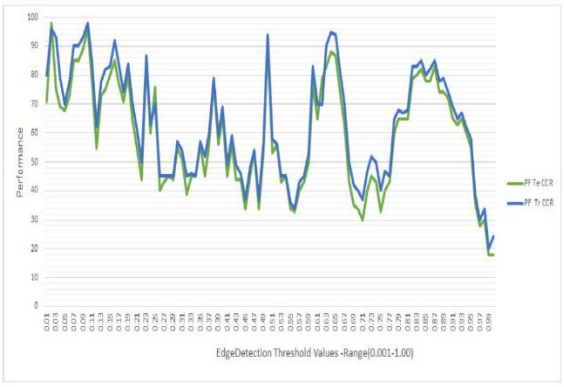

Figure 6: Performance of proposed System based on hybrid MDoG , Prewitt, DWT, and LRS

Figure 7: Performance of proposed System based on hybrid MDoG, Roberts , DWT, and LRS

For all algorithms, the OANN is used. The algorithms that use the SLP method outperform the other PCA versions, with the MDoG-SF, DWT, SLP, and OANN algorithm being the best, achieving optimal mean and valuable values in two different percentages of the training data

Figure 8: Performance of proposed System based on hybrid MDoG, Zero cross , DWT, and LRS

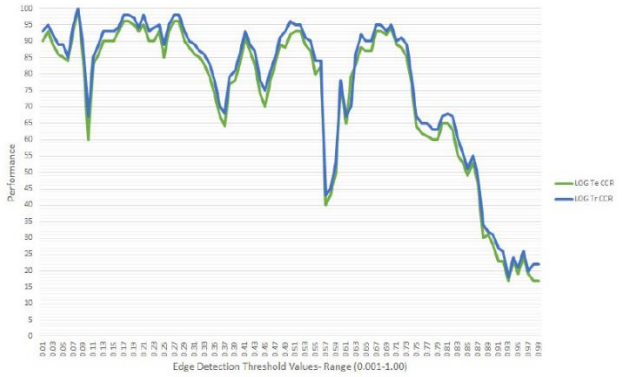

Figure 9: Performance of proposed System based on hybrid MDoG, LoG, DWT, and LRS

Figure 10: Performance of proposed System based on hybrid MDoG, Canny , DWT, and LRS

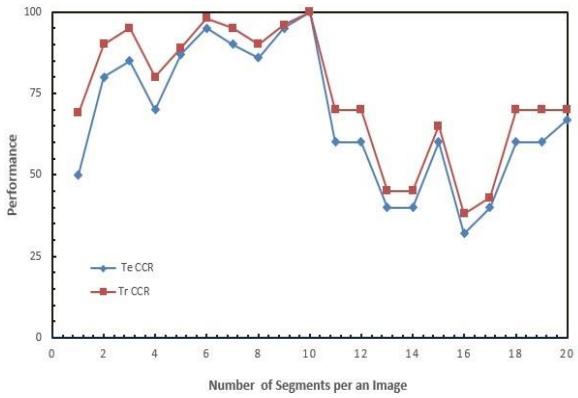

Figure 11. Distribution of Segments per image

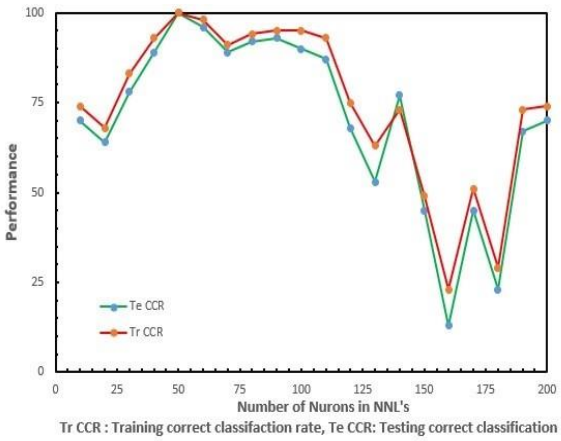

Figure 12. Distribution of Neurons in the Layers of OANN

. The best algorithm in Table 4 is the MDoG-ZF DWT LRS OANN, achieving a correct classification metric of 96.8%. The results confirm the performance of the proposed combination of MDoG and edge operators with the new proposed LRS feature extraction method and OANN.

CONCLUSIONS

This research paper presents a new framework for face recognition that addresses the challenges of uncontrolled face recognition, such as varying illumination conditions, occlusions, low-resolution images, and sunglasses, among others. While several methods have been reported for controlled conditions-based face recognition, very few methods have been reported for uncontrolled conditions. The proposed framework uses a hybrid face representation method that combines the modified difference of the Gaussian method, discrete wavelet transforms, edge operators, linear regression slope for feature extraction, and optimized artificial neural networks for classification.

The series of the experiments was devoted to the search of the best parameters for each of the algorithms. Another crucial parameter of this algorithm is the number of minutes in each data block, which is the main part of the proposed algorithm. Proposed SLP method includes the creation of a positions matrix of pixel values which further calculates the slope of each point the SLP algorithm is capable to be resistant towards environmental factors such as light conditions and complex backgrounds. Another convenience is that this technique doesn't necessitate complicated normalization processes, such as dimensionality reduction, increasing the length of execution time.

The research also suggests that 2nd order filters like LF, ZF, and CF are more effective when it comes to binary image representation for face recognition than 1st order filters such as SF, PF, and RF. The algorithm was tested by experimenting with several techniques on the datasets such as LFW, CK, Jaffee and CAS in the training and testing sections and shows significant improvement over the state-of-the-art methods.

The research paper's concluding part is devoted to a detailed comparison of our proposed method that has demonstrated its efficacy, while experimental results show the highest accuracy rates as well as good performance of DWT-SF-LRS-OANN combination, which outperforms other algorithms. Further research can take place where other classifier options are being used on the proposed method.

REFERENCES

[1] D. S. Trigueros, L. Meng, and M. Hartnett, “Face Recognition: From Traditional to Deep Learning Methods,” 2018, [Online]. Available: http://arxiv.org/abs/1811.00116

[2] A. A. Valke and D. G. Lobov, “Evaluation of Face Recognition Algorithms Under Noise,” *J. Phys. Conf. Ser.*, vol. 1210, no. 1, 2019.

[3] S. Sayed, M. Nassef, A. Badr, and I. Farag, “Building an ensemble feature selection approach for cancer microarray datasets using different classifiers,” *Int. J. Intell. Eng. Syst.*, vol. 12, no. 4, pp. 50–61, 2019, doi: 10.22266/ijies2019.0831.06.

[4] W. Wen, “Efficient and Scalable Deep Learning Efficient and Scalable Deep Learning,” no. May, 2019.

[5] S. Sande and M. L. Privalsky, “Identification of TRACs (T3 receptor-associating cofactors), a family of cofactors that associate with, and modulate the activity of, nuclear hormone receptors,” *Mol. Endocrinol.*, vol. 10, no. 7, pp. 813–825, 1996, doi: 10.1210/me.10.7.813.

[6] M. Sivaram, V. Porkodi, A. S. Mohammed, and ..., “Detection of Accurate Facial Detection Using Hybrid Deep Convolutional Recurrent Neural Network.,” *Ictact J. Soft Comput.*, pp. 1844–1850, 2019, doi: 10.21917/ijsc.2019.0256.

[7] A. J. Moshayedi, A. S. Roy, A. Kolahdooz, and Y. Shuxin, “Deep Learning Application Pros And Cons Over Algorithm,” *EAI Endorsed Trans. AI Robot.*, vol. 1, pp. 1–13, 2022, doi: 10.4108/airo.v1i.19.

[8] K. T. Khudhair, F. H. Najjar, S. R. Waheed, H. M. Al-Jawahry, H. H. Alwan, and A. Al-Khaykan, “A novel medical image enhancement technique based on hybrid method,” *J. Phys. Conf. Ser.*, vol. 2432, no. 1, 2023, doi: 10.1088/1742-6596/2432/1/012021.

[9] U. Scherhag, C. Rathgeb, J. Merkle, and C. Busch, “Deep Face Representations for Differential Morphing Attack Detection,” *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 3625–3639, 2020, doi: 10.1109/TIFS.2020.2994750.

[10] A. Alagah Komlavi, K. Chaibou, and H. Naroua, “Comparative study of machine learning algorithms for face recognition,” *Rev. Africaine Rech. en Inform. Mathématiques Appliquées*, vol. Volume 40, 2024, doi: 10.46298/arima.9291.

[11] G. Zhang, “Face Recognition based on Fuzzy Linear Discriminant Analysis,” *IERI Procedia*, vol. 2, pp. 873–879, 2012, doi: 10.1016/j.ieri.2012.06.185.

[12] H. Fan, Z. Cao, Y. Jiang, Q. Yin, and C. Doudou, “Learning Deep Face Representation,” pp. 1–10, 2014, [Online]. Available: http://arxiv.org/abs/1403.2802

[13] S. Venkatesh, R. Ramachandra, K. Raja, and C. Busch, “Face Morphing Attack Generation and Detection: A Comprehensive Survey,” *IEEE Trans. Technol. Soc.*, vol. 2, no. 3, pp. 128–145, 2021, doi: 10.1109/tts.2021.3066254.

[14] H. Benradi, A. Chater, and A. Lasfar, “A hybrid approach for face recognition using a convolutional neural network combined with feature extraction techniques,” *IAES Int. J. Artif. Intell.*, vol. 12, no. 2, pp. 627–640, 2023, doi: 10.11591/ijai.v12.i2.pp627-640.

[15] I. J. Jacob and P. E. Darney, “Design of Deep Learning Algorithm for IoT Application by Image based Recognition,” *J. ISMAC*, vol. 3, no. 3, pp. 276–290, 2021, doi: 10.36548/jismac.2021.3.008.

[16] R. I. Bendjillali, M. Beladgham, K. Merit, and A. Taleb-Ahmed, “Illumination-robust face recognition based on deep convolutional neural networks architectures,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 18, no. 2, pp. 1015–1027, 2020, doi: 10.11591/ijeecs.v18.i2.pp1015-1027.

[17] R. Ogla, A. A. Saeid, and S. H. Shaker, “Technique for recognizing faces using a hybrid of moments and a local binary pattern histogram,” *Int. J. Electr. Comput. Eng.*, vol. 12, no. 3, pp. 2571–2581, 2022, doi: 10.11591/ijece.v12i3.pp2571-2581.

[18] O. S. Ekundayo and S. Viriri, “Facial Expression Recognition: A Review of Trends and Techniques,” *IEEE Access*, vol. 9, pp. 136944–136973, 2021, doi: 10.1109/ACCESS.2021.3113464.

[19] A. Singer, “Spectral independent component analysis,” *Appl. Comput. Harmon. Anal.*, vol. 21, no. 1, pp. 135–144, 2006, doi: 10.1016/j.acha.2006.03.003.

[20] J. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, “Face recognition using kernel direct discriminant analysis algorithms,” *IEEE Trans. Neural Networks*, vol. 14, no. 1, pp. 117–126, 2003, doi: 10.1109/TNN.2002.806629.

[21] J. Jia, W. Liang, and Y. Liang, “A Review of Hybrid and Ensemble in Deep Learning for Natural Language Processing,” 2023, [Online]. Available: http://arxiv.org/abs/2312.05589

[22] L. Tao and M. Xueqiang, “Hybrid Strategy Improved Sparrow Search Algorithm in the Field of Intrusion Detection,” *IEEE Access*, vol. 11, no. April, pp. 32134–32151, 2023, doi: 10.1109/ACCESS.2023.3259548.

[23] S. F. Ahmed *et al.*, *Deep learning modelling techniques: current progress, applications, advantages, and challenges*, vol. 56, no. 11. Springer Netherlands, 2023. doi: 10.1007/s10462-023-10466-8.

[24] K. Santoso and G. P. Kusuma, “Face Recognition Using Modified OpenFace,” *Procedia Comput. Sci.*, vol. 135, pp. 510–517, 2018, doi: 10.1016/j.procs.2018.08.203.

[25] P. S. M. Gowda and D. H. N. Suresh, “Facial Expression Recognition using Robust Algorithm based on Modern Machine Learning Technique,” *Int. J. Eng. Adv. Technol.*, vol. 11, no. 5, pp. 30–39, 2022, doi: 10.35940/ijeat.e3535.0611522.

[26] P. Modi and S. Patel, “A State-of-the-Art Survey on Face Recognition Methods,” *Int. J. Comput. Vis. Image Process.*, vol. 12, no. 1, pp. 1–19, 2021, doi: 10.4018/ijcvip.2022010101.

[27] S. Hosgurmath, V. V. Mallappa, N. B. Patil, and V. Petli, “A face recognition system using convolutional feature extraction with linear collaborative discriminant regression classification,” *Int. J. Electr. Comput. Eng.*, vol. 12, no. 2, pp. 1468–1476, 2022, doi: 10.11591/ijece.v12i2.pp1468-1476.

[28] H. Liu *et al.*, “Development of a Face Recognition System and Its Intelligent Lighting Compensation Method for Dark-Field Application,” *IEEE Trans. Instrum. Meas.*, vol. 70, 2021, doi: 10.1109/TIM.2021.3111076.

[29] H. Ge, Y. Dai, Z. Zhu, and B. Wang, “A robust face recognition algorithm based on an improved generative confrontation network,” *Appl. Sci.*, vol. 11, no. 24, 2021, doi: 10.3390/app112411588.

[30] M. Taha, T. Mostafa, and T. Abd El-Rahman, “A Novel Hybrid Approach to Masked Face Recognition using Robust PCA and GOA Optimizer,” *Sci. J. Damietta Fac. Sci.*, vol. 0, no. 0, pp. 0–0, 2023, doi: 10.21608/sjdfs.2023.222524.1117.

[31] M. T. H. Fuad *et al.*, “Recent advances in deep learning techniques for face recognition,” *IEEE Access*, vol. 9, pp. 99112–99142, 2021, doi: 10.1109/ACCESS.2021.3096136.

[32] A. Rajagopalan, “The Repository at St . Cloud State Real-Time Deep Learning-Based Face Recognition System,” 2022.

[33] E. E. Cureton and R. B. D’Agostino, “Component Analysis,” *Factor Anal.*, vol. 13, no. 6, pp. 296–338, 2019, doi: 10.4324/9781315799476-12.

[34] I. Almomani, W. El-Shafai, A. AlKhayer, A. Alsumayt, S. S. Aljameel, and K. Alissa, “Proposed Biometric Security System Based on Deep Learning and Chaos Algorithms,” *Comput. Mater. Contin.*, vol. 74, no. 2, pp. 3515–3537, 2023, doi: 10.32604/cmc.2023.033765.

[35] M. Ghasemi and H. Hassanpour, “A three-stage filtering approach for face recognition using image hashing,” *Int. J. Eng. Trans. B Appl.*, vol. 34, no. 8, pp. 1856–1864, 2021, doi: 10.5829/ije.2021.34.08b.06.

[36] H. Ge, Y. Dai, Z. Zhu, and B. Wang, “Robust face recognition based on multi-task convolutional neural network,” *Math. Biosci. Eng.*, vol. 18, no. 5, pp. 6638–6651, 2021, doi: 10.3934/mbe.2021329.

[37] T. Tuncer, S. Dogan, M. Abdar, and P. Pławiak, “A novel facial image recognition method based on perceptual hash using quintet triple binary pattern,” *Multimed. Tools Appl.*, vol. 79, no. 39–40, pp. 29573–29593, 2020, doi: 10.1007/s11042-020-09439-8.