*Computer Systems Science & Engineering*
DOI: 10.32604/csse.2023.036377
***Article***

Tech Science Press

# Earthworm Optimization with Improved SqueezeNet Enabled Facial Expression Recognition Model

**N. Sharmili[1], Saud Yonbawi[2], Sultan Alahmari[3], E. Laxmi Lydia[4], Mohamad Khairi Ishak[5], Hend Khalid Alkahtani[6,*], Ayman Aljarbouh[7] and Samih M. Mostafa[8]**

[1]Computer Science and Engineering Department, Gayatri Vidya Parishad College of Engineering for Women, Visakhapatnam, Andhra Pradesh, India
[2]Department of Software Engineering, College of Computer Science and Engineering, University of Jeddah, Jeddah, Saudi Arabia
[3]King Abdul Aziz City for Science and Technology, Riyadh, Kingdom of Saudi Arabia
[4]Department of Computer Science and Engineering, Vignan's Institute of Information Technology, Visakhapatnam, 530049, India
[5]School of Electrical and Electronic Engineering, Engineering Campus, Universiti Sains Malaysia (USM), Nibong Tebal, Penang, 14300, Malaysia
[6]Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh, 11564, Saudi Arabia
[7]Department of Computer Science, University of Central Asia, Naryn, 722600, Kyrgyzstan
[8]Faculty of Computers and Information, South Valley University, Qena, 83523, Egypt
*Corresponding Author: Hend Khalid Alkahtani. Email: Hkalqahtani@pnu.edu.sa
Received: 28 September 2022; Accepted: 08 December 2022

**Abstract:** Facial expression recognition (FER) remains a hot research area among computer vision researchers and still becomes a challenge because of high intra-class variations. Conventional techniques for this problem depend on hand-crafted features, namely, LBP, SIFT, and HOG, along with that a classifier trained on a database of videos or images. Many execute perform well on image datasets captured in a controlled condition; however not perform well in the more challenging dataset, which has partial faces and image variation. Recently, many studies presented an endwise structure for facial expression recognition by utilizing DL methods. Therefore, this study develops an earthworm optimization with an improved SqueezeNet-based FER (EWOISN-FER) model. The presented EWOISN-FER model primarily applies the contrast-limited adaptive histogram equalization (CLAHE) technique as a pre-processing step. In addition, the improved SqueezeNet model is exploited to derive an optimal set of feature vectors, and the hyperparameter tuning process is performed by the stochastic gradient boosting (SGB) model. Finally, EWO with sparse autoencoder (SAE) is employed for the FER process, and the EWO algorithm appropriately chooses the SAE parameters. A wide-ranging experimental analysis is carried out to examine the performance of the proposed model. The experimental outcomes indicate the supremacy of the presented EWOISN-FER technique.

**Keywords:** Facial expression recognition; deep learning; computer vision; earthworm optimization; hyperparameter optimization

## 1 Introduction

Facial expression recognition (FER) plays a pivotal role in the artificial intelligence (AI) period. Following the emotional information of humans, machines could offer personalized services. Several applications, namely customer satisfaction, virtual reality, personalized recommendations, and many more, rely upon an effective and dependable way of recognizing facial expressions [1]. This topic has grabbed the attention of several research scholars for years. Still, it becomes a challenging topic as expression features change significantly with environments, head poses, and discrepancies in the different individuals involved [2]. Technologies for transmission have conventionally been formulated based on the senses that serve as the main part of human communication. Specifically, AI voice recognition technology utilizing AI speakers and sense of hearing was commercialized due to the enhancements in AI technology [3]. By using this technology that identifies voice and language, there are AI robots that could communicate thoroughly with real life for handling the daily lists of individuals. But sensory acceptance was needed to communicate more accurately [4]. Thus, the most essential technology was a vision sensor, since vision becomes a large part of human perception in many communications.

AI robots communicate between a machine and a human, human faces present significant data as a hint to understand the user's present state. Thus, the domain of FER has been studied broadly for the past 10 years [5]. Currently, with the rise of appropriate data and continuous progression of deep learning (DL), a FER mechanism that precisely identifies facial expressions in several surroundings is being studied actively [6]. FER depends on an evolutionary and ergonomic technique. Depending on physiological and evolutionary properties, universality, similarity, and emotions in FER research are categorized into 6 categories: anger, happiness, disgust, sadness, surprise, and fear [7]. Moreover, emotions are further categorized into 7 categories, along with neutral emotion. Motivated by the achievement of Convolutional Neural Networks (CNNs), numerous existing DL techniques were devised for facial expression recognition and were superior to conventional techniques. CNN-related techniques could automatically learn and model the extracted features of the facial object due to the neural network structure; therefore, such methods could have superior outcomes in real-time applications [8]. In recent times, by using DL and particularly CNNs, numerous features have been learned and extracted for a decent FER mechanism [9]. It is noted that, in facial expressions, several clues arise from some parts of the face, for example, the eyes and mouth; other parts, like the hair and ears, play little parts in the result [10]. This indicates that the machine learning (ML) structure preferably must concentrate only on significant parts of the face and be less delicate than other facial areas.

This study develops an earthworm optimization with an improved SqueezeNet-based FER (EWOISN-FER) model. The presented EWOISN-FER model primarily applies contrast limited adaptive histogram equalization (CLAHE) technique as a pre-processing step. In addition, the improved SqueezeNet model is exploited to derive an optimal set of feature vectors, and the stochastic gradient boosting (SGB) model performs the hyperparameter tuning process. Finally, EWO with sparse autoencoder (SAE) is employed for the FER process, and the EWO algorithm appropriately chooses the SAE parameters. A wide-ranging experimental analysis is carried out to demonstrate the enhanced performance of the EWOISN-FER technique.

The remaining sections of the paper are organized as follows. Section 2 provides the literature review, and Section 3 elaborates on the proposed model. Then, Section 4 offers performance validation, and Section 5 concludes.

## 2 Literature Review

Nezami et al. [11] introduced a DL algorithm for improving engagement recognition from images that overwhelm the challenges of data sparsity via pre-training on an easily accessible facial expression dataset and beforehand training on a specialized engagement dataset. Firstly, FER can be trained to give a rich face representation using DL. Next, the model weight is employed for initializing the DL-based methodology for recognizing engagement; we term this the engagement module. Bargshady et al. [12] report on a newly improved DNN architecture intended to efficiently diagnose pain intensity in 4-level thresholds with facial expression images. To examine the robustness of presented techniques, the UNBC-McMaster Shoulder Pain Archive Database contains facial images of humans and has been first balanced after being utilized for testing and training the classifier technique, in addition, to optimally tuning the VGG-Face pre-trainer as a feature extracting tool. The pre-screened attributes, utilized as method inputs, were transmitted to generate a novel enhanced joint hybrid CNN-BiLSTM (EJH-CNN-BiLSTM) DL technique that encompasses CNN, which can be linked to the joint Bi-LSTM for multi-classifying the pain.

The authors in [13] concentrate on a semi-supervised deep belief network (DBN) method for predicting facial expressions. To achieve precise facial expression classification, a gravitational search algorithm (GSA) was applied to optimise certain variables in the DBN network. The HOG features derived from the lip patch offer optimal performance for precise facial expression classification. In [14], a new deep model can be presented to enhance facial expressions' classifier accuracy. The presented method includes the following merits: initially, a pose-guided face alignment technique was presented to minimize the intra-class modification surpassing environmental noise's effect. A hybrid feature representation technique has been presented to gain high-level discriminatory facial features that attain superior outcomes in classifier networks. A lightweight fusion backbone was devised that integrates the ResNet and the VGG-16 to obtain low-data and low-calculation training.

Minaee et al. [15] modelled a DL technique related to attentional convolutional networks to focus on significant face areas and attain important enhancement over prior methods on many datasets. Kim et al. [16] present a novel FER mechanism related to the hierarchical DL technique. The feature derived from the appearance feature-oriented network can be merged with the geometric feature in a hierarchical framework. The presented technique integrates the outcome of the softmax function of 2 features by considering the mistake linked with the second highest emotion (Top-2) predictive outcome. Further, the author presents a method for generating facial imageries with neutral emotion utilizing the autoencoder (AE) method. By this method, the author could derive the dynamic facial features among the emotional and neutral images without sequence data.

## 3 The Proposed Model

In this study, a new EWOISN-FER algorithm was devised to recognise and classify facial emotions. The presented EWOISN-FER model employed the CLAHE technique as a pre-processing step. In addition, an improved SqueezeNet method is exploited to derive an optimal set of feature vectors, and the SGB model performs the hyperparameter tuning process. Lastly, EWO with SAE is employed for the FER process and the EWO algorithm appropriately chooses the SAE parameters. Fig. 1 demonstrates the block diagram of the EWOISN-FER algorithm.

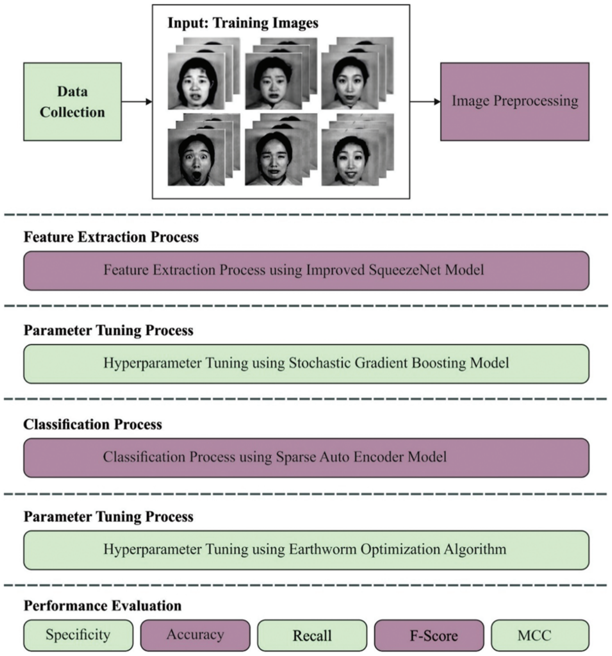

Data Collection
Input: Training Images
Image Preprocessing
Feature Extraction Process
Feature Extraction Process using Improved SqueezeNet Model
Parameter Tuning Process
Hyperparameter Tuning using Stochastic Gradient Boosting Model
Classification Process
Classification Process using Sparse Auto Encoder Model
Parameter Tuning Process
Hyperparameter Tuning using Earthworm Optimization Algorithm
Performance Evaluation
Specificity
Accuracy
Recall
F-Score
MCC

**Figure 1:** Block diagram of EWOISN-FER approach

### *3.1 Image Pre-Processing*

CLAHE splits the image into MxN local tiles. For all the tiles, histograms can be individually calculated [17]. First, we should compute the average amount of pixels for each region to computer the histogram as follows:

$$N_A = \frac{N_X \times N_Y}{N_G} \tag{1}$$

Here, $N_a$ indicates the average amount of pixels, $N_x$ and $N_Y$ denotes the pixel count in the X and $Y$ dimensions, *and* $N_G$ denotes the number of gray levels. Next, determine the clip limit to clip the histogram.

$$N_{CL} = N_A \times N_{NCL} \tag{2}$$

N$_{CL}$ shows the clip limit, and $N_{NCL}$ indicates the normalized clip limits within [0, 1]. Then, the clip limit can be exploited to the histogram height for all the tiles.

$$H_i = \begin{cases} N_{CL} & if\ N_i \geq N_{CL} \\ N_i & else \end{cases} \quad i = 1,\ 2,\ \ \ldots,\ \ 1 - 1 \tag{3}$$

where $H_i$ indicates the height of the histogram of the $i - th$ tile, $N_i$ denotes the histogram of *the* $i - th$ tile, and $L$ represents the number of gray levels.

The overall amount of clipped pixels is calculated by the following equation.

$$N_c = (N_X \times N_Y) - \sum\nolimits_{i=0}^{L-1} H_i \tag{4}$$

Let, $N_C$ be the number of clipped pixels. Afterwards, computing $N_C$, we should reallocate the clipped pixel. The pixel is either redistributed uniformly or non-uniformly. The following equation is used to calculate the number of pixels to be redistributed.

$$N_R = \frac{N_C}{L} \tag{5}$$

Now, $N_R$ indicates the number of pixels to be redistributed. Then, the clipped histogram can be normalized as follows.

$$H_i = \begin{cases} N_{CL} & if N_i + N_R \geq N_{CL} \\ N_i + N_R & else \end{cases} \quad i = 1,\ 2,\ \ \ldots,\ \ 1 - 1 \tag{6}$$

The amount of undistributed pixels is calculated using Eqs. (4) and (5). Eq. (6) is repeated until each pixel is redistributed. Lastly, the cumulative histogram of the context region is formulated as follows.

$$C_i = \frac{1}{(N_X \times N_Y)} \sum\nolimits_{j=0}^{i} H_j \tag{7}$$

After each calculation is accomplished, the histogram of the contextual region is matched with Rayleigh, exponential or uniform likelihood distribution. For the output image, the tile is fused and the removal of the artefacts among the independent tiles is completed using the bilinear interpolation, the novel value of $s$ that is represented by $s$' as follows.

$$s' = (1 - y)((1 - X) \times R_1(s) + X \times R_2(s)) + y((1 - X) \times R_3(s) + X \times R_4(s))) \tag{8}$$

Then, the enhanced image is attained.

### *3.2 Feature Extraction Using Improved SqueezeNet*

In this stage, the improved SqueezeNet method is exploited to derive an optimal set of feature vectors, and the hyperparameter tuning process can be performed by the SGB model. The CNN initiates with a standalone convolutional layer (conv1) and slowly increases the filter number for all the fire modules from the start to the end of the networks. The CNN architecture implements max pool with a stride of 2 afterwards layers conv1, fire4, fire8, and conv10. Since the light weighted SqueezeNet architecture trained on the insect databases with 9 object classes, we adapted the conv10 layers (because they are personalized to other tasks) that comes from the pre-trained SqueezeNet CNN architecture and replaces newly adapted conv10 layers with a nine-class output [18]. In this study, the Fire module contains 1 × 1 filter $(s_{1\times1})$ in the squeeze layer, 1 × 1 filters $(e_{1\times1})$ and 3 × 3 filters $(e_{3\times3})$ in expand layer 3D hyperparameter, hence the module with eight Fire modules that contain twenty-four dimension hyper-parameters. Then, describe B indicates the amount of expanding filters in the initial Fire module; afterwards M Fire model, we increased the expanded filters using C. Hence, for $i - th$ modules, the amount of expanding filter is $e_i = e_{i1\times1} + e_{i3\times3} = B + \left(C \times \left|\frac{i}{M}\right|\right)$. In other words, $e_{i3\times3} = e_i \times per$ signifies the percentage of expanding filter that is 3 × 3. Lastly, determine SR that signifies the filter count in the squeeze layer of Fire modules: $s_{i1\times1} = SR \times e_i$. The enhanced module has the: B = 128, C = 128, per = 0.4, M = 2, and SR = 0.2.

Furthermore, a bypass connection is added to improve the representation bottleneck presented by squeeze layers, however, the limitation is that the quantity of input channels of the Fire module and amount of output channels is dissimilar, and the component-wise addition operator could not be satisfied. Hence, determine a complicated bypass as a bypass that involves a 1 × 1 convolutional layer with the number of filters equal to the number of output channels. In such a way, we require the module for learning a residual function among inputs and outputs, in the meantime, attaining the upper convolutional layer with rich semantic data. Fig. 2 illustrates the architecture of SqueezeNet. The experiment result shows the best object detection performance compared to the pre-trained, light-weighted SqueezeNet

CNN module. The final softmax layers frequently handle multi-classification problems. In softmax regression, the probability $p$ that an input ×belonging to class $c$ is formulated below

$$p(y_i = c|x_i;\ \ 0) = \frac{e^{\theta_n^c} x_i}{\sum_{m=1}^{C} e^{\theta_m^c} x_i} \tag{9}$$

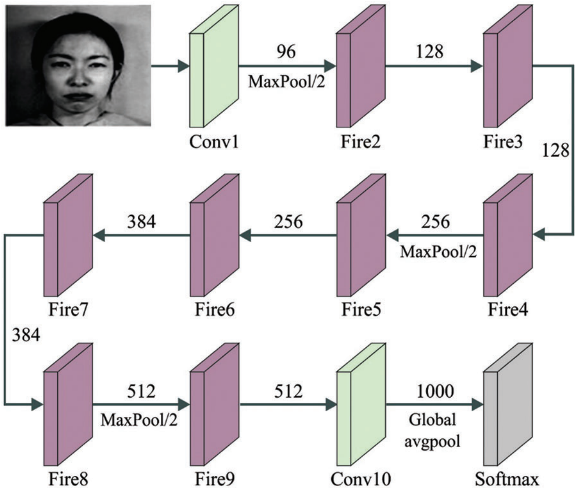


**Figure 2:** Structure of SqueezeNet

In Eq. (9), $\theta$ indicates the parameter of the DL algorithm. The loss function represents the cross-entropy loss between the actual class and the output probability vectors:

$$j(\theta) = -\frac{1}{N}\left[\sum_{i-1}^{N}\sum_{i-c}^{c} 1\{y_i = c\}\log^{p}\right] \tag{10}$$

In Eq. (10), 1 signifies an indicative function whose value is 1 as long as the $i^{th}$ image of the test is accurate.

Friedman [19] built an SGB model by presenting the concept of gradient descent (GD) into the boosting algorithm. Gradient boosting is an ensemble learning mechanism fused with the decision and boosting trees, and the novel paradigm is constructed alongside the GD direction of the loss function of the predetermined method. The core of the SGB approach is to minimalize the loss function among the classification and real functions by training the classifier function $*(X)$.

The loss function distribution is the core element of the application of the SGB algorithm, and it has pertinence towards each loss function. For the K-class problem, the substitute loss function (multiclass log-loss) is recommended by Friedman and is extensively employed in different fields, and it is expressed in the following equation.

$$\psi\left(y_k,\ \ F_k\left(X\right)_1^K\right) = -\sum_{k=1}^{K} y_k \log p_k(X) = -\sum_{k=1}^{K} y_k \log\left[\exp\left(F_k(X)\right)/\sum_{l=1}^{K} \exp\left(F_l(X)\right)\right] \tag{11}$$

Now $X = \{x_1,\ \ x_2,\ \ \ldots x_n\}$ indicates the input parameter, k shows the number of classes, y indicates the output parameter, *and* $p_k(X)$ shows the probability [20]. Next, the subsequent formula is attained using Eq. (12):

$$\tilde{y}_{im} = -\left[\frac{\partial\psi\left(y_i,\ F_j(x_i)\right)_{j=1}^{K}}{\partial F(x_i)}\right]_{\left(F_j(X)=F_{j,m-1}(X)\right)_1^K} = y_i^k - p_k(x_i) \tag{12}$$

In Eq. (12), $y_i^k - p_k(x_i)$ indicates the existing residuals and thereby, induces the $K$-trees. In the meantime, it produces each $K$ tree with an $L$-terminal node at iteration $m$, $R_{klm}$. Next, all the tree terminal nodes are resolved using a different line search.

$$\gamma_{lm} = \arg\min_{\gamma} \sum\nolimits_{ix\in R_{lm}} \psi(y_i,\ F_{m-1}(x_i) + \gamma) \tag{13}$$

Every function is upgraded and later established as the SGB.

### *3.3 FER Process Using Optimal SAE*

At the final stage, the SAE model is employed for FER. During this case, it can execute the hyperbolic tangent activation function to encode and decode [21]. Primarily, the bias vector and weight matrix were allocated arbitrary values. For obtaining a network with generalized ability, the number of trained instances is at least 10 times the count of degrees of the freedoms. The recreated errors decreased significantly with training for determining suitable values of weight and bias. Utilizing the sparse AE, the cost function signifies the reconstructed errors demonstrated in Eq. (14).

$$E = \frac{1}{N}\sum\nolimits_{n=1}^{N}\sum\nolimits_{k=1}^{K}(\mathcal{X}_{nk} - \mathcal{X}_{nk}^{'})^2 + \lambda\cdot\Omega_w + \beta\cdot\Omega_s, \tag{14}$$

whereas $E$ signifies the cost function of AE; $N$ denotes the number of trained instances, $K$ refers to the dimensional of data; $\mathcal{X}_{nk}$ and $\mathcal{X}_{nk}^{'}$ demonstrate the input and output values correspondingly of k-dimensional from the $n^{th}$ sample; λ and β define the coefficients; $\Omega_w$ signifies the $L_2$ regularization, and $\Omega_s$ implies the sparse regularization.

To improve AE's generation capability and avoid overfitting, the $L_2$ regularized, and the sparse regularized were executed. The $L_2$ regularized prevent the improving value of weighted illustrated in Eq. (15).

$$\Omega_w = \frac{1}{2}\sum\nolimits_{l}^{c}\sum\nolimits_{j}^{N}\sum\nolimits_{i}^{K}[W_{ji}^{(l)}]^2, \tag{15}$$

In which C indicates the count of hidden layers and $W_{ji}^{(l)}$ defines the weighted of $i^{th}$ dimensional of $j^{th}$ instance from the $l^{th}$ layer. The sparse regularization was defined in Eqs. (16) and (17).

$$\hat{\rho}_i = \frac{1}{N}\sum\nolimits_{j=1}^{N}\mathcal{H}_i(\mathcal{X}_j) \tag{16}$$

$$\Omega_s = \sum\nolimits_{i=1}^{D} KL(\rho||\hat{\rho}_i) = \sum\nolimits_{i=1}^{D}\ \ plog\left(\frac{\rho}{\widehat{\hat{\rho}}_i}\right) + (1-\rho)\ \log\left(\frac{1-\rho}{1-\widehat{\hat{\rho}}_i}\right), \tag{17}$$

whereas $\hat{\rho}_i$ signifies the average value of hidden units $i$; $N$ implies the count of trained instances, and $\rho$ stands for the value which is set to AE. The Kullback-Leibler divergence $KL$ measures the errors among the desired values $\hat{\rho}_i$ and $\rho$, and $D$ represents the hidden nodes count.

The EWO algorithm is exploited in this study to adjust the SAE parameters. The EWO algorithm was exploited that is simulated the reproductive process of earthworms (EW) to overcome the optimization problems [22]. The EW is a kind of hermaphrodite and performs at all of them and employs female and male sex organs. Consequently, the single parent EW produces a child EW via themselves:

$$u_{i1,k} = u_{max,k} + u_{min,k} - \alpha u_{i,k} \tag{18}$$

The abovementioned equation described the process of producing *the* $k^{th}$ component of a child's EW $i1$ in parent EW $i$. $u_{i1,k}$ and $u_{i,k}$ indicates the $k^{th}$ component of EW $i1$ and $i$. $u_{\max,k}$ and $u_{\min,k}$ shows the effective restriction of $k^{th}$ component of all the EW. $\alpha$ indicates the similarity factor that ranges within [0, 1] and defines the movement in parent-to-child EW.

The Reproduction_2 applies an enhanced kind of crossover operator. Consider M as the number of child EWs, which is 2 or 3 in major components. The amount of parent EWs (N) is an integer that is greater than 1. In the study, the uniform crossover was employed with $N = 2$ and $M = 1$. In 2 parent EWs $P_1$ and $P_2$ are selected for employing the roulette wheel chosen as follows:

$$P = \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} \tag{19}$$

Firstly, 2 offspring $u_{12}$ and $u_{22}$ are made in 2 parents. The arbitrary number rand within [0, 1] is complete and *the* $k^{th}$ component of $u_{12}$ and $u_{22}$ are made in the following:

If $rand > 0.5$,

$$u_{12,k} = P_{1,k} \tag{20}$$

$$u_{22,k} = P_{2,k} \tag{21}$$

Then,

$$u_{12,k} = P_{2,k} \tag{22}$$

$$u_{22,k} = P_{1,k} \tag{23}$$

Finally, the produced EW $u_{i2}$ in Reproduction-2 is illustrated as (24). Let $rand1$ be another arbitrarily produced value within [0, 1].

$$u_{i2} = \begin{cases} u_{12}\ for\ rand\ 1 < 0.5 \\ u_{22}\ else \end{cases} \tag{24}$$

Then, the producing EWs $u_{i1}$ and $u_{i2}$, the EW $u_i^{'}$ for the following generation as follows:

$$u_i^{'} = \beta u_{i1} + (1 - \beta)u_{i2} \tag{25}$$

In Eq. (25), $\beta$ denoted as the "proportional factor". It is used for manipulating the proportion of $u_{i1}$ and $u_{i2}$ that global and local searching effectiveness was recollected from balancing as follows:

$$\beta^{t+1} = \gamma\beta^t \tag{26}$$

Now, t means the present generation. Initially, at $t = 0$, $\beta = 1$. $\gamma$ denotes the variable viz., outcomes to cooling factor. The solution needs that existing run-away in local optimal. Therefore, the "Cauchy Mutation" (CM) is implemented. It enhanced the searchability of "EWO".

$$W_k = \left(\sum_{i=1}^{N_{pop}} u_{i,k}\right) / N_{pop} \tag{27}$$

In Eq. (27), $W_k$ shows the weighted vector for a $k^{th}$ component of population $i$ and $N_{pop}$ indicate the population size:

$$u_i^{''} = u_i^{'} + W_k \times Cd \tag{28}$$

Now, $Cd$ indicates the arbitrary value drawn in the "Cauchy distribution" regarding $= 1$. Where $\tau$ signifies the "scale parameter".

The EWO technique mostly defines a fitness value to accomplish maximal classification outcomes. It calculates a positive integer for demonstrating enhanced outcomes on the candidate solution. In the study, decreasing the classification error rate could be processed as the fitness function, as follows. The optimal holds the least error rate, and the poorly attained solution provides a higher error rate.

$$fitness(x_i) == \frac{No.\ of\ misclassified\ samples}{Total\ No.\ of\ samples} \times 100 \tag{29}$$

## 4 Experimental Validation

The FER performance of the EWOISN-FER model is tested using two datasets namely CK+ [23] and RaFD [24] datasets. The CK+ dataset holds 630 samples under seven classes, and the RaFD dataset comprises 3377 samples under seven classes, as depicted in Table 1. A few sample images are displayed in Fig. 3. Fig. 4 indicates the confusion matrices of the EWOISN-FER model on CK+ dataset. The confusion matrices demonstrated that the EWOISN-FER model can effectually recognize seven distinct types of facial expressions.

**Table 1:** Dataset details

| Labels | Class | No. of samples | |
|---|---|---|---|
| | | CK+ | RaFD |
| C-1 | Anger | 63 | 494 |
| C-2 | Disgust | 91 | 494 |
| C-3 | Fear | 40 | 466 |
| C-4 | Happiness | 130 | 493 |
| C-5 | Sadness | 42 | 485 |
| C-6 | Surprise | 150 | 452 |
| C-7 | Neutral | 114 | 493 |
| **Total number of samples** | | **630** | **3377** |

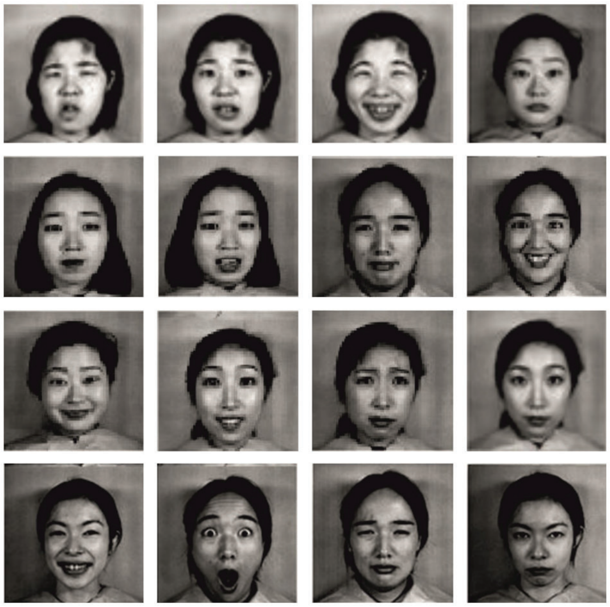
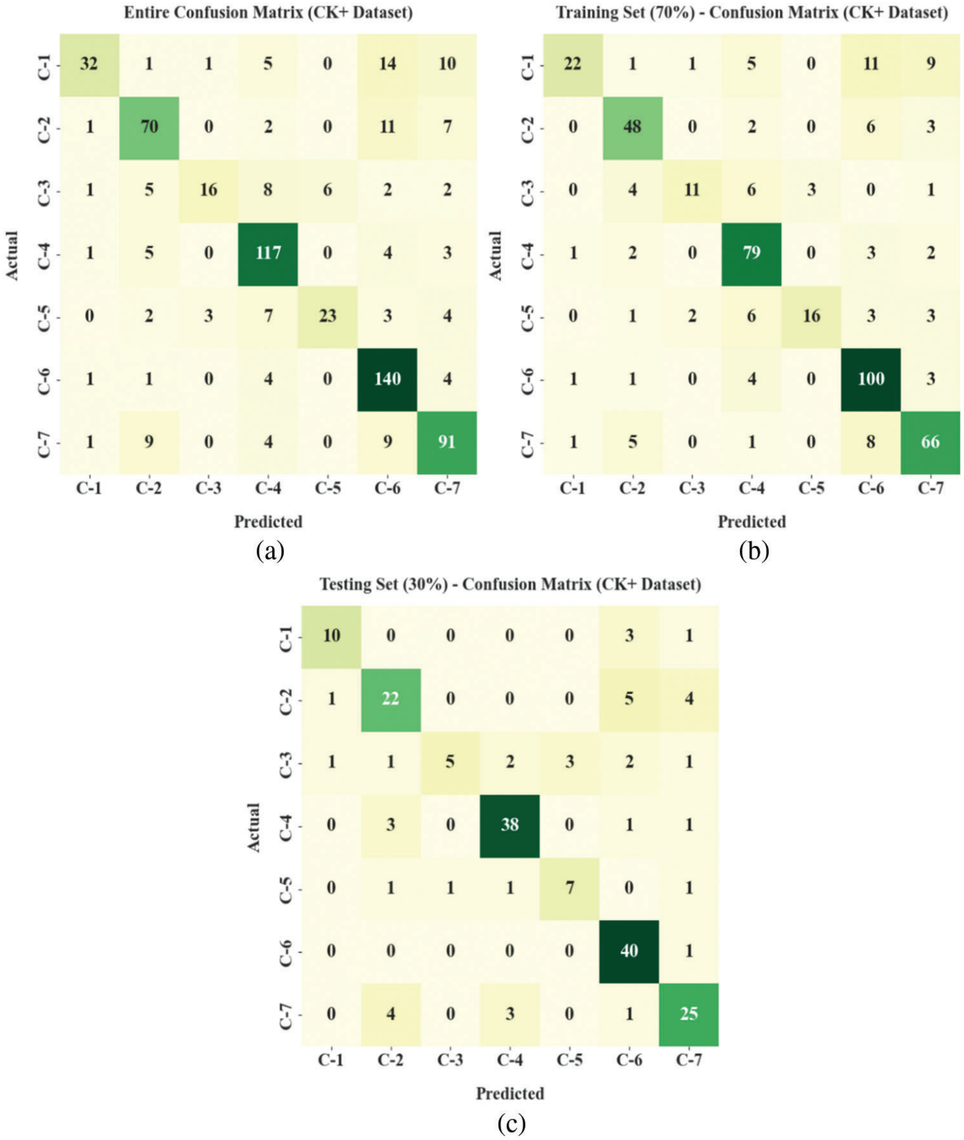


**Figure 3:** Sample images



(a) (b)

(c)

**Figure 4:** Confusion matrices of EWOISN-FER approach under CK+ dataset (a) Entire dataset, (b) 70% of TR data, and (c) 30% of TS data

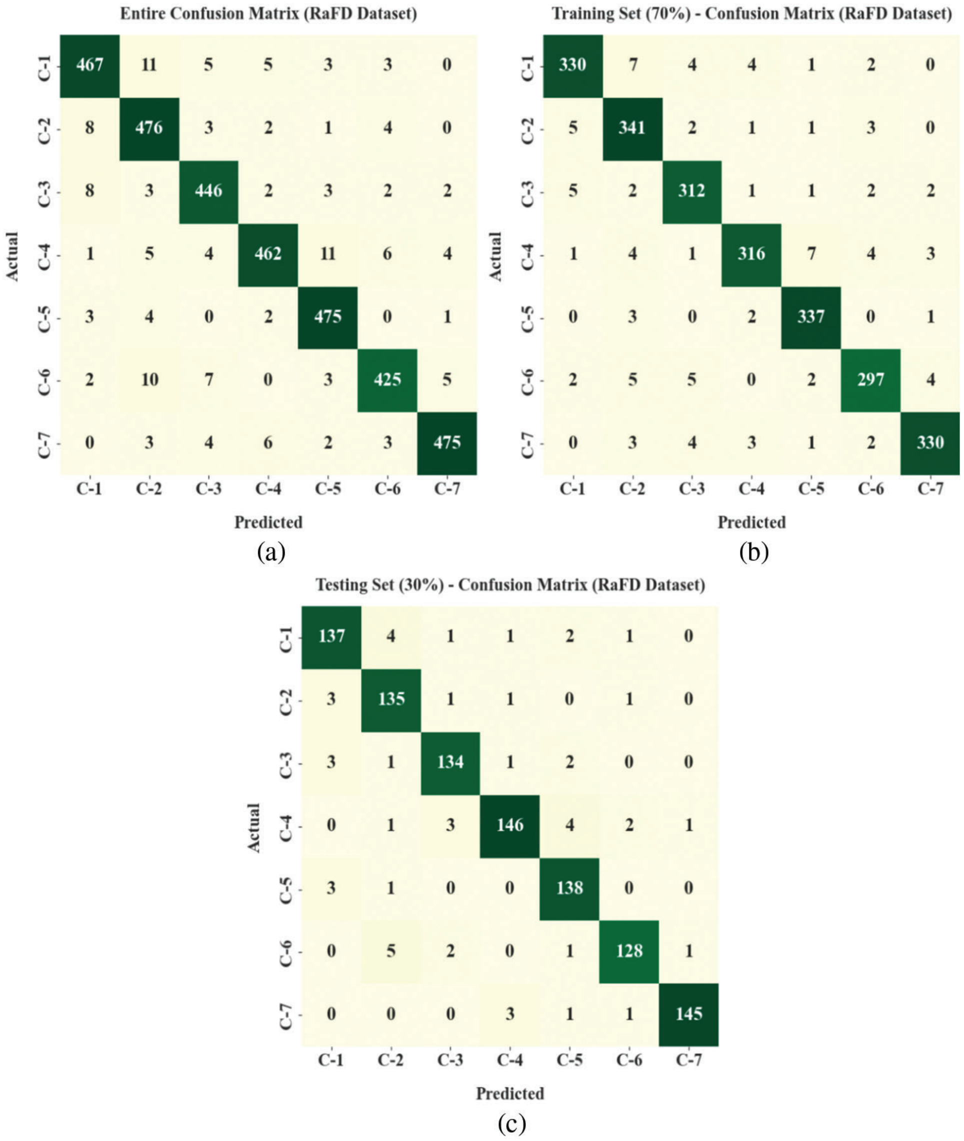
Table 2 shows brief FER outcomes of the EWOISN-FER method on the CK+ dataset. The results inferred the effectual FER performance of the EWOISN-FER model over other methods. For example, on the entire dataset, the EWOISN-FER model has offered $accu_y$, $reca_l$, $spec_y$, $F_{score}$, and MCC of 93.61%, 69.38%, 96.05%, 72.03%, and 69.46%, respectively. In Parallel, on 70% of TR data, the EWOISN-FER approach has presented $accu_y$, $reca_l$, $spec_y$, $F_{score}$, and MCC of 93.59%, 69.41%, 96.02%, 72.04%, and 69.66%, correspondingly. Eventually, on 30% of TS data, the EWOISN-FER technique has provided $accu_y$, $reca_l$, $spec_y$, $F_{score}$, and MCC of 93.65%, 71.26%, 96.13%, 72.72%, and 69.99% correspondingly.

**Table 2:** Result analysis of EWOISN-FER approach with distinct class labels under the CK+ dataset

| CK+ Dataset | | | | | |
|---|---|---|---|---|---|
| Labels | Accuracy | Recall | Specificity | F-score | MCC |
| **Entire dataset** | | | | | |
| C-1 | 94.29 | 50.79 | 99.12 | 64.00 | 63.68 |
| C-2 | 93.02 | 76.92 | 95.73 | 76.09 | 72.00 |
| C-3 | 95.56 | 40.00 | 99.32 | 53.33 | 54.69 |
| C-4 | 93.17 | 90.00 | 94.00 | 84.48 | 80.37 |
| C-5 | 96.03 | 54.76 | 98.98 | 64.79 | 63.97 |
| C-6 | 91.59 | 93.33 | 91.04 | 84.08 | 79.16 |
| C-7 | 91.59 | 79.82 | 94.19 | 77.45 | 72.33 |
| **Average** | **93.61** | **69.38** | **96.05** | **72.03** | **69.46** |
| **Training phase (70%)** | | | | | |
| C-1 | 93.20 | 44.90 | 99.23 | 59.46 | 59.98 |
| C-2 | 94.33 | 81.36 | 96.34 | 79.34 | 76.09 |
| C-3 | 96.15 | 44.00 | 99.28 | 56.41 | 57.08 |
| C-4 | 92.74 | 90.80 | 93.22 | 83.16 | 79.03 |
| C-5 | 95.92 | 51.61 | 99.27 | 64.00 | 64.06 |
| C-6 | 90.93 | 91.74 | 90.66 | 83.33 | 77.79 |
| C-7 | 91.84 | 81.48 | 94.17 | 78.57 | 73.61 |
| **Average** | **93.59** | **69.41** | **96.02** | **72.04** | **69.66** |
| **Testing phase (30%)** | | | | | |
| C-1 | 96.83 | 71.43 | 98.86 | 76.92 | 75.49 |
| C-2 | 89.95 | 68.75 | 94.27 | 69.84 | 63.82 |
| C-3 | 94.18 | 33.33 | 99.43 | 47.62 | 50.51 |
| C-4 | 94.18 | 88.37 | 95.89 | 87.36 | 83.59 |
| C-5 | 96.30 | 63.64 | 98.31 | 66.67 | 64.79 |
| C-6 | 93.12 | 97.56 | 91.89 | 86.02 | 82.56 |
| C-7 | 91.01 | 75.76 | 94.23 | 74.63 | 69.17 |
| **Average** | **93.65** | **71.26** | **96.13** | **72.72** | **69.99** |

Fig. 5 exhibits the confusion matrices of the EWOISN-FER algorithm on the RaFD dataset. The confusion matrices illustrated that the EWOISN-FER technique could effectually recognise seven different types of facial expressions.

Table 3 displays the detailed FER outcomes of the EWOISN-FER technique on the RaFD dataset. The outcomes signify the effectual FER performance of the EWOISN-FER method over other models. For example, on the entire dataset, the EWOISN-FER approach has presented $accu_y$, $reca_l$, $spec_y$, $F_{score}$, and MCC of 98.72%, 95.52%, 99.25%, 95.52%, and 94.79% correspondingly. Simultaneously, on 70% of TR data, the EWOISN-FER approach has presented $accu_y$, $reca_l$, $spec_y$, $F_{score}$, and MCC of 98.79%, 95.75%, 99.29%, 95.76%, and 95.06% correspondingly. Finally, on 30% of TS data, the EWOISN-FER method has shown *accu*$_y$, *reca*$_l$, *spec*$_y$, $F_{score}$, and MCC of 98.56%, 94.98%, 99.16%, 94.96%, and 94.14% correspondingly.


Entire Confusion Matrix (RaFD Dataset)

| Actual \ Predicted | C-1 | C-2 | C-3 | C-4 | C-5 | C-6 | C-7 |
|---|---|---|---|---|---|---|---|
| C-1 | 467 | 11 | 5 | 5 | 3 | 3 | 0 |
| C-2 | 8 | 476 | 3 | 2 | 1 | 4 | 0 |
| C-3 | 8 | 3 | 446 | 2 | 3 | 2 | 2 |
| C-4 | 1 | 5 | 4 | 462 | 11 | 6 | 4 |
| C-5 | 3 | 4 | 0 | 2 | 475 | 0 | 1 |
| C-6 | 2 | 10 | 7 | 0 | 3 | 425 | 5 |
| C-7 | 0 | 3 | 4 | 6 | 2 | 3 | 475 |

(a)

Training Set (70%) - Confusion Matrix (RaFD Dataset)

| Actual \ Predicted | C-1 | C-2 | C-3 | C-4 | C-5 | C-6 | C-7 |
|---|---|---|---|---|---|---|---|
| C-1 | 330 | 7 | 4 | 4 | 1 | 2 | 0 |
| C-2 | 5 | 341 | 2 | 1 | 1 | 3 | 0 |
| C-3 | 5 | 2 | 312 | 1 | 1 | 2 | 2 |
| C-4 | 1 | 4 | 1 | 316 | 7 | 4 | 3 |
| C-5 | 0 | 3 | 0 | 2 | 337 | 0 | 1 |
| C-6 | 2 | 5 | 5 | 0 | 2 | 297 | 4 |
| C-7 | 0 | 3 | 4 | 3 | 1 | 2 | 330 |

(b)

Testing Set (30%) - Confusion Matrix (RaFD Dataset)

| Actual \ Predicted | C-1 | C-2 | C-3 | C-4 | C-5 | C-6 | C-7 |
|---|---|---|---|---|---|---|---|
| C-1 | 137 | 4 | 1 | 1 | 2 | 1 | 0 |
| C-2 | 3 | 135 | 1 | 1 | 0 | 1 | 0 |
| C-3 | 3 | 1 | 134 | 1 | 2 | 0 | 0 |
| C-4 | 0 | 1 | 3 | 146 | 4 | 2 | 1 |
| C-5 | 3 | 1 | 0 | 0 | 138 | 0 | 0 |
| C-6 | 0 | 5 | 2 | 0 | 1 | 128 | 1 |
| C-7 | 0 | 0 | 0 | 3 | 1 | 1 | 145 |

(c)

**Figure 5:** Confusion matrices of EWOISN-FER approach under RaFD dataset (a) Entire dataset, (b) 70% of TR data, and (c) 30% of TS data

Table 4 provides a detailed *accu*$_y$ analysis of the EWOISN-FER method on two test datasets [25]. Fig. 6 demonstrates a comparative analysis of the EWOISN-FER method with recent methodologies on the CK+ dataset. The outcomes indicated the CNN model had shown poor results with the least *accu*$_y$ of 80.47%. EWOISN-FER model has shown a maximum *accu*$_y$ of 93.65%. Next, the AlexNet model has resulted in a slightly improved *accu*$_y$ of 85.91% whereas the GoogleNet and AlexNet-SVM models have gained reasonably closer *accu*$_y$ of 86.09% and 86.49%, respectively. Though the WCRAFM model has reached near optimal *accu*$_y$ of 89.90%, the presented model exhibits maximum performance.

**Table 3:** Result analysis of the EWOISN-FER approach with distinct class labels under the RaFD dataset

| RaFD dataset | | | | | |
|---|---|---|---|---|---|
| Labels | Accuracy | Recall | Specificity | F-score | MCC |
| **Entire dataset** | | | | | |
| C-1 | 98.55 | 94.53 | 99.24 | 95.02 | 94.17 |
| C-2 | 98.40 | 96.36 | 98.75 | 94.63 | 93.71 |
| C-3 | 98.73 | 95.71 | 99.21 | 95.40 | 94.66 |
| C-4 | 98.58 | 93.71 | 99.41 | 95.06 | 94.25 |
| C-5 | 99.02 | 97.94 | 99.20 | 96.64 | 96.08 |
| C-6 | 98.67 | 94.03 | 99.38 | 94.97 | 94.21 |
| C-7 | 99.11 | 96.35 | 99.58 | 96.94 | 96.42 |
| **Average** | **98.72** | **95.52** | **99.25** | **95.52** | **94.79** |
| **Training phase (70%)** | | | | | |
| C-1 | 98.69 | 94.83 | 99.35 | 95.51 | 94.75 |
| C-2 | 98.48 | 96.60 | 98.81 | 94.99 | 94.11 |
| C-3 | 98.77 | 96.00 | 99.21 | 95.56 | 94.85 |
| C-4 | 98.69 | 94.05 | 99.46 | 95.32 | 94.57 |
| C-5 | 99.20 | 98.25 | 99.36 | 97.26 | 96.79 |
| C-6 | 98.69 | 94.29 | 99.37 | 95.04 | 94.29 |
| C-7 | 99.03 | 96.21 | 99.50 | 96.63 | 96.06 |
| **Average** | **98.79** | **95.75** | **99.29** | **95.76** | **95.06** |
| **Testing phase (30%)** | | | | | |
| C-1 | 98.22 | 93.84 | 98.96 | 93.84 | 92.80 |
| C-2 | 98.22 | 95.74 | 98.63 | 93.75 | 92.74 |
| C-3 | 98.62 | 95.04 | 99.20 | 95.04 | 94.23 |
| C-4 | 98.32 | 92.99 | 99.30 | 94.50 | 93.53 |
| C-5 | 98.62 | 97.18 | 98.85 | 95.17 | 94.39 |
| C-6 | 98.62 | 93.43 | 99.43 | 94.81 | 94.03 |
| C-7 | 99.31 | 96.67 | 99.77 | 97.64 | 97.25 |
| **Average** | **98.56** | **94.98** | **99.16** | **94.96** | **94.14** |

**Table 4:** Comparative analysis of the EWOISN-FER technique with recent methodologies under two datasets

| Accuracy (%) | | |
|---|---|---|
| Methods | CK+ | RaFD |
| EWOISN-FER | 93.65 | 98.79 |
| GoogLeNet Model | 86.09 | 95.54 |

(Continued)

**Table 4 (continued)**

| Accuracy (%) | | |
|---|---|---|
| Methods | CK+ | RaFD |
| AlexNet Model | 85.91 | 95.23 |
| AlexNet-SVM Model | 86.49 | 95.17 |
| CNN Model | 80.47 | 94.56 |
| WCRAFM | 89.90 | 96.02 |

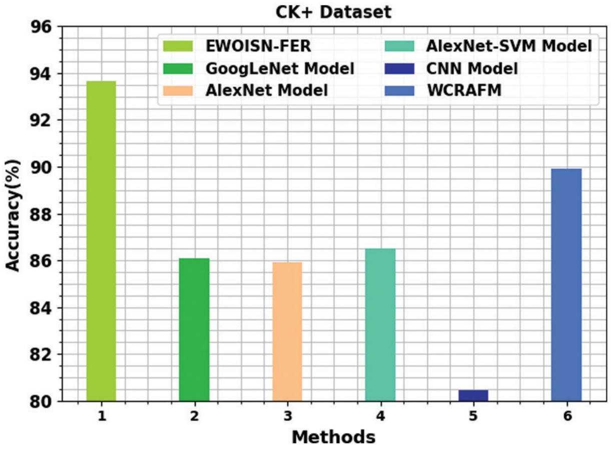


**Figure 6:** *Accu$_y$* analysis of EWOISN-FER approach under CK+ dataset

Fig. 7 establishes a comparative study of the EWOISN-FER algorithm with recent methodologies on the RaFD dataset. The outcomes denote the CNN methodology has exhibited poor results with the least *accu$_y$* of 94.56%. Then, the AlexNet approach has resulted in a slightly improved *accu$_y$* of 95.23%, whereas the GoogleNet and AlexNet-SVM methods have obtained reasonably closer *accu$_y$* of 95.54% and 95.17%, correspondingly. However, the WCRAFM method has reached near optimal *accu$_y$* of 96.02%; the presented EWOISN-FER approach has shown a maximal *accu$_y$* of 98.79%.

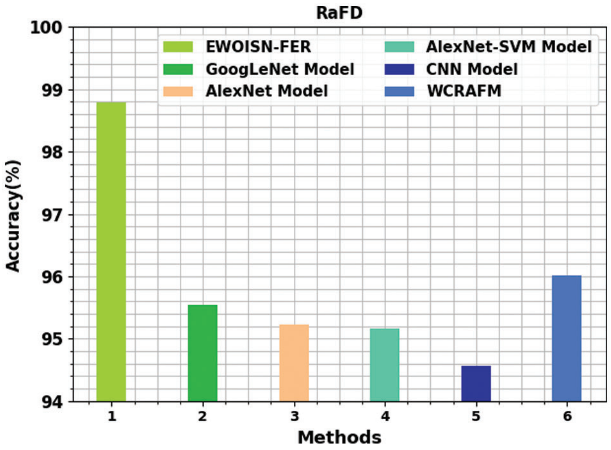


**Figure 7:** *Accu$_y$* analysis of the EWOISN-FER approach under the RaFD dataset

Therefore, the EWOISN-FER model has exhibited maximum FER performance over other DL models.

## 5 Conclusion

In this study, a novel EWOISN-FER approach was projected to recognise and classify facial emotions. The presented EWOISN-FER model employed the CLAHE technique as a pre-processing step. In addition, an improved SqueezeNet model is exploited to derive an optimal set of feature vectors, and the SGB model can execute the hyperparameter tuning process. Lastly, EWO with SAE is employed for the FER process, and the EWO algorithm appropriately chooses the SAE parameters. A wide-ranging experimental analysis is carried out to demonstrate the enhanced performance of the EWOISN-FER technique. The experimental outcomes indicate the supremacy of the presented EWOISN-FER technique. Thus, the EWOISN-FER technique can be exploited for enhanced FER outcomes. In the future, hybrid metaheuristics algorithms can be designed to improve the parameter-tuning process.

**Funding Statement:** The authors received no specific funding for this study.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

[1] S. Li and W. Deng, “Deep facial expression recognition: A survey,” *IEEE Transactions on Affective Computing*, vol. 13, no. 3, pp. 1195–1215, 2020.

[2] A. Agrawal and N. Mittal, “Using CNN for facial expression recognition: A study of the effects of kernel size and number of filters on accuracy,” *The Visual Computer*, vol. 36, no. 2, pp. 405–412, 2020.

[3] P. Giannopoulos, I. Perikos and I. Hatzilygeroudis, “Deep learning approaches for facial emotion recognition: A case study on FER-2013,” in *Advances in Hybridization of Intelligent Methods, Smart Innovation, Systems and Technologies Book Series*, Cham: Springer, vol. 85, pp. 1–16, 2018.

[4] N. Samadiani, G. Huang, B. Cai, W. Luo, C. H. Chi *et al.,* “A review on automatic facial expression recognition systems assisted by multimodal sensor data,” *Sensors*, vol. 19, no. 8, pp. 1863, 2019.

[5] P. W. Kim, “Image super-resolution model using an improved deep learning-based facial expression analysis,” *Multimedia Systems*, vol. 27, no. 4, pp. 615–625, 2021.

[6] K. Kottursamy, “A review on finding efficient approach to detect customer emotion analysis using deep learning analysis,” *Journal of Trends in Computer Science and Smart Technology*, vol. 3, no. 2, pp. 95–113, 2021.

[7] L. Schoneveld, A. Othmani and H. Abdelkawy, “Leveraging recent advances in deep learning for audio-visual emotion recognition,” *Pattern Recognition Letters*, vol. 146, pp. 1–7, 2021.

[8] A. H. Sham, K. Aktas, D. Rizhinashvili, D. Kuklianov, F. Alisinanoglu *et al.,* “Ethical AI in facial expression analysis: Racial bias,” *Signal, Image and Video Processing*, vol. 146, pp. 1–8, 2022.

[9] S. Li and Y. Bai, “Deep learning and improved HMM training algorithm and its analysis in facial expression recognition of sports athletes,” *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–12, 2022.

[10] S. Umer, R. K. Rout, C. Pero and M. Nappi, “Facial expression recognition with trade-offs between data augmentation and deep learning features,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 2, pp. 721–735, 2022.

[11] O. M. Nezami, M. Dras, L. Hamey, D. Richards, S. Wan *et al.,* “Automatic recognition of student engagement using deep learning and facial expression,” in *Joint European Conf. on Machine Learning and Knowledge Discovery in Databases*, Cham: Springer, vol. 11908, pp. 273–289, 2020.

[12] G. Bargshady, X. Zhou, R. C. Deo, J. Soar, F. Whittaker *et al.,* “Enhanced deep learning algorithm development to detect pain intensity from facial expression images,” *Expert Systems with Applications*, vol. 149, pp. 113305, 2020.

[13] W. M. Alenazy and A. S. Alqahtani, “Gravitational search algorithm based optimized deep learning model with diverse set of features for facial expression recognition,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 2, pp. 1631–1646, 2021.

[14] J. Liu, Y. Feng and H. Wang, “Facial expression recognition using pose-guided face alignment and discriminative features based on deep learning,” *IEEE Access*, vol. 9, pp. 69267–69277, 2021.

[15] S. Minaee, M. Minaei and A. Abdolrashidi, “Deep-emotion: Facial expression recognition using attentional convolutional network,” *Sensors*, vol. 21, no. 9, pp. 3046, 2021.

[16] J. H. Kim, B. G. Kim, P. P. Roy and D. M. Jeong, “Efficient facial expression recognition algorithm based on hierarchical deep neural network structure,” *IEEE Access*, vol. 7, pp. 41273–41285, 2019.

[17] U. Kuran and E. C. Kuran, “Parameter selection for CLAHE using multi-objective cuckoo search algorithm for image contrast enhancement,” *Intelligent Systems with Applications*, vol. 12, pp. 200051, 2021.

[18] H. J. Lee, I. Ullah, W. Wan, Y. Gao and Z. Fang, “Real-time vehicle make and model recognition with the residual SqueezeNet architecture,” *Sensors*, vol. 19, no. 5, pp. 982, 2019.

[19] J. H. Friedman, “Stochastic gradient boosting,” *Computational Statistics & Data Analysis*, vol. 38, no. 4, pp. 367–378, 2002.

[20] N. Esmaeili, F. E. K. Saraei, A. E. Pirbazari, F. S. Tabatabai-Yazdi, Z. Khodaee *et al.,* “Estimation of 2, 4-dichlorophenol photocatalytic removal using different artificial intelligence approaches,” *Chemical Product and Process Modeling*, vol. 18, no. 1, pp. 1–17, 2022, https://doi.org/10.1515/cppm-2021-0065.

[21] K. Zhang, J. Zhang, X. Ma, C. Yao, L. Zhang *et al.,* “History matching of naturally fractured reservoirs using a deep sparse autoencoder,” *SPE Journal*, vol. 26, no. 4, pp. 1700–1721, 2021.

[22] S. R. Kanna, K. Sivakumar and N. Lingaraj, “Development of deer hunting linked earthworm optimization algorithm for solving large scale traveling salesman problem,” *Knowledge-Based Systems*, vol. 227, pp. 107199, 2021.

[23] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar *et al.,* “The extended cohn-kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression,” in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition-Workshops*, San Francisco, CA, USA, pp. 94–101, 2010.

[24] O. Langner, R. Dotsch, G. Bijlstra, D. H. J. Wigboldus, S. T. Hawk *et al.,* “Presentation and validation of the radboud faces database,” *Cognition and Emotion*, vol. 24, no. 8, pp. 1377–1388, 2010.

[25] B. F. Wu and C. H. Lin, “Daptive feature mapping for customizing deep learning based facial expression recognition model,” *IEEE Access*, vol. 6, pp. 12451–12461, 2018.