Asia Pacific Academy of Science Pte. Ltd.

Journal of Biological Regulators and Homeostatic Agents 2026, 40(1), 8285.
https://doi.org/10.54517/jbrha8285

Article

Transforming clinical diagnostics: A deep learning approach for biometric face recognition using multistage regression capsule networks and optimized grey wolf algorithm

Ashlin Jenitha Justin Joseph Samuel Rani[1,*], Thankamony Saradhadevi Sivarani[2]

[1] Department of Information Technology, St.Joseph's College of Engineering, Chennai 600119, India
[2] Department of Electrical and Electronics Engineering, Arunachala College of Engineering for Women, Vellichanthai, Tamilnadu 629203, India
*** Corresponding author:** Ashlin Jenitha Justin Joseph Samuel Rani, ashlinjenitha@outlook.com

CITATION

Rani AJJJS, Sivarani TS. Transforming clinical diagnostics: A deep learning approach for biometric face recognition using multistage regression capsule networks and optimized grey wolf algorithm. Journal of Biological Regulators and Homeostatic Agents. 2026; 40(1): 8285. https://doi.org/10.54517/jbrha8285

ARTICLE INFO

Received: 27 October 2025
Revised: 4 January 2026
Accepted: 6 February 2026
Available online: 27 March 2026

COPYRIGHT

Copyright © 2026 by author(s). *Journal of Biological Regulators and Homeostatic Agents* is published by Asia Pacific Academy of Science Pte. Ltd. This work is licensed under the Creative Commons Attribution (CC BY) license. https://creativecommons.org/licenses/by/4.0/

Abstract: Face recognition (FR) technology is increasingly being used in clinical diagnostics and customized medicine, in addition to typical security applications. However, reliably identifying patients based on face traits in big and heterogeneous datasets remains a major difficulty. The study proposed a novel framework, the Multistage Regression Capsule Network with Modified Grey Wolf Optimization (MRCN-MGWO), to improve the accuracy and efficiency of patient identification in healthcare settings. The MRCN-MGWO model uses deep learning to increase diagnostic accuracy by assessing facial features and using specialized preparation techniques for medical photos. Facial images are first denoised using a median filter (MF) before being enhanced with contour-based image enhancement (EIC) to improve clarity. The Multistage Regression Capsule Network (MRCN) generates robust feature vectors to detect distinct facial patterns, whereas the Modified Grey Wolf Optimization (MGWO) approach optimizes the weights and biases of a stacked autoencoder (SAE). The MRCN-MGWO architecture is tested on the benchmark FEI dataset and shows promise for accurate patient identification in a variety of clinical settings by outperforming current face recognition techniques. As a result, the proposed MRCN-MGWO model improves precision and increases the rate of facial recognition while maintaining high accuracy.

Keywords: face recognition, modified grey wolf optimization, multistage regression capsule network, medical diagnostics, clinical imaging, biometric patient identification, deep learning in healthcare

1. Introduction

FR technology is a rapidly evolving biometric tool that identifies and verifies individuals based on unique facial features. Initially developed for security and surveillance applications, FR has expanded into various fields, including healthcare, banking, and personalized user experiences. The technology relies on advanced AI and deep learning algorithms to analyze facial patterns, extract distinctive features, and match them against stored datasets. Recent advancements have improved its accuracy, robustness, and adaptability, making it a valuable tool in clinical diagnostics and patient identification. Deep learning techniques, such as capsule networks and autoencoders, have greatly increased the ability to extract, learn, and classify facial information with little human interaction. Unlike traditional machine learning approaches, deep learning automatically detects complex patterns in images, making it more resilient to variations in lighting, pose, and occlusions. In healthcare, deep

learning-based face recognition is increasingly used for patient identification, medical record management, and even early disease detection through facial feature analysis.

The integration of optimization algorithms further enhances model performance by fine-tuning network parameters, ensuring higher accuracy and faster processing. Big data is a word that has just been created to describe data that has been gathered in enormous quantities and at a rapid rate. Today, authorities, businesses, and academics are focusing on big data. However, because big data has multiple sources, a large volume, and a high rate of change, managing traditional data processing methods like data mining can be challenging. For big data application to run more efficiently, a robust big data processing structure is required.

Face recognition is a versatile biometric recognition technology. The necessity for a dependable facial recognition strategy in a large data environment is crucial, as present technologies fall short of the requirements for face recognition in that situation [1]. CNN is one of the most extensively used deep learning approaches for object recognition and identification. Convolutional neural networks are one of the most important components of deep learning. They provide distinct advantages in image processing due to convolutional neural networks' weight sharing and other aspects, such as the local connection. CNN training has a direct impact on the overall recognition rate and effectiveness of network training. The original data set is utilized to develop the model, resulting in a significant increase in recognition rate [2], yet it still accurately depicts the usefulness of the upgraded method.

MagFace will learn unified face recognition and quality evaluation characteristics. MagFace's broad architecture can be modified to accommodate a wide range of categorization applications, including human re-identification and fine-grained item recognition. Furthermore, the proposed technique of analyzing feature magnitude enables quality estimation for various objects, like as a person's body in Reid or a quick action in activity categorization. [3].

Face recognition using CNN performs well because to the massive amount of data accessible. Conventional recognition algorithms, such as SVM, can only extract surface image attributes that are quickly influenced by other factors, resulting in a low identification rate. Deep learning systems, such as CNN, may extract conceptual and abstract information with great depth. As the sample size increased, so did the distance between the two models and their recognition rates. LeNet-5's recognition accuracy was 87.1% with a sample size of 4000, while the updated LeNet-5's was 97.9% [4]. The face recognition system can execute the recognition function in most lighting conditions due to its 400FPS speed, 97.25% recognition rate, and high resilience, according to software simulation and board measurement [5].

The concept of a grey wolf so EICty serves as the foundation for one of the most recent optimization algorithms, GWO. When compared to other nature-inspired algorithms, GWO has shown promising outcomes for many nonlinear test functions and it has been effectively used in a number of application domains, including image processing and data mining. In GWO, the best answer is discovered using the wolves' collective behavior. Like a grey wolf would do when hunting, it first explores the search space before gradually exploiting it. Exploration and exploitation must always be balanced in an optimization algorithm. GWO exhibits good computational efficiency, but occasionally it experiences the drawback of getting stuck in local

optima because the exploration and exploitation phases are not balanced properly. The Modified Grey Wolf Optimizer does not include these variations. Also used for the identification and classification of faces is modified grey wolf optimization (MGWO), where the GWO algorithm selects the stacked autoencoder model's weight and bias parameters. An extensive result analysis of a benchmark dataset is conducted to demonstrate the superiority of the MRCN-MGWO method. The innovation is in combining MRCN with MGWO-optimized SAE to improve patient identification accuracy and facial feature learning in clinical face recognition. Here are the primary steps of the proposed approach for further contributions:

- The proposed MRCN-MGWO model to enhance accuracy and efficiency in patient identification using facial recognition in healthcare settings.
- Utilizes MF for noise reduction and contour-based image enhancement (EIC) for improved facial image clarity.
- Employs the MRCN to generate robust feature vectors for distinguishing unique facial patterns.
- Enhances the performance of the SAE by optimizing its weights and biases using the Modified Grey Wolf Optimization MGWO algorithm.
- Demonstrates improved accuracy and efficiency over existing face recognition techniques when evaluated on the FEI dataset.

The following describes the way the paper is established: Section 1 illustrates the introduction. Section 2 provides a description of the relevant works. Section 3 describes the proposed techniques; Section 4 presents the experiment's results; and Section 5 summarizes the study's findings and plans for further research.

2. Related works

The adoption of big data, machine learning, and sensing technologies in contemporary animal farming is growing with the rise of agriculture 4.0. In pandemic settings, where constraints make It is challenging for farmers, nutritionists, and veterinarians to visit the feed mills, barns, and farms in real time, round-the-clock insights on the behavior, consumption, and output of the animals are necessary. These perceptions made possible by sensing technology led to remotely accessible data that improves performance and reduces expenses in response to client demands. While AI and ML algorithms are evolving quickly, there are no global standards for data collection or exchange. Artificial intelligence (AI) and sensing technologies are expected to become increasingly important in helping farmers find solutions to pressing issues in contemporary animal husbandry as more farms become tech-connected [6]. We choose individuals with the same gender and race as negative pairs in order to reduce the attributes difference [7].

For facial identification, low resolution local binary patterns are used. Its three key parts are categorization, face representation and feature extraction. Although the Face representation restricts the detection and identification algorithms, it also characterizes the behavior of the Face input. After producing a new result for feature extraction from our LBPH histogram, the input that was classified found faces when compared to the recommended dataset. After that, we can ascertain whether our algorithm identified a known individual or not [8]. Face recognition is facing many

challenges with the advent of big data. One way to get around these problems is to use multifeatured extraction and concurrent processing. A useful pre-processing technique for reducing the dimension and raising the precision of facial image categorization (extraction) is feature selection.

To confirm that it is appropriate for a big data context, the comparison can be extended to a very large number of photos with the same outcome [9]. In this paper, a technique for implementing facial recognition on FPGA is provided. It is based on the CNN principle. The CNN realization process is split into two steps using this technique: network training on a PC and network implementation on an FPGA. On the FPGA, real-time facial recognition works well. This approach makes use of line portraits and image color and texture analysis. To differentiate between the face and mask in the image, pixel data from the Channel in the HSV color space is extracted and processed. Help CNN use the input image to learn more useful information. The analysis of the MAFA dataset demonstrates that our method is better than the other methods mentioned. This problem's solution is useful for understanding multi-angle problems as well. We can employ a face detection algorithm that can tell whether someone is wearing a mask or not in practical applications [10].

A hybrid face recognition method that predicts the need to extract features from five locations at first. Region 1 is calculated by the face segmentation procedure and used to extract the holistic features using the SURF (Speed Up Robust Features) feature extraction algorithm. The regional features are extracted using Regions 2, 3, 4, and 5. The MSER algorithm extracts the feature from the area surrounding the eyes in regions 3 and 4. The midpoint between the two eyes serves as an estimate for the nasal bridge, which is the focus of region 2. The area around the nose and mouth is known as region 5, and the MSER is used to extract the features from this area [11]. The selection of more effective methods for the tasks of feature extraction, classification, and facial picture detection was influenced by this research. The efficiency, computational complexity, and applicability of these techniques for deployment in a real-time setting were evaluated using image and video datasets [12].

Based on the amount of lighting present in the photos, the suggested method calculates the quantity of discrete cosine transform (DCT) coefficients at low frequencies for every image of an illuminated face. Face recognition is significantly more accurate with KELM after lighting normalization than it was with the earlier techniques. Owing to the approaches that operate in the DCT domain using DCT, inverse discrete cosine transforms (IDCT), and certain vector operations only when required [13], it makes it possible to detect face rotation in both pitch and yaw. After that, a single value that sums up the quality of the face is created by combining the measured parameters [14].

Frontal face recognition approaches with good performance in controlled environments include PCA and LDA-based methods. However, as with the majority of face recognition techniques, they perform much worse when there are changes in lighting, position, age, or occlusion. The retrieved feature vectors are then projected onto the newly formed global face subspace in order to provide the necessary features for carrying out a previous discrimination between the face classes [15]. Before uploading the desired chunk, a system first recognizes the face to pinpoint the facial region (ROI). Additionally, processing and feature extraction are only done on the

chosen area, which saves energy and allows the face recognition method to calculate results accurately. By removing the redundant image portions that are not required for feature vector extraction, the research has been able to show a 93% reduction in energy use. The Fog server's dynamic feature extraction and picture comparison techniques significantly cut down on energy use. By combining effective image processing methods with fog computing, the mobile-fog environment that has been implemented in this approach has opened up new possibilities for optimizing and addressing issues related to scarce energy resources [16].

The most crucial goal in video surveillance applications is facing identification in videos. due to the difficulties with image quality and the large amount of video information. Our approach beats various deep face recognition methods in terms of rates, similar to deep methods [17]. By putting out a brand-new, effective technique for calculating the sparse coding vector, the effectiveness of sparse coding coefficients is studied and enhanced. In order to identify the regularization parameters adaptively and increase stability, it is based on the introduction of a specific level of sparsity. The regularization and sparse coding terms together produced higher recognition rates than the original SRC approach, which only uses the local constraint term, according to the outcomes of the performance analyses [18]. With a greater identification rate across numerous open data sets, the fusion facial semantic feature (FFSF) efficiently extracts the features of the facial shape and semantic components. The robust performance accuracy and efficiency of the incremental learning mechanism (ILM) for classifying sample learning were confirmed after an independent assessment. In terms of accuracy and efficacy, the FFSF. The ILM approach outperformed other cutting-edge facial recognition algorithms [19].

The classifications can be broken down into three groups: hybrid techniques, feature-based techniques, and appearance-based techniques. Depending on the application domain and datasets used, each approach offers unique benefits and drawbacks. Most methods are ineffective for faces that are not frontal or that have a different distribution of poses [20].

3. Proposed method

Their findings proposed a new framework, the MRCN-MGWO, to increase the accuracy and efficiency of patient identification in healthcare settings. The MRCN-MGWO model use deep learning to improve diagnostic accuracy by analyzing facial features and employing specific medical photo preparation procedures.

To improve clarity, facial images are first denoised with a median filter (MF), followed by EIC. The MRCN creates robust feature vectors for detecting discrete facial patterns, whereas the Modified Grey Wolf Optimization (MGWO) method optimizes the weights and biases of a SAE. The MRCN-MGWO architecture is evaluated on the benchmark FEI dataset and shows potential for effective patient identification in a range of clinical contexts, exceeding current face recognition approaches.

Figure 1 shows how the MRCN-MGWO technique used the MF and EIC techniques to denoise and superiority-enhance face photos during the preprocessing step. The multi-stage regression and the capsule neural network are made up of three

elements: The feature extraction network extracts the fundamental features, the capsule network aggregates the features, and the probability vector for each stage of multi-stage regression is generated. We break down the feature extraction strategy in this organization into three stages. To further develop include processing, reinforce feature weights of critical information, and work on the capacity to extract facial qualities, each step is handled by a permanent attention block and a SE block. To guarantee that the impact of feature extraction is further developed layer by layer, there must be continuity between the phases. The feature aggregation network receives the feature maps developed in these three steps. Multi-stage regression will be used to combine the element maps from the three stages by providing the important probability vectors, expanding the accuracy of our forecast. Face recognition and classification are accomplished using the stacked autoencoder (SAE) model and the modified grey wolf optimization (MGWO) algorithm. The MGWO algorithm determines the weight and bias values of the SAE model.

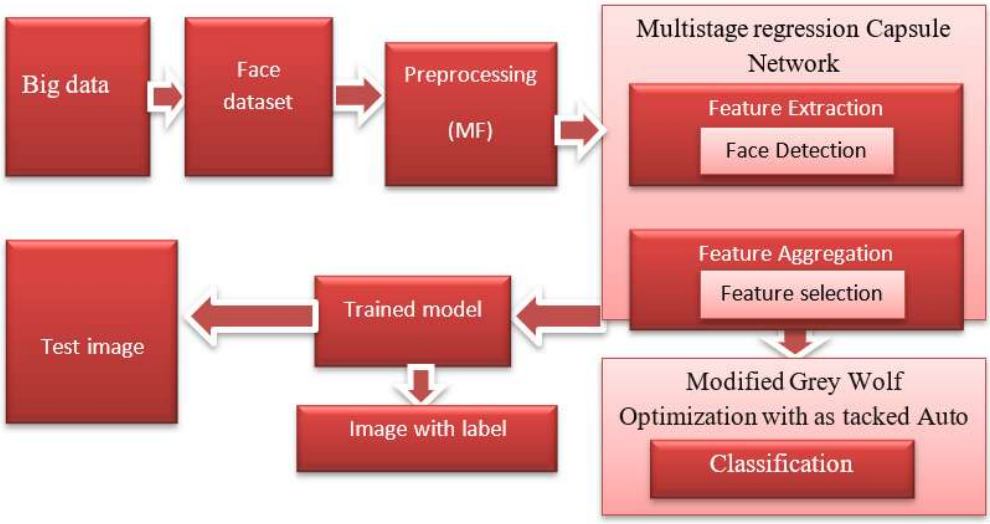

Figure 1. Architecture of proposed flow diagram

3.1. Face data set

This work uses the "CAS-PEAL" data sets for face recognition. The pure-pose, expression, light, accessory, background, duration, and distance variations are the seven prospects accessible in the CAS-PEAL face assortment. Starting around nine cameras are recording each subject at the same time; nine position (perspective) changes are utilized to consolidate all fluctuations consequently. 80% of the information are utilized for training, while just 20% are utilized for testing. As such, there are 69 test photos and 273 training images.

3.2. Pre-processing

It is possible to improve an image's accuracy and clarity during the preprocessing stage. The obtained face data set may be cleaned of noise using MF, and the brightness of the face photos has been improved using EIC. A nonlinear filtering called MF was utilized to reduce the distortion from the face data set in digital imaging. Since MF can maintain the advantage while minimizing distortion in specific situations, it is

commonly utilized. When using MF to window wise filter a picture, each element is replaced with the middle of the subsequent nearest ones. Certain noise edges (such random and salt-pepper noise) are preserved while others are completely eliminated by the nonlinear smoothness technique known as MF. Critical information is contained in the edge's shape for efficient recognition. Because it protects the edge design, the MF is crucial in preprocessing.

3.2.1. Enhancement of images based on contour (EIC)

EIC is a vital component for outlines. EIC can be used to determine the face's border. The face is not included when creating the digital image's shape. The binary image of the face section and the real image are then combined to generate the actual picture and the face portion. In terms of time and space, it might move. this strategy is especially helpful because it helps to boost contrast. The contrast of the image is defined as a parameter using the contrast augmentation index (CAI) as showed in Equation (1) and (2).

$$CAI = \frac{C_{processed}}{C_{actual}} \tag{1}$$

where $C_{processed}$ = value of the processed image's contrast and C_{actual} value of the actual image's contrast.

$$Image\ area\ contrast = \frac{m - s}{m + s} \tag{2}$$

where m is the image's "foreground" Gray-level value and s is the image's "background" Gray-level value. Finally, during the preprocessing stage, we use the MF and EIC approaches to obtain the denoised and superiority-enhanced face images.

3.3. Feature extraction

Three components comprise both the capsule neural network and the multi-stage regression: The capsule organization and the feature extraction network isolate the important highlights. aggregates the features, and the probability vector for each stage of multi-stage regression is generated. Our network for feature extraction consists of three branches. Each branch is composed of three basic permanent blocks, activation, weight normalization, convolution, a pooling layer, and SE blocks. There are likewise steady attentional restrictions in each step. The parts of the extremely durable consideration block's development are convolution, weight standardization; channel, spatial attention, and a fusion layer same like in **Figure 2**. A scope of fusion layer cores and down sampling techniques are utilized for the permanent durable unit. The feature maps with different kernel sizes measures are mixed by multiplying the components of the two feature maps created by channel consideration. The face recognition system is then created by adding the feature maps to the aggregate space.

3.3.1. Face detection

Face detection and facial key point detection have never been combined before, but MTCNN is an algorithmic technique that does just that. This method develops a face detection system by cascading three unique Deep Neural Networks (DNN). Utilizing such a bit-by-bit judgment, the beginning phases of face recognition handling can rapidly filter various non-facial region pieces of data. It can then create predictions

with even higher accuracy by increasing the number of calculations in the following network. For face detection, this approach also performs well in real-time. The concept behind MTCNN's face detection and recognition is similar to that of the V-J face detector. PNet, ONet, and RNet are the three DNNs that are a part of the cascade, respectively. These three DNNs all have consistent output and loss functions. In line with this, the MTCNN output has a three-part structure and uses the same loss function. The two-class loss function for each sample x^i.

$$L_i^{det} = -\left(y_i^{det}\log(p_i) + \left(1 - y_i^{det}\right)(1 - \log(P_i))\right) \tag{3}$$

The network's projected face probability is represented by Equation (3), and the initial image's assessment of the results of the face annotation is denoted by $y_i^{det} \in \{0,1\}$. Additionally, the face candidate frame-based regression loss function is written as showed in Equation (4).

$$L_i^{box} = \left\|\hat{y}_i^{box} - y_i^{box}\right\|_2^2 \tag{4}$$

Where, L_i^{box} refers to the network output, $\hat{y}_i^{box}$ is the location of the candidate prediction frame, which corresponds to the labelled face, and y_i^{box} is the face frame's true location. The aforementioned equation expressed as showed in Equation (5).

$$L_i^{landmark} = \left\|\hat{y}_i^{landmark} - y_i^{landmark}\right\|_2^2 \tag{5}$$

While $\hat{y}_i^{landmark}$indicates the exact location the $y_i^{landmark}$denotes the critical point's position in reference to the network output. Consequently, the entire loss function can be written as showed in Equation (6).

$$min\sum_{i=1}^{N}\sum_{j\in\{det,box,landmark\}}\alpha_j\beta_i^j L_i^j \tag{6}$$

The three loss functions' weighting coefficient's α are shown in Equation (6) as, while the β positive and negative distributions of the learning examples are shown as. α selected by different networks have chosen various different. Onet will select the larger $\alpha_{landmark}$, whereas Pnet and Rnet networks will select the smaller landmark. Three distinct network topologies are separately trained and then linked together in a sequence. Finally, it is possible to acquire the MTCNN network. In the meantime, it serves as the embedded system's face detection algorithm module.

Since each neural network has a varied receptive field size, creating images of various sizes necessitates using the image pyramid processing technique. More faces can be found thanks to the creation of additional photo pyramids in light of the regular face detection. The picture size is decreased and the lower furthest reaches of the identifying pixels for the face size is put to 80 together to lessen the input size of the picture.

Permanent attention block: The following changes are made to the permanent form of attention unit called an attention block, in order to promote the extraction of facial features. See Equation (7).

$$F: X \rightarrow \tilde{X}, X \in R^{H \times W \times C},\ \tilde{X} \in R^{H/ \times W/ \times C/} \tag{7}$$

The *F*(.) convolution operation represents a standard convolution applied across both spatial and channel dimensions. To perform the mapping, multi-scale kernel characteristics are applied to the channels. Pooling operations are then used to generate distinct feature vectors, and channel-wise multiplication is employed to combine the results. The same scaling factors used to determine the channel size are also applied to determine the spatial dimensions.

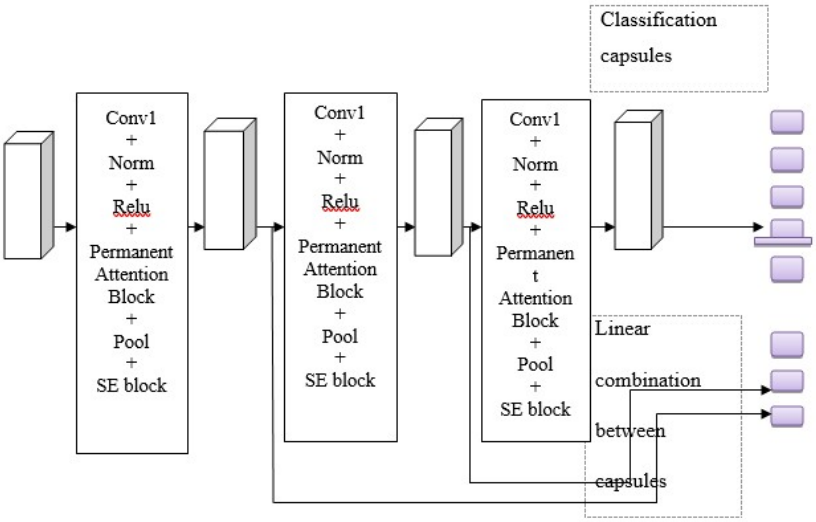

Figure 2. The feature extraction algorithm's flow.

Block squeeze-and-excitation: The SE block is a specific sort of consideration block built on an element diagram channel. The essential objective of a SE block is to improve results by developing the ability to change feature weights in response to loss, increasing the weights of useful feature maps. The effectiveness of the network can be increased by SE blocks at a low cost to computation.

$$F_{tr}: X \rightarrow \cup,\ X \in R^{H' \times W' \times C'}, U \in R^{H \times W \times C} \tag{8}$$

In this Equation (8), the extracted feature is U, and the input graph is X. To determine the link between channels, compress the element u and total the feature chart to deliver a diagram of aspects WH that is utilized as the feature descriptor. The feature chart, some of the time called a descriptor, is then upgraded with the spatial information of each feature graph's global receptive field, or F_{sq} (.). Utilizing the descriptor feature map, the organization layer can then be educated about the global responsive field.SE blocks use the $F_{sq}(.)$ Utilizing the correlation between channels can be resolved by using a squeeze operation to establish correlations between the channels.

$$Z_c = F_{sq}(u_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i,j) \tag{9}$$

In Equation (9), the u_c two-dimensional matrix with channel C in U where the subscript c stands for the channel. The dependence on channel dimensions is then completely captured using the aggregate data from the compression operation. In order to accomplish this, we use the following:

$$F_{ex(.\ ,w)=}\ \sigma\big(g(z,W)\big) = \sigma\big(W_2.\sigma.(W_1.z)\big) \tag{10}$$

In Equation (10) W_1, W_2 are the two fully connected layers, and σ is the Relu function. The sigmoid function comes after the second fully connected layer. After these operations are finished, the feature map weights are obtained and combined with the original view features.

$$F_{scale}(u_c, s_c) = u_c s_c \tag{11}$$

In Equation (11) the feature map $u_c \in R^{H\times W}$ scaling index is multiplied by, and $F_{scale}(u_c, s_c)$ represents s_c. The feature map data from each channel is combined by the two complete connection layers. The improved feature map is then produced by multiplying S' by the input feature map.

3.3.2. Feature aggregation

The CNN's convolutional structure makes it the best approach for determining whether features exist. When CNN is used to examine the relationship between feature attributes, The input image obscures the feature detector's exact target information. Because to things like rotation and similar events, CNN misidentifies the item. A CapsNet-based solution is described, taking into account that the human face typically has a broad rotation angle, in order to mitigate the limitations of the CNN methodology. The feature-fusion-oriented RS-CapsNet architecture serves as the foundation for the CapsNet used in this study. We have therefore given our feature aggregation module the name CapsNet. We also employ a 1 1 convolution layer to decrease the number of channels in order to minimize computation and encapsulation. We convert each of the realized feature maps into a capsule and use the linear relationship between the capsules to combine features.

We convert all of the realized feature maps into capsules in order to maximize the depiction of features, and then we integrate features by taking use of the linear link between the capsules. Utilizing an assortment of capsule types for various nearby feature maps and a dynamic routing procedure, we produce capsules that might reasonably represent most of the items. The final product is the categorized capsule, which is the combination of the original feature-mapped capsule. The construction of capsule networks is made possible by each local feature map, and each capsule is D (2).

Feature selection: Partition the feature map created by the input picture's final convolution into smaller local feature maps before constructing the "intermediate capsule." The most popular goods can be represented by this capsule. The classified capsule is created by combining the first capsule generated by feature mapping with the intermediate capsule. Regarding the topic "how to slice the feature map," are better preserved when using the "vertical and horizontal sliding window" technique. Using the smallest local feature maps is what we want to do.

Linear combination between capsules: Through a linear link inside each capsule and the reconstruction of the feature map into capsules, each of which ought to represent an object that can be identified in the source image. This fixes the problem of duplicate data in the backdrop of the input image. Using the linear combination method that was previously discussed, the capsules are flattened, their direction is made constant, and their length is maintained within the range [0, 1].

3.3.3 Multistage regression

For instance, DEX trained a CNN-based network to classify the faces into various categories. The procedure of their approach can be described as showed in Equation (12).

$$\tilde{y} = \vec{p}.\vec{l} = \sum_{i=0}^{R} p_i.l_i \tag{12}$$

Where $\vec{l}$ denotes indices of each probability and $\vec{p} = (P_0, P_1, \ldots . P_R)$ isbased on the input facial data and obtained from the model, showing the distribution of face probability. Separating face recognition based on a one-year interval is accurate and straightforward. The network must be trained; however, it is difficult and time-consuming due to the network's vast number of parameters and the number of computational resources needed. by making the deep neural network's size more manageable and effective for multistage regression. With the aid of multistage regression, the face is predicted.

$$\tilde{y} = \vec{p}.\vec{l} = \sum_{s=1}^{s}\sum_{i=0}^{s_i-1} p_i^s.l_i^s = \sum_{s=1}^{s}\sum_{i=0}^{s_i-1} p_i^s.i\left(\frac{R}{\sum_{p=0}^{s-1} s_p}\right) \tag{13}$$

In this Equation (13), the indicators $s, p_i^s.l_i^s$ for each stage denote, respectively, the class indexes within each stage. A categorization of facial length into wholly different classes does not sufficiently address issues connected to ageing, according to observations based on earlier research. Indicator selection and stage width scaling strategies have been used to overcome the issue. The following is an explanation of how stage width adjustment works. See Equation (14).

$$\overline{S_l} = s_i(1 + \theta_i) \tag{14}$$

where $\overline{S_l}$ and s_i denote the i-th stage width before and after the adjustment, respectively; θ_i is a factor that affects the network and controls how much the stage width has changed. As a result, the stage's width is showed in Equation (15).

$$w_s = \frac{R}{\sum_{p=0}^{s-1} \overline{s_p}} \tag{15}$$

The offset vector $\vec{\beta}^s$ has the same size as $l_i^s$is for moving the class indices inside each stage. Our model's outputs additionally include the value of $\vec{\beta}^s$ is showed in Equation (16).

$$\vec{\beta}^s = (\beta_0^s, \beta_1^s, \ldots, \beta_{s_i-1}^s) \tag{16}$$

Then, the index now becomes in Equation (17).

$$\vec{l} = i + \beta_i^s \tag{17}$$

Both operations rely on input facial data, which enables our model to develop the ability to predict the apparent face in real time.

3.4. Image classification

In the end, the feature vectors are accepted as input by the SAE model, which correctly classes the facial images. The initial data is changed into a coded result, which is then expanded by the network's later layers into a finished output. You can learn more about autoencoders by using a "denoising" autoencoder. By mixing the original and noisy input, the denoising autoencoder improves the output. Encoding and decoding are both components of the taught AE method. The network's successive layers subsequently extend the coded result, which was created from the encoded portion that was used to create the initial input, into a finished output. You can learn more about autoencoders by using a "denoising" autoencoder. By mixing the original and noisy input, the denoising autoencoder improves the output. For categorization, image processing, and other deep learning applications, autoencoders are helpful. Encoding and decoding are also parts of the taught AE method. The encoded part was utilized to map the info information to stowed away portrayals, while the decoded part made it conceivable to imitate the input data in the hidden portrayals. In the wake of mapping the info information to hidden away portrayals, the decoded portion was displayed as information data being imitated in the hidden representations. It is basic to give the info dataset without labels as $\{x_n\}_{n-1}^N$, where $x_n \in R^{m\times 1}$, h_n represents the computed hidden encoded vector inx_n and $\hat{x}_n$ denotes the final state's decoded vector. Given this, the following is the encoding procedure. See Equation (18).

$$h_n = f(W_1 x_n + b_1) \tag{18}$$

where f stands for the encoded functions, $W_1$is the weighted encoding matrix, and b_1 stands for the bias vectors. The decoded process yields the following results. See Equation (19).

$$\hat{x}_n = g(W_2 h_n + b_2) \tag{19}$$

where g stands for the decoded functions, W_2showed the decoding weighted matrix, and b_2 signifies the bias vectors. The optimization to reduce the reconstruction error served as the AE parameter set. See Equation (20).

$$\emptyset(\Theta) = \arg \min_{\theta, \theta'} \frac{1}{n} \sum_{i=1}^{n} L\left(x^i, \hat{x}^i\right) \tag{20}$$

where L is a loss function, $L\left(x^i, \hat{x}^i\right) = \|x - \hat{x}\|^2$. An unsupervised statewise learning strategy is used to stack nAEs to n concealed states, and then a supervised approach is used to refine the architecture of SAE. The SAE-based approach is therefore divided into three phases: The BP approach was utilised to minimise the cost function for each hidden state after training, and fine-tuning was made possible.

- Use input data to train the primary AE and produce learned feature vectors.

- Until the training is complete, the next stage's input was the feature vector from the prior state.
- Following the training of each hidden state, the BP technique was used to upgrade the weight that has been trained and labelled order to achieve fine-tuning.

The majority of optimization algorithms go through two similar stages: Exploration and exploitation. Investigating the search space is the process of exploration. An algorithm's initial iterations examine the search space in the pursuit of better answers. Through this method, searching agents can scan the whole search space without hitting local optima. The stage of an algorithm known as exploitation is the stage where exploitation grows and exploration gradually decreases until the algorithm converges to an optimal solution. For the algorithm to work properly, these two stages must be balanced properly. Therefore, suggesting a novel strategy is preferred.

These two steps in MGWO are managed by the parameter ~a. This value is linearly decreased, as was indicated in the preceding section. Early iterations of the algorithm place more of an emphasis on exploration, while later iterations focus more on exploitation Develop a balanced strategy between these two periods by altering the linear behavior of the strength of exploration. Our new control parameter is as showed in Equation (21).

$$\vec{a}(t) = 2 - 2\left(\frac{t}{t_{max}}\right)^{k} \tag{21}$$

where the constant k is present, t denotes the current iteration, and t_{max} represents the maximum number of iterations. Here, the parameter $\vec{a}$ is still decreasing nonlinearly from 2 to 0. The emphasis will be on exploitation for k values between 0 and 1, but the quality of searching capacity may degrade. Before the algorithm moves on to exploitation, all possible values larger than 1 will be considered in the search space. Trial and error should be used to get a suitable value for *k*. Though there is certainly room for improvement, GWO's performance should increase we map a local search around the optimal response to compensate is fully utilized. If the grey wolf achieves a greater degree of fitness after applying this approach to the best grey wolf position, it will be moved to the new site. The following factors determine the new position is showed in Equation (22).

$$\vec{X}_n = \vec{X}_a + r(U - L)(z - 0.5) \tag{22}$$

where z is the mapping parameter that is changed after each iteration, r is the centre, and $U$and $L_{are}$are the upper and lower boundaries as showed in Equation (23).

$$\vec{z}_{t+1} = 4 \times \vec{Z}_t \times \left(1 - \vec{Z}_t\right) \tag{23}$$

The following pseudo-code represents the modified GWO. MGWO enhances convergence by incorporating adaptive control parameters and improved leader-guided search mechanisms, enabling faster exploitation of promising regions while maintaining balanced exploration compared to standard GWO is showed in Algorithm 1.

Algorithm 1 MGWO

Random generation of gray wolvesX_i (i=1,2,….,n)

Setting initial values of a ⃗,A ⃗, and C ⃗

t_max= total number of iterations

Fitness evaluation of each gray wolf

X ⃗_α: the best solution

X ⃗_β: second best solution

X ⃗_γ: third best solution

while t<t_maxdo

for each X ⃗_iin X ⃗do

Update the position of X ⃗_i

end for

Update a ⃗,(A,) ⃗and C ⃗

Evaluate the fitness of all gray wolves

Map a new position for X ⃗_αand move it there if it's improved

Update X ⃗_α,X ⃗_β, and X ⃗_γ

t=t+1

end while

return X ⃗_αas the solution

3.4.1. Parameter tuning using the MGWO algorithm

The SAE model is subjected to the MGWO algorithm in order to best adapt its parameters (such as weight and bias). A revolutionary SI-optimized approach is now known as MGWO. In actuality, it can be a novel technique that, by default, quickens the social and hunting hierarchy of GW. In order to develop GW's social performance, it can be divided into 4 states, including α, β, δ and ω. α took into consideration the best solutions carried out by $\beta$and δ respectively, and the permanent solution derived in ω: The three main fittest wolves, α, β, and δ who are close to the prey, provide assistance in recognising the meal in the difficult area. Wolves increase the position of or during the surrounding stride, βor δ demonstrated in Equation (24).

$$\vec{D} = \left|\vec{C}.\vec{X}_{P(t)} - \vec{X}(t)\right| \tag{24}$$

Where t refers to the current iteration, $\vec{X}_{p(t)}$ indicates the location of the prey at the moment, and $\vec{X}_{(t)}$ indicates the location of the wolves at the moment. The mathematical procedure produces coefficient vectors $\vec{A}$and $\vec{C}$, while $\vec{D}$ denotes the separation between wolves and their prey.

$$\vec{A} = 2\vec{a}\vec{r}_1 - \vec{a} \tag{25}$$

$$\vec{C} = 2\vec{r}_2 \tag{26}$$

In Equations (25) and (26), during all repetitions, the $\vec{a}$ elements have linearly decreased from two to zero,$\vec{r}_1$ and $\vec{r}_2$denoting the two vectors formed in steps of zero

and one, respectively. The locations closer to the location of the prey are now defined as $\alpha, \beta,$ and δ. The top 3 were the best options in the hunting scenario, ω permanent wolves are suitable for replacing the core elements of the first three best wolves. The wolf's den received an upgrade on the basis of Equation (27)–(33).

$$\vec{D}_{\alpha} = \left|C_1.\vec{X} - \vec{X}\right| \tag{27}$$

$$\vec{D}_{\beta} = \left|C_2.\vec{X} - \vec{X}\right| \tag{28}$$

$$\vec{D}_{\delta} = \left|C_3.\vec{X} - \vec{X}\right| \tag{29}$$

$$\vec{X}_1 = \vec{X}_{\alpha} - \vec{A}_2.\left(\vec{D}_a\right) \tag{30}$$

$$\vec{X}_2 = \vec{X}_{\beta} - \vec{A}_2.\left(\vec{D}_{\beta}\right) \tag{31}$$

$$\vec{X}_3 = \vec{X} - \vec{A}\,.\left(\vec{D}_{\delta}\right) \tag{32}$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{33}$$

Where $\vec{X}_{\alpha}$ denotes the location of α; $\vec{X}_{\beta}$denotes the location of β; $\overrightarrow{X_{\delta}}$ for the location of $\vec{X}$ denotes the location of previous solutions; and $\vec{C}_1, \vec{C}_2$and$\vec{C}_3$ denote the vectors produced in an arbitrary manner. Here, the arbitrary vectors $\vec{A}_1, \vec{A}_2$and$\vec{A}_3$ are shown, and t stands for the number of rounds. The step size of the wolves was used, $\alpha, \beta,$ and δ the accompanying equations in (24) to (26). The locations of the ω wolves that arise are then assessed using the fundamentals of Equations (27)–(33).

A Feedback Framework (FF) is derived by the MGWO technique to improve classification performance. It specifies a positive number to indicate the prospective solution's ideal efficiency. Equation (34) gives the study's minimal classification error rate, which is assumed to be FF.

$$\begin{aligned} fitness(x_i) = {} & Classifier\ Error\ Rate(x_i) \\ = {} & \frac{number\ of\ misclassified\ instances}{Total\ number\ of\ instances} * 10 \end{aligned} \tag{34}$$

4. Performance measures

Yes, there are positive and bad categories for FR. The face data sets were divided into FOUR trials by combining trainee type and genuine type. (i.e., "true positive," "true negative," "false positive," and "false negative"). Some of the specific metrics for the planned work for FR that are evaluated in this section include accuracy, precision, recall, and *F*-score. The following is an illustration of one of these measures.

4.1. Accuracy

With the necessary face data, accuracy offers recognition as showed in Equation (35).

$$A = \frac{(tp + tn)}{(tp + tn + fp + fn)} \tag{35}$$

where the numbers for true positive, false positive, and accurate predictions for both positive and negative examples (*fp*), separately, are equivalent to the quantity of accurate predictions made for a positive sample (*tp*), accurate predictions made for a negative sample (*tn*), and vice versa.

4.2. Precision (B)

It is the proportion of beneficial retrieved instances as showed in Equation (36).

$$B = \frac{(tp)}{(tp + fp)} \tag{36}$$

4.3. Recall (C)

It calculates the proportion of recovered relevant photographs as showed in Equation (37).

$$C = \frac{(tp)}{(tp + fn)} \tag{37}$$

4.4. F-Score(D)

A statistic for determining how exactly a set of face data can be recognized is the F-score as showed in Equation (38).

$$D = \frac{tp}{tp + 0.5(fp + fn)} \tag{38}$$

4.5. Expressivity score (E)

Emotion strength, positive expressivity, and negative expressivity make up the expressivity score.

4.6. Recognition score (F)

The recognition score for each phase of the cycle depends on the extent of good decisions pursued out of the multitude of decisions made. The complete number of decisions is equivalent to the result of the positive and negative possibilities.

5. Result and discussion

To recognize faces, one uses the "CAS-PEAL" data sets. Seven distinct faces are included in the CAS-PEAL face collection. Since nine cameras are simultaneously recording each subject, all variations are automatically combined using nine position (viewpoint) modifications. The CASPEAL face datasets containing the variations are shown in **Table 1.**

Table 1. Face data sets are described.

View points	**Nine**						
	Facing direction	Expressions	Accessories	Lighting	Duration	Background	Distance
Variants	3	6	6	15	2	4	2
Mixed	27	54	54	135	18	36	18
Full amount	342						

Only 20% of the data are used for testing, compared to 80% for training. Consequently, there are 273 training photos and 69 test images. The suggested work is evaluated using a number of measures, including accuracy, precision, recall, and *F*-score.

The components of expressivity score include negative expressivity, positive expressivity, and emotion strength. The expressivity ratings of both suggested and current approaches are displayed in **Table 2** and **Figure 3**.

Table 2. Scores for expressivity of suggested and used techniques.

Parameters	**Ensemble-aided FR**	**Coupled mapping**	**Deep-DA**	**MRCN-MGWO**
Happiness	55	65	44	80
Anger	43	77	35	86
Sadness	67	76	75	95
Surprise	56	36	65	78
Disgust	58	67	70	82
Fear	60	53	66	88

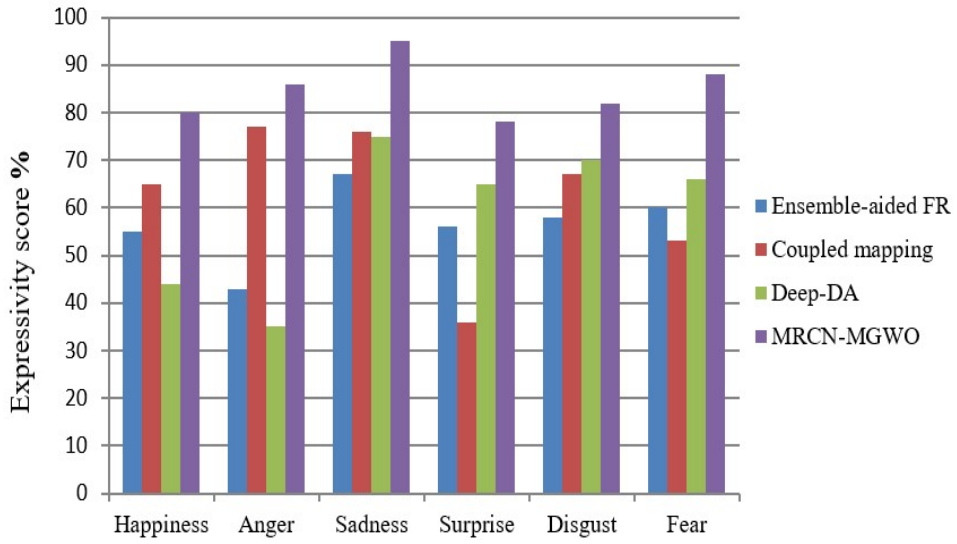

Figure 3. Scores for suggested and existing methodologies' expressivity (%).

The percentage of favorable decisions made out of all feasible options makes up the recognition score. The recognition rating of both proposed and existing approaches is displayed in **Table 3** and **Figure 4**. Score of planned and active recognition

strategies (%). **Table 4** Analysis of the MRCN-MGWO technique's accuracy, precision, recall, and F-Score in comparison to existing methods.

Table 3. Rating of proposed and existing recognition techniques (%).

Parameters	Ensemble-aided FR	Coupled mapping	Deep-DA	MRCN-MGWO
Happiness	62	20	44	77
Anger	66	70	32	93
Sadness	25	54	82	90
Surprise	72	60	46	70
Disgust	45	55	64	80
Fear	20	35	44	88

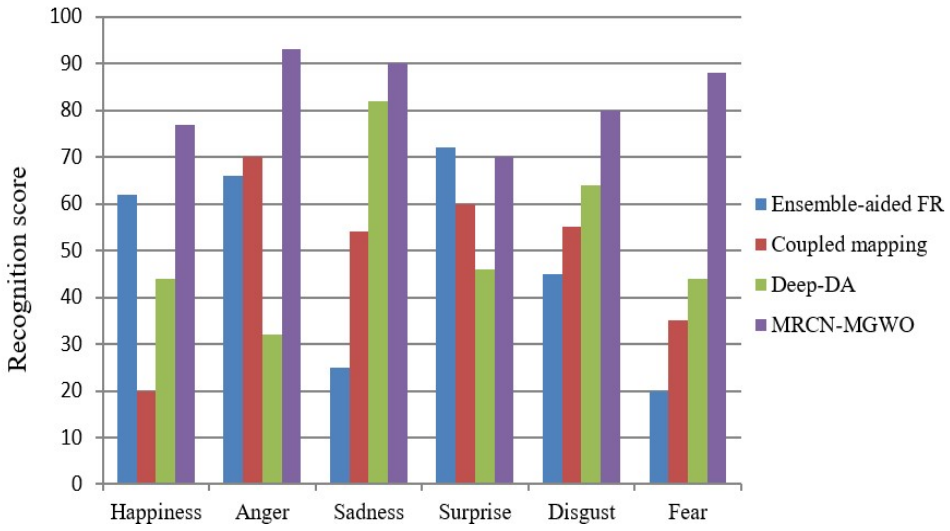

Figure 4. Recognition rating (%) for both current and suggested techniques.

Table 4. Compares the accuracy, precision, recall, and *F*-score of the individual face collections when using proposed and existing techniques.

Techniques	Accuracy (%)	Precision (%)	Recall (%)	*F*-Score (%)
Enabled-aided FR	94.28	95.28	90.28	94.22
Coupled mapping	96.09	89.09	86.09	86.09
Deep-DA	97.34	92.34	87.34	87.34
MRCN-MGWO	98.67	96.13	94.21	93.11

Figure 5 graph's y-axis displays accuracy, precision, recall, and F-score using expected accuracy rates for the provided data sets, while the x-axis displays the face data sets. Lastly, we used our proposed technique to perform the highest accurate face recognition. Our research of performance indicators revealed that the suggested strategy surpassed the current face recognition algorithms in terms of accuracy (98.67%), precision (96.13%), recall (94.21%), and *F*-score (93.11%).

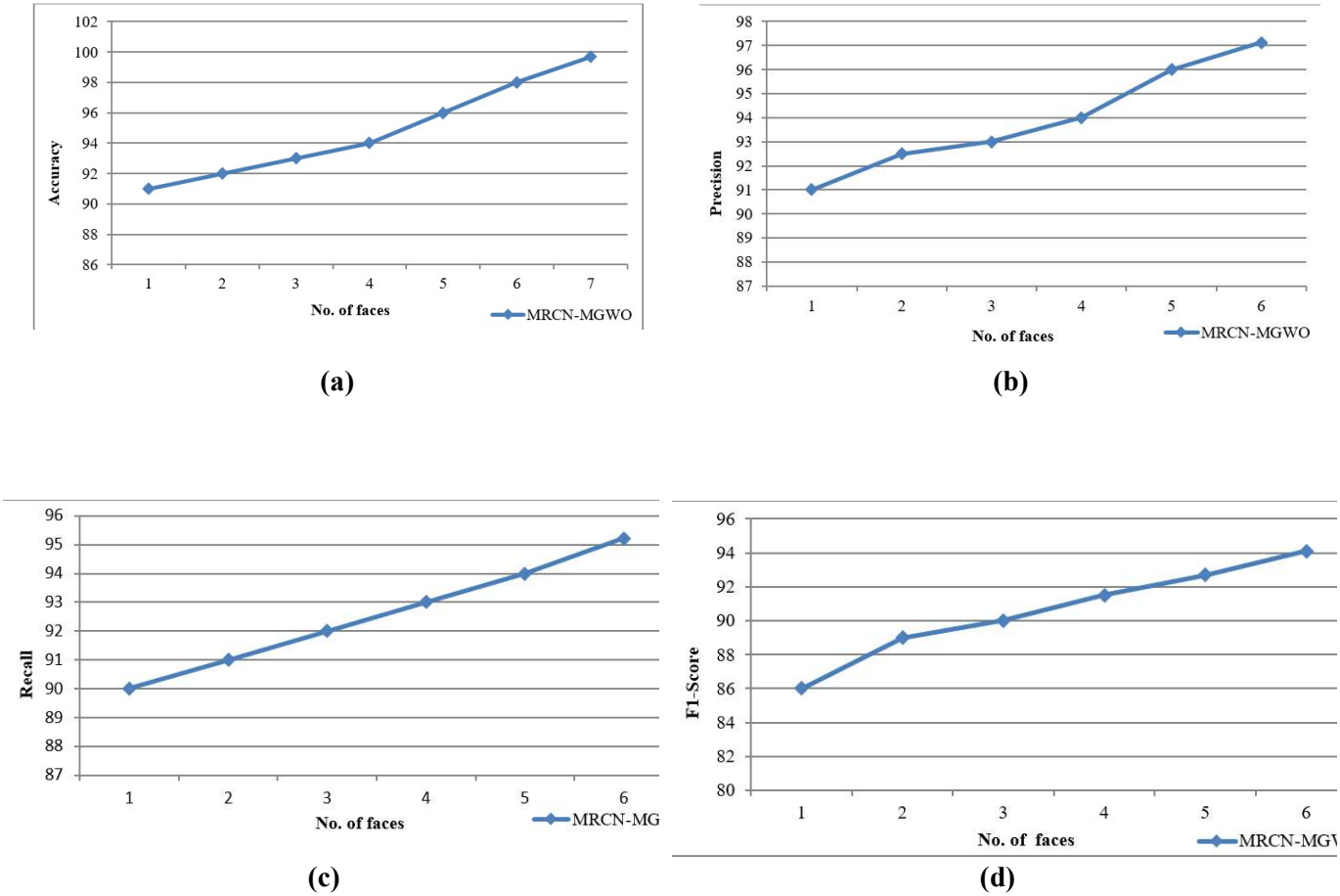

Figure 5. Performance of suggested MRCN-MGWO in terms of **(a)** accuracy; **(b)** precision; **(c)** recall; **(d)** *F*-score.

6. Conclusion

The proposed MRCN-MGWO framework demonstrates significant advancements in patient identification using face recognition technology in clinical settings. By integrating deep learning with specialized preprocessing techniques, the model effectively enhances facial feature analysis for improved diagnostic accuracy. The optimization of the stacked autoencoder using MGWO further refines feature extraction, leading to superior performance compared to existing face recognition techniques. Experimental evaluation on the FEI dataset confirms the model's effectiveness, achieving high accuracy (98.67%), precision (96.13%), recall (95.21%), and F-score (93.11%). These results highlight the potential of MRCN-MGWO for reliable and efficient biometric identification in healthcare applications. In the future, hybrid DL models and hyperparameter optimizers might be employed to improve recognition performance.

Author contributions: Conceptualization, AJJJSR and TSS; methodology, AJJJSR; software, TSS; validation, AJJJSR and TSS; formal analysis, TSS; investigation, TSS; resources, AJJJSR; data curation, TSS; writing—original draft preparation, AJJJSR; writing—review and editing, AJJJSR; visualization, TSS; supervision, TSS; project administration, AJJJSR; funding acquisition, AJJJSR. All authors have read and agreed to the published version of the manuscript.

Funding: None.

Ethical approval: Not applicable.

Informed consent statement: Not applicable.

Acknowledgments: The author would like to express their heartfelt gratitude to the supervisor for his guidance and unwavering support during this research.

Conflict of interest: The authors declare no conflict of interest.

References

1. Zhu Y, Jiang Y. Optimization of face recognition algorithm based on deep learning multi feature fusion driven by big data. Image and Vision Computing. 2020; 104: 104023. doi: 10.1016/j.imavis.2020.104023
2. Wang D, Yu H, Wang D, et al. Face recognition system based on CNN. 2020 International conference on computer information and big data applications (CIBDA). IEEE. 2020. pp. 470–473. doi: 10.1109/CIBDA50819.2020.00111
3. Meng Q, Zhao S, Huang Z, et al. Magface: A universal representation for face recognition and quality assessment. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021. pp. 14225–14234.
4. Ni H. Face recognition based on deep learning under the background of big data. Informatica. 2020; 44(4): 491–495. doi: 10.31449/inf.v44i4.3390.
5. Qu X, Wei T, Peng C, et al. A fast face recognition system based on deep learning. 2018 11th international symposium on computational intelligence and design (ISCID). IEEE. 2018; 1: 289–292. doi: 10.1109/ISCID.2018.00072
6. Neethirajan S. The role of sensors, big data and machine learning in modern animal farming. Sensing and Bio-sensing Research. 2020; 29: 100367. doi: 10.1016/j.sbsr.2020.100367
7. Zheng T, Deng W. Cross-pose lfw: A database for studying cross-pose face recognition in unconstrained environments. Beijing University of Posts and Telecommunications, Tech. Rep. 2018; 5(7): 5.
8. Ahmed A, Guo J, Ali F, et al. LBPH based improved face recognition at low resolution. 2018 international conference on Artificial Intelligence and big data (ICAIBD). IEEE. 2018. pp. 144–147. doi: 10.1109/icaibd.2018.8396183
9. Dalali S, Suresh L. Face Recognition: Multi-features extraction with parallel computation for big data. 2018 3rd International Conference on computational systems and information technology for sustainable solutions (CSITSS). IEEE. 2018. pp. 133–144. doi: 10.1109/csitss.2018.8768746
10. Li S, Ning X, Yu L, et al. Multi-angle head pose classification when wearing the mask for face recognition under the COVID-19 coronavirus epidemic. 2020 International conference on high performance big data and intelligent systems (HPBD&IS). IEEE. 2020. pp. 1–5. doi: 10.1109/HPBDIS49115.2020.9130585
11. Shoba VBT, Sam IS. A hybrid features extraction on face for efficient face recognition. Multimedia Tools and Applications. 2020; 79(31): 22595–22616. doi: 10.1007/s11042-020-08997-1
12. Sanchez-Moreno AS, Olivares-Mercado J, Hernandez-Suarez A, et al. Efficient face recognition system for operating in unconstrained environments. Journal of Imaging. 2021; 7(9): 161. doi: 10.3390/jimaging7090161
13. Vishwakarma VP, Dalal S. A novel approach for compensation of light variation effects with KELM classification for efficient face recognition. Advances in VLSI, communication, and signal processing: select proceedings of VCAS 2018. Singapore: Springer Singapore. 2019. pp. 1003–1012. doi: 10.1007/978-981-32-9775-3_89
14. Bahroun S, Abed R, Zagrouba E. KS-FQA: Keyframe selection based on face quality assessment for efficient face recognition in video. IET Image Processing. 2021; 15(1): 77–90. doi: 10.1049/ipr2.12008
15. Mukherjee D, Das R, Majumdar S, et al. Energy efficient face recognition in mobile-fog environment. Procedia Computer Science. 2019; 152: 274–281. doi: 10.1016/j.procs.2019.05.016
16. Benouareth A. An efficient face recognition approach combining likelihood-based sufficient dimension reduction and LDA. Multimedia Tools and Applications. 2021; 80(1): 1457–1486. doi: 10.1007/s11042-020-09527-9
17. Abed R, Bahroun S, Zagrouba E. KeyFrame extraction based on face quality measurement and convolutional neural network for efficient face recognition in videos. Multimedia Tools and Applications. 2021; 80(15): 23157–23179. doi: 10.1007/s11042-020-09385-5

18. Alajmi M, Awedat K, Essa A, et al. Efficient face recognition using regularized adaptive non-local sparse coding. IEEE Access. 2019; 7: 10653–10662. doi: 10.1109/ACCESS.2019.2890845
19. Zhong R, Wu H, Chen Z, et al. Fusion facial semantic feature and incremental learning mechanism for efficient face recognition. Soft Computing. 2021; 25(14): 9347–9363. doi: 10.1007/s00500-021-05915-x
20. Anwarul S, Dahiya S. A comprehensive review on face recognition methods and factors affecting facial recognition accuracy. Proceedings of ICRIC 2019: Recent Innovations in Computing. 2019; 597: 495–514. doi: 10.1007/978-3-030-29407-6_36