Computer Modeling in
Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2022.023312

ARTICLE

Human-Computer Interaction Using Deep Fusion Model-Based Facial Expression Recognition System

Saiyed Umer[1,*], Ranjeet Kumar Rout[2], Shailendra Tiwari[3], Ahmad Ali AlZubi[4], Jazem Mutared Alanazi[4] and Kulakov Yurii[5]

[1]Department of Computer Science & Engineering, Aliah University, Kolkata, 700156, India

[2]Department of Computer Science and Engineering, National Institute of Technology, Srinagar, Jammu and Kashmir, 190006, India

[3]Department of Computer Science & Engineering, Thapar University, Patiala, 147004, India

[4]Computer Science Department, King Saud University, Riyadh, 11451, Saudi Arabia

[5]Department of Computer Engineering, National Technical University of Ukraine, Igor Sikorsky Kyiv Polytechnic Institute, Kyiv, 03056, Ukraine

*Corresponding Author: Saiyed Umer. Email: saiyed.umer@aliah.ac.in

Received: 19 April 2022 Accepted: 21 June 2022

ABSTRACT

A deep fusion model is proposed for facial expression-based human-computer Interaction system. Initially, image preprocessing, i.e., the extraction of the facial region from the input image is utilized. Thereafter, the extraction of more discriminative and distinctive deep learning features is achieved using extracted facial regions. To prevent overfitting, in-depth features of facial images are extracted and assigned to the proposed convolutional neural network (CNN) models. Various CNN models are then trained. Finally, the performance of each CNN model is fused to obtain the final decision for the seven basic classes of facial expressions, i.e., fear, disgust, anger, surprise, sadness, happiness, neutral. For experimental purposes, three benchmark datasets, i.e., SFEW, CK+, and KDEF are utilized. The performance of the proposed system is compared with some state-of-the-art methods concerning each dataset. Extensive performance analysis reveals that the proposed system outperforms the competitive methods in terms of various performance metrics. Finally, the proposed deep fusion model is being utilized to control a music player using the recognized emotions of the users.

KEYWORDS

Deep learning; facial expression; emotions; recognition; CNN

1 Introduction

Facial expressions are an important way of communication for understanding emotions in human beings. These human emotions are identified by various traits such as text, Electroencephalography, speech, and face. These emotions performed by these traits are more noticeable and observable [1]. There is a wide range of applications of these emotions in Computer Vision applications, such as sentiment analysis for pain analysis in the human body, security, criminal interrogation, patient communication, psychological treatment, etc. Facial emotions play an essential role in the various emotional traits that contribute to more exciting expressions. According to Ekman et al. [2], there are seven basic expressions on the human face such as fear, neutral, sadness, disgust, anger, happiness, and surprise. In facial expression recognition, these expressions are recognized. The capturing of these expressions is less invasive and tangible than other emotional traits. Moreover, the intensity variations over the facial region differ due to these expressions. Some examples of these seven basic facial expressions are shown in Fig. 1.

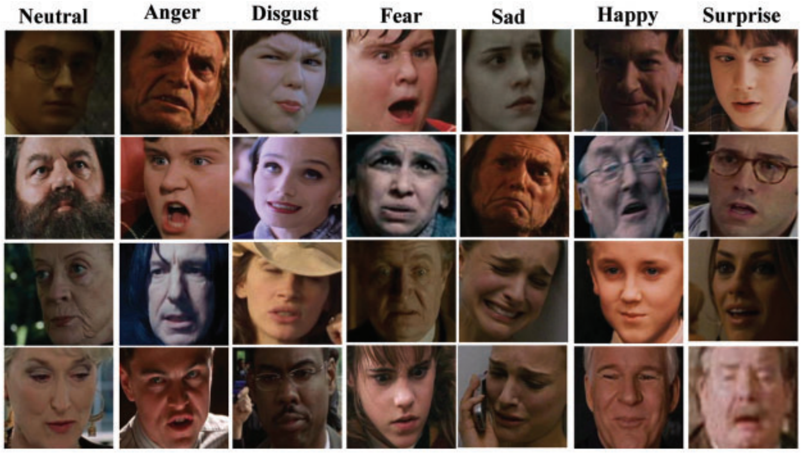


Figure 1: Some basic human facial expression images

The facial features play an essential role in identifying the emotions on the human face. The seven basic emotions (expressions) where each expression has its significance based on its intensity on the face. Moreover, it has also been observed that there are some mixed emotions [3] which are the combinations of these basic seven emotions. Capturing expressions under unconstrained environments is less invasive and tangible and requires non-interruption while the person is far or moving from some distance. So, over the past few years, emotion recognition using facial expressions has brought much more attention to affective computing and cognitive science research areas. There are various aspects of FER (facial expression recognition) models in human-computer interaction, augmented reality, driving assistant models, etc. During the implementation of the FER model, the categorical subject model derives the emotions in terms of discrete primary emotion [4].

The facial expressions are obtained from the eye, mouth, and cheeks portion of the face region. In contrast, the other parts of the facial region support enhancing the expression level in the face region. The research areas of facial expression are in the study of affective computing [5] which is an application of computer vision problems. In affective computing research areas, the recognition of facial expressions is a categorical-based model. The analysis of the coding of facial action units is a continuous-based model. We have considered the categorical model for the facial expression recognition (FER) model in this work. The FER model includes both images, and video-based recognition [6]. The spatial information is extracted as feature representation in the image-based FER model, whereas both spatial and temporal features are considered in the video-based FER model. The spatial features have high distinctiveness and discriminating power than temporal features [6]. Using a small number of training instances in genetic programming for face image classification has been proposed by Bi et al. [7]. Similarly, the multi-objective genetic programming for feature learning for face recognition system has been proposed by Bi et al. [8].

Initially, Ekman et al. [9] defined six facial expressions such as fear, anger, disgust, happiness, sadness, and surprise and performed emotion recognition for the FER model. Further, Ekman et al. proposed the concept of the Facial action coding model [10] to measure the facial movement using facial action points. The recognition of facial expressions mainly depends on the types of feature extraction, which are classified as (i) Appearance-based, (ii) Geometric-based feature representation [11]. Many works have been done based on these appearances and geometrical features from the facial images. For example, Castrillon et al. [12] designed a gender classification model by considering several models of analyzing the texture patterns within the facial region in. By incorporating the RGB colour channel features along with depth, informative features about the facial region proposed for the FER model in [13]. In their FER model, Yan et al. [14] employed the image filtering based feature representation for the low-resolution based image samples. Sadeghi et al. [15] built the histogram distance learning-based feature representation for the proposed model. Makhmudkhujaev et al. [16] presented the various directional descriptors with prominent local patterns as features from the facial images. These Models and employed techniques follow local to global feature representation schemes, and most of these features are structural and statistical-based features.

In the computer vision research areas, the above-discussed features have succeeded in solving object recognition, biometric identification, face recognition, instance-based recognition, and texture classification problems. But due to current state-of-the-art problems, these models have limited performance. Learning robust and discriminative low-rank representations for face recognition with occlusion has been proposed in [17]. In the current cutting-edge problems, the deep learning-based approaches have gained great success in solving problems either in computer vision or in business world research areas. The deep learning-based approach is described as a neural network with many layers and parameters. This approach defines some fundamental network architectures such as unsupervised pre-trained networks [18], convolutional [19], recurrent [20], and recursive neural networks [21]. Among these networks, the convolutional neural networks [19] are used for the FER model. Ye et al. [22] proposed a region-based convolutional fusion network for the facial expression recognition model. By identifying relationships among different regions of a facial image, the FERS is built by Sun et al. [23]. Lai et al. [24] developed CNN models to recognize facial expressions. A FER model based on local fine-grained temporal and global spatial appearance features using a global-local CNN network has been built in [25]. Hence, with the several benefits of deep learning-based CNN architectures, a facial expression recognition model has been proposed in this work that can predict the challenging expressions in the facial region in both controlled and uncontrolled environments. There are several existing works in the FER model using image/video-based, but still, there are several challenging issues [26]. During image acquisition of the facial region, the images suffer from motion blur, noise artifacts, occlusion by the hair, illumination variations, and occlusion by accessories such as glass, makeup, scarf, and mark. Accepting these challenges, we have developed a categorical model-based facial expression recognition model using images in this work. The contributions of this paper are summarized as follows:

- Deep fusion-based facial expression recognition model is proposed for human-computer interaction.
- Proposed deep learning models extract more distinctive and discriminant features from the facial images.
- To improve the recognition performance of the proposed model, some influential factors such as data augmentation, fine-tuning the hyper-parameters, and multi-resolution with progressive image sizing are employed to improve the recognition model's performance.

- Different deep learning-based approaches are fused at the post-classification stage to obtain the final decision for the recognition model.
- The proposed model is tested on three benchmark datasets: SFEW, CK+, and KDEF, and the performance and comparison with the existing state-of-the-art models due to these datasets have been demonstrated with the proposed system.

This paper is organized as Section 2 describes each step of the proposed Modelology; The experimental dataset description, results in discussion, and comparisons have been demonstrated in Section 3; Finally, the findings of this research have been concluded in Section 4.

2 Proposed Scheme

This section discusses implementing the proposed deep fusion-based facial expression recognition (FER) model. Depending upon the input face, the proposed model predicts the type of expressions among the seven facial expressions (anger, sadness, surprise, disgust, happiness, neutral, and fear) classes. The proposed model is decomposed into four steps: (i) the first step is image preprocessing, where the face region ($\mathscr{F}$) is detected from the input image $\mathscr{I}_{m\times n}$, (ii) in the second step, deep learning-based approaches have been employed for feature learning, and classification purposes, (iii) in the third step several parameters regarding the performance improvement of the proposed model have been discussed, (iv) for the usability of different training models the scores due to these training models are fused to obtain the final decision for the facial expression class in the fourth component. The working principle of the proposed model is represented in Fig. 2.

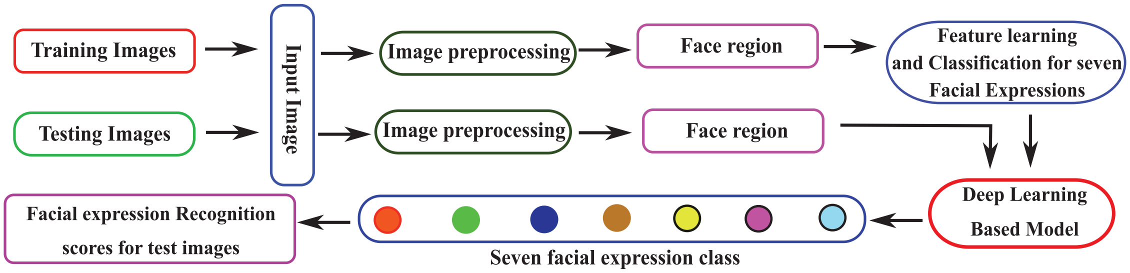

Training Images
Testing Images
Input Image
Image preprocessing
Image preprocessing
Face region
Face region
Feature learning and Classification for seven Facial Expressions
Deep Learning Based Model
Seven facial expression class
Facial expression Recognition scores for test images

Figure 2: Block diagram of the proposed system

2.1 Image Preprocessing

During an unconstrained imaging environment, noise, illuminations, variations in poses, and cluttered backgrounds are mainly the problems, and these may arise some irrelevant features. So, to extract more relevant and valuable features, the face region has been detected as a region of interest from the input image. The extracted face region has been normalized to similar dimensions that the same dimensional feature vector can be extracted. In this work for face detection, a tree-structured part model [27] has been employed, which works for all variants of face poses. This model computes sixty-eight landmark points for the frontal face, while thirty-nine landmark points have been extracted for the profile face. Then these landmark points are employed to calculate the face region from the input image. The Bilinear image interpolation technique has been applied to the detected face region for the normalization purpose. The face detection process for the proposed model is depicted in Fig. 3.

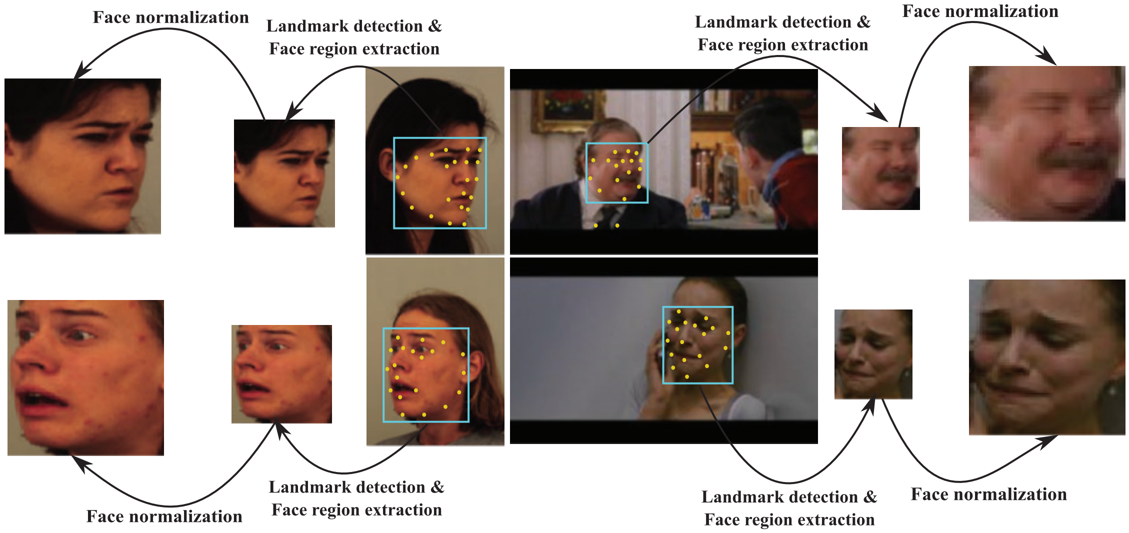


Figure 3: Face preprocessing for the proposed model

2.2 Feature Learning Followed by Classification

The proposed facial expression recognition model belongs to a pattern recognition problem. The objective of this problem is to extract more distinctive and discriminative features as a feature vector from the facial region images. Then, the classifiers learn these feature vectors to derive a model that will predict the class for facial expressions in the facial region. There exist several structural and statistical-based approaches [28] to solving the FER problem. But nowadays, deep learning-based approaches have gained tremendous success in solving the various issues and problems in the computer vision research area. The deep learning-based approaches work in an encapsulated way by the combined effect of both feature learning and classification task. There are several deep learning-based approaches, and among them, the convolutional neural network (CNN) [29] based models have been employed in this work. The CNN based approaches are based on the core building blocks of convolutional layers, pooling layers, fully connected layers, and dense layers [29]. The convolutional layer is the layer where the input is an image that has been convoluted with several distinct filters (kernels). Then, the convoluted images are computed as feature maps concerning the kernels. The computation of these feature maps increases the complexity of the CNN network by increasing the image size and the number of kernels employed for that convolutional layer in the network.

During feature learning, the weights in the kernel are adjusted as parameter settings. The benefits of the convolutional layer are (i) it performs local connectivity by obtaining correlations between neighbours pixels, (ii) weight-sharing in the same feature map reduces the complexity of the network, and (iii) it maintains the shift-invariant properties about the location of the objects. So, the input and output in the convolution layer is $\mathscr{F}_{n\times n\times 3} \xrightarrow{w_{k\times l\times l}} \mathscr{F}'_{n\times n\times k}$, where $\mathscr{F}_{n\times n\times 3}$ is a 3-color channel image, $w_{k\times l\times l}$ be the k number of kernels with each kernel has $l \times l$ size, $\mathscr{F}'_{n\times n\times k}$ be the derived feature maps while each feature map has $n \times n$ size. To extract more discriminanting features from the feature maps, the max-pooling layers [30] have been employed. The technique of max-pooling layer downsamples the matrices of the features to its half size if 2×2 filter size has been employed. In this layer, the filter of size 2×2 strides over the feature map and compute in its region the maximum value first horizontally and then vertically for the computation of discriminant features in the matrix maps to the next layer. The benefits of using max-pooling layers are (i) it decreases the parameters, (ii) reduces the computational overheads, (iii) makes the process of parameter settings faster within the network, and (iv) avoids overfitting problems.

The addition of the fully connected layer is performed at the end of the network to perform a classification task for the learned features from the previous layer. It ensures all neurons from the previous layers are fully connected to the next layer in the form of a 1-dimensional feature map. Another layer is the dense layer [31] which is also a type of fully-connection layer. The main differences between fully-connected and dense layers are (i) linear operations are being performed in a dense layer, and (ii) the dense layer computes the matching scores for each input sample as outcomes using the softmax activation function [32] at the end of the network. In addition to these layers, some other layers, such as batch normalization [33] and dropout layers [34] have also been adopted in this work. The batch normalization layer also reduces the computational overheads while maintaining the homogeneity in the batch of data for learning the parameters in the network. The dropout layer is being used to ignore some randomly selected neurons in the network from learning, i.e., the weights to that neurons will not be updated during training. The use of dropout layers in the network prevents overfitting problems and combines the predictions of various neural nets.

Here using convolutional layers, max-pooling layers, fully connected, batch normalization, and dropout layers, we have built some convolutional neural networks (CNNs) architectures. The proposed CNN architectures contain the combination of these layers. The diagram for the first CNN architecture is shown in Fig. 4. This figure shows five blocks where each block has a sequence of layers, i.e., Convolutional + Activation + Maxpooling + Batch-Normalization. After the five blocks, there are two fully connected layers (Dense + Dropout). For better understanding and clarity, the number of convolutional layers with kernel size, number of kernels, the number of max-pooling layers, batch normalization, dropouts, feature map's output shape, and the number of parameters concerning each layer is reported in Table 1. Similarly, the second CNN architecture is shown in Fig. 5, and the explanation of the layers and parameters for this network is reported in Table 2. From Tables 1 and 2, it may be concluded that some activation functions such as ReLu (Rectified Linear Unit), Softmax, and Adam as optimizer have been adopted for learning the parameters in the network. Both CNN architectures have been learned for seven class FER problems in this work.

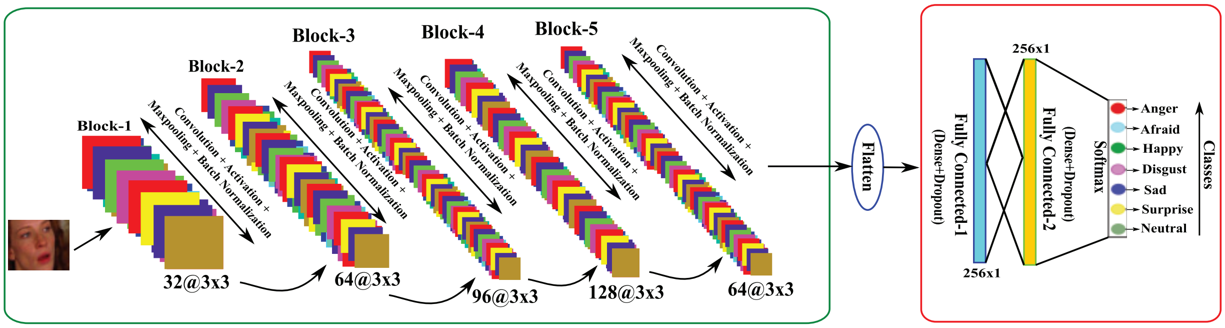

Block-1
Block-2
Block-3
Block-4
Block-5
Convolution + Activation + Maxpooling + Batch Normalization
Convolution + Activation + Maxpooling + Batch Normalization
Convolution + Activation + Maxpooling + Batch Normalization
Convolution + Activation + Maxpooling + Batch Normalization
Convolution + Activation + Maxpooling + Batch Normalization
32@3x3
64@3x3
96@3x3
128@3x3
64@3x3
Flatten
Fully Connected-1
(Dense+Dropout)
256x1
Fully Connected-2
(Dense+Dropout)
256x1
Softmax
Anger
Afraid
Happy
Disgust
Sad
Surprise
Neutral
Classes

Figure 4: Proposed CNN_1 architecture for the FER model

Table 1: Description of parameters, layers, and output shapes for CNN_1 architecture

Layer	Output shape	Image size	Parameters
		Block-1	
Convolution2D (3 × 3@32) (Activation: Relu)	$(n, n, 32)$	(96, 96, 32)	$((3 \times 3 \times 3) + 1) \times 32 = 896$
Maxpooling2D (2 × 2)	$(n_1, n_1, 32)$	(48, 48, 32)	0
Batch normalization	$(n_1, n_1, 32)$	(48, 48, 32)	$4 \times 32 = 128$
		Block-2	
Convolution2D (3 × 3@64) (Activation: Relu)	$(n_1, n_1, 64)$	(48, 48, 64)	$((3 \times 3 \times 32) + 1) \times 64 = 18496$
Maxpooling2D (2 × 2)	$(n_2, n_2, 64)$	(24, 24, 64)	0
Batch normalization	$(n_2, n_2, 64)$	(24, 24, 64)	$4 \times 64 = 256$
		Block-3	
Convolution2D (3 × 3@96) (Activation: Relu)	$(n_2, n_2, 96)$	(24, 24, 96)	$((3 \times 3 \times 64) + 1) \times 96 = 55392$
Maxpooling2D (2 × 2)	$(n_3, n_3, 96)$	(12, 12, 96)	0
Batch normalization	$(n_3, n_3, 96)$	(12, 12, 96)	$4 \times 96 = 384$
		Block-4	
Convolution2D (3 × 3@128) (Activation: Relu)	$(n_3, n_3, 96)$	(12, 12, 128)	$((3 \times 3 \times 96) + 1) \times 128 = 110720$
Maxpooling2D (2 × 2)	$(n_4, n_4, 128)$	(6, 6, 128)	0
Batch normalization	$(n_4, n_4, 128)$	(6, 6, 128)	$4 \times 128 = 512$
		Block-5	
Convolution2D (3 × 3@64) (Activation: Relu)	$(n_4, n_4, 64)$	(6, 6, 64)	$((3 \times 3 \times 128) + 1) \times 64 = 73792$
Maxpooling2D (2 × 2)	$(n_5, n_5, 64)$	(3, 3, 64)	0
Batch normalization	$(n_5, n_5, 64)$	(3, 3, 64)	$4 \times 64 = 256$
		Fully connected	
Flatten	$3 \times 3 \times 64 = 576$		0
Dense + Dropout	256		$(576 + 1) \times 256 = 147712$
Dense + Dropout	256		$(256 + 1) \times 256 = 65792$
Dense + Softmax	7		$(256 + 1) \times 7 = 1799$
Total parameters			**476,135**

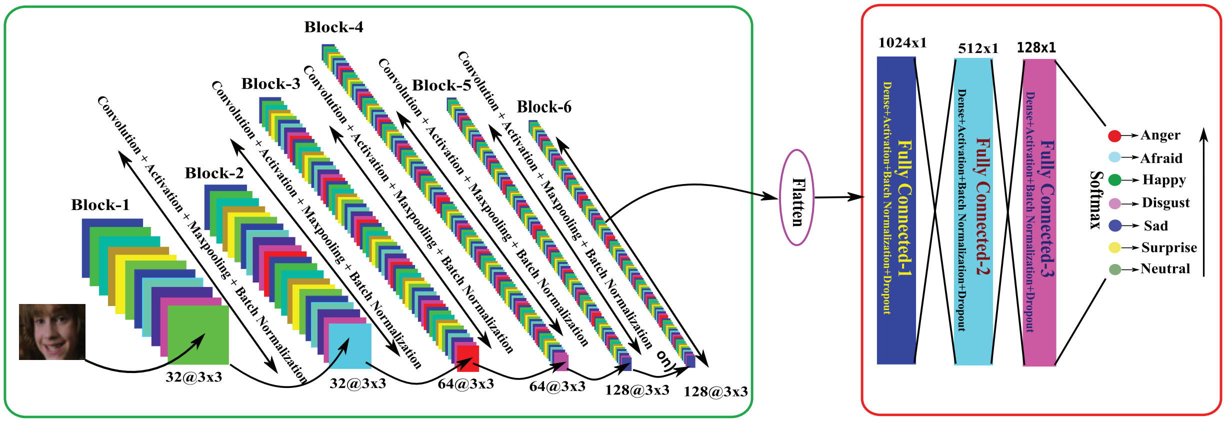

Block-1
Block-2
Block-3
Block-4
Block-5
Block-6
Convolution + Activation + Maxpooling + Batch Normalization
Convolution + Activation + Maxpooling + Batch Normalization
Convolution + Activation + Batch Normalization
Convolution + Activation + Maxpooling + Batch Normalization
Convolution + Activation + Batch Normalization
Convolution + Activation + Maxpooling + Batch Normalization
only
32@3x3
32@3x3
64@3x3
64@3x3
128@3x3
128@3x3
Flatten
1024x1
512x1
128x1
Fully Connected-1
Dense+Activation+Batch Normalization+Dropout
Fully Connected-2
Dense+Activation+Batch Normalization+Dropout
Fully Connected-3
Dense+Activation+Batch Normalization+Dropout
Softmax
Anger
Afraid
Happy
Disgust
Sad
Surprise
Neutral

Figure 5: Proposed CNN_2 architecture for the FER model

Table 2: Description of parameters, layers, and output shapes for CNN_2 architecture

Layer	Output shape	Image size	Parameters
		Block-1	
Convolution2D (3 × 3@32) (Activation: Relu)	$(n, n, 32)$	(96, 96, 32)	$((3 \times 3 \times 3) + 1) \times 32 = 896$
Batch normalization	$(n, n, 32)$	(96, 96, 32)	$4 \times 32 = 128$
		Block-2	
Convolution2D (3 × 3@32) (Activation: Relu)	$(n, n, 32)$	(96, 96, 32)	$((3 \times 3 \times 32) + 1) \times 32 = 9248$
Batch normalization	$(n, n, 32)$	(96, 96, 32)	$4 \times 32 = 128$
Maxpooling2D (2 × 2)	$(n_1, n_1, 32)$	(48, 48, 32)	0
		Block-3	
Convolution2D (3 × 3@64) (Activation: Relu)	$(n_1, n_1, 64)$	(48, 48, 64)	$((3 \times 3 \times 32) + 1) \times 64 = 18496$
Batch normalization	$(n_1, n_1, 64)$	(48, 48, 64)	$4 \times 64 = 256$
		Block-4	
Convolution2D (3 × 3@64) (Activation: Relu)	$(n_1, n_1, 64)$	(48, 48, 64)	$((3 \times 3 \times 64) + 1) \times 64 = 36928$
Batch Normalization	$(n_1, n_1, 64)$	(48, 48, 64)	$4 \times 256 = 256$
Maxpooling2D (2 × 2)	$(n_2, n_2, 64)$	(24, 24, 64)	0
Dropout	$(n_2, n_2, 64)$	(24, 24, 64)	0
		Block-5	
Convolution2D (3 × 3@128) (Activation: Relu)	$(n_2, n_2, 128)$	(24, 24, 128)	$((3 \times 3 \times 64) + 1) \times 128 = 73856$
Batch normalization	$(n_2, n_2, 128)$	(24, 24, 128)	$4 \times 128 = 512$

(Continued)

Table 2 (continued)

Layer	Output shape	Image size	Parameters
	Block-6		
Convolution2D (3 × 3@128) (Activation: Relu)	$(n_2, n_2, 128)$	(24, 24, 128)	((3 × 3 × 128) + 1) × 128 = 147584
Batch normalization	$(n_2, n_2, 128)$	(24, 24, 128)	4 × 128 = 512
Maxpooling2D (2 × 2)	$(n_3, n_3, 128)$	(12, 12, 128)	0
Dropout	$(n_3, n_3, 128)$	(12, 12, 128)	0
	Fully connected		
Flatten	12 × 12 × 128 = 18432		0
Dense + ReLu + Batch normalization + Dropout	1024		(18432 + 1) × 1024 = 18,875,392 + (4 × 1024) = 18,879,488
Dense + ReLu + Batch normalization + Dropout	512		(1024 + 1) × 512 = 524800
Dense + ReLu + Batch normalization + Dropout	256		(512 + 1) × 256 = 131328
Dense + ReLu	7		(256 + 1) × 7 = 1799
Total parameters			**19,829,287**

2.3 Factors Affecting the Recognition System's Performance

2.3.1 Image Augmentation

In machine learning, the image augmentation technique has been employed for increasing the number of samples that corresponds to each input image, and it is done by applying several filtering and affine transformation techniques [35]. The benefits of using the image augmentation techniques are (i) handling the overtraining situation of the convolution neural networks, (ii) reducing the overfitting problems, and (iii) helping the process of fine-tuning for learning the hyper-parameters to get better CNN performance. The image augmentation techniques generate several samples without changing the image fidelity and their visual qualities [36]. The generated samples enhance the CNN learning parameters, and learning these better models can be predicted to recognise the required problems. There are several data augmentation techniques and among them, we have employed image filtering techniques such as Bilateral Filtering [37], Unsharp Filter [38], Sharpening Filter [39], Affine transformation [40]: reflection, rotation, scaling [41], shearing [42], zooming [43], filling [44], and horizontally flipping [45] techniques applied on images. Hence, by applying these data augmentation techniques, there are eighteen (original + seventeen augmented) images are generated to correspond to each training image. The image augmentation algorithm for the proposed model has been demonstrated in Fig. 6. Algorithm 1 shows the step-by-step computation of the image augmentation technique.

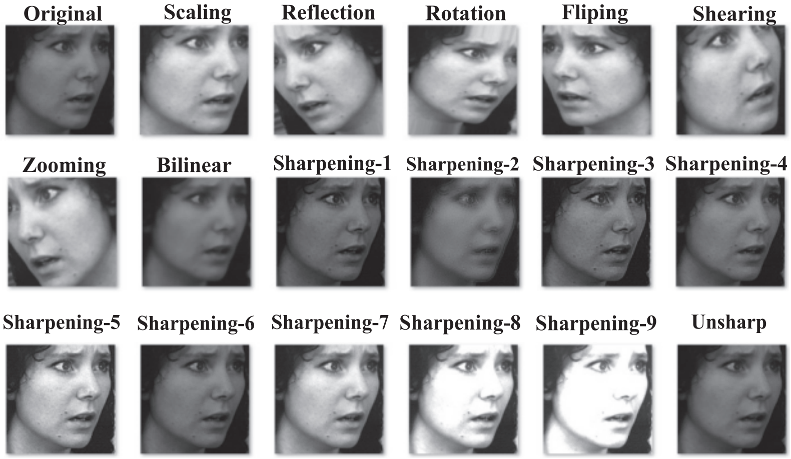


Figure 6: Demonstration of image augmentation applied on each image $\mathscr{F}$ in the proposed model

```
Algorithm 1: Image Augmentation
Input: Face Region $\mathscr{F}$
Output: $\mathscr{F}_{aug}$
1. Apply Bilateral Filtering [37] on $\mathscr{F}$ to get $\mathscr{F}_1$
2. Apply Unsharp Filtering [38] on $\mathscr{F}$ to get $\mathscr{F}_2$
3. Apply Sharpening Filters [39] with different filter mask such as $\{\omega_1, \omega_2, \omega_3, \ldots, \omega_9\}$ on $\mathscr{F}$ to get
   $\mathscr{F}_3, \ldots, \mathscr{F}_{11}$
4. Apply image rotation [40] on $\mathscr{F}$ to get $\mathscr{F}_{12}$
5. Apply image scaling [41] on $\mathscr{F}$ to get $\mathscr{F}_{13}$
6. Apply image shearing [42] on $\mathscr{F}$ to get $\mathscr{F}_{14}$
7. Apply image zooming [43] on $\mathscr{F}$ to get $\mathscr{F}_{15}$
8. Apply image filling [44] on $\mathscr{F}$ to get $\mathscr{F}_{16}$
9. Apply image horizontal flipping [45] on $\mathscr{F}$ to get $\mathscr{F}_{17}$
10. Final augmented set for each $\mathscr{F}$ is $\mathscr{F}_{aug} = \{\mathscr{F}_1, \ldots, \mathscr{F}_{17}\}$
```

2.3.2 Fine Tuning

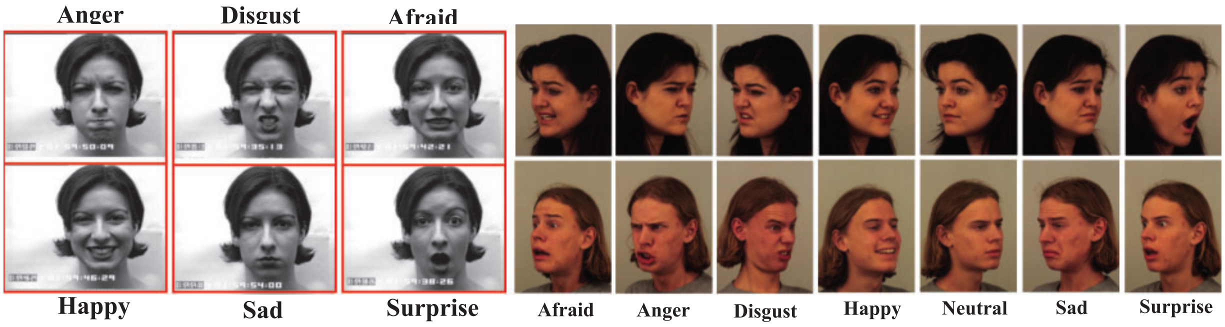
In deep learning approaches, the performance of CNN may be improved by fine-tuning the hyper-parameters of the trained model [46]. The fine-tuning considers a trained network model and initializes it by its trained weight, and uses the data from the same domain for further training of that model to a new model. The fine-tuning technique speeds up the training process while also overcoming the small dataset size problem. In fine-tuning, either the whole layers of the trained network are retrained, or some of the layers of the trained model are frozen, and the remaining layers are trained. The performance of the proposed CNN models can also be improved by tuning the hyper-parameters such as learning rate, L2-regularization, batch size, and increasing the model depth [47]. Moreover, increasing the image resolution, i.e., progressive resizing of the face region, can also improve the performance of the proposed CNN model.

2.3.3 Scores Fusion

The techniques under this category are sum-rule, and product-rule based fusion models [48]. These fusion techniques are based on scores which are obtained in this work from the proposed CNN trained models with respect to each test sample. Let assume that for any test sample t_i, $s_1 \in \mathbb{R}^{1\times M}$ and $s_2 \in \mathbb{R}^{1\times M}$, M be the class number, are two score vectors obtained from the proposed CNN_1 and CNN_2 facial expression trained models. Then the final score vector for the test sample t_i using (i) sum-rule based fusion technique is given by $s = s_1 + s_2$, and (ii) product-rule based fusion technique is given by $s = s_1 \times s_2$. Now the final score vector s is used to find the predicted class label for the test sample t_i.

3 Experimental Results

This section explains the experiments performed for the proposed facial expression recognition model (FERS). Here we have employed the three benchmark datasets for experimental purposes. The first employed dataset is Cohn-Kanade Extended (CK+) [49] which is composed of 593 short videos from 123 subjects with different lighting and aging variations. For experimental purposes, 981 image samples were selected from 123 subjects, where the image samples are of six (Surprise, Happiness, Fear, Disgust, Sadness, and Anger) facial expression classes. Fig. 7a demonstrates some image samples from this dataset. Karolinska directed emotional faces (KDEF) [50] is our second dataset which is a seven facial expression class dataset. This dataset comprises 4900 emotional images of human faces with a collection of 35 females and 35 males. In this work, we have downloaded 2447 images, and 1222 images are used for training, while the remaining 1225 are used for testing purposes. Fig. 7b demonstrates some images of this dataset. The third dataset is Static Facial Expressions in the Wild (SFEW) [51] which is also a seven facial expression class dataset. This dataset selects frames from the AFEW (Acted Facial Expressions in the Wild) dataset, a dynamic temporal facial expression dataset. This dataset covers several challenges of FER problems. The images of this dataset face several challenges such as varied focus, different resolution of face, various head poses, significant variation in age, considerable variation in occlusions, etc. In this dataset total of 700 frames are extracted from the AFEW dataset, where each frame has been labelled as sadness, surprise, happiness, fear, disgust, anger, and neutral expression class. During experimentation, 346 images were selected as training images, and 354 images were selected as testing images. Some images of this dataset have been shown in Fig. 7c. Table 3 summarizes the detailed description of the employed datasets for the proposed model.



(a) (b)

Figure 7: (Continued)

(c)

Figure 7: Some image samples from (a) CK+, (b) KDEF, and (c) SFEW datasets

Table 3: Summarizing the employed dataset for the proposed model

Dataset	Class	Number of training samples	Number of testing samples
CK+	6	500	481
KDEF	7	1222	1225
SFEW	7	346	354

Here, CK+ and KDEF datasets have been randomly partitioned, with 50% of the samples from each class being used to form the training set while the remaining 50% of samples from each class form testing set. In the SFEW dataset, the number of training-testing samples is already mentioned in [51].

3.1 Results and Discussion

The implementation of the proposed model has been performed in Python on Ubuntu 16.04 LTS O/S version with Intel Core i7 processor 3.20 GHz and 32 ′GB RAM. For deep learning approaches, several packages have been employed from Keras [52], and for building the CNN architecture, the Theano Python library has been employed. The performance of the proposed model is shown in the correct recognition rate, i.e., accuracy in % $\left(= \frac{Number\ of\ samples\ classified\ correctly\ by\ model}{Total\ number\ of\ samples\ tested\ by\ model} \times 100\right)$.

During face preprocessing, the face region $\mathscr{F}$ using the TSPM model is detected from the given input image $\mathscr{I}$. Then the extracted face region $\mathscr{F}$ is normalized to $\mathscr{N} \times \mathscr{N}$ fixed size such that a fixed dimensional feature vector can be extracted from each $\mathscr{F}_{\mathscr{N}\times\mathscr{N}}$. Then the extracted facial regions from the training samples undergo the proposed convolutional neural network architectures, i.e., CNN_1 and CNN_2. During experimentation, the size of the face region $\mathscr{N} \times \mathscr{N}$ is 48 × 48 while the batch size and the number of epochs vary. To improve the performance of the proposed model, the data augmentation techniques (discussed in Section 2.3.1) have been applied on each $\mathscr{F}_{48\times48}$ using Algorithm 1 and hence for each $\mathscr{F}$, $\{\mathscr{F}_1, \ldots, \mathscr{F}_{18}\}$ augmented images are obtained.

- **Different loss functions impact:** At first, the experiment was performed by training the CNN_1 architecture with $\mathscr{F}_{48\times48}$ input images by varying the different loss functions to minimize the errors in the network. Here, the mean squared error (MSE) [53], binary cross-entropy [54], and Hinge loss [55] loss functions have been considered for the measuring their impact on the performance of facial expression recognition (FER) system using the proposed CNN_1 model. These performances have been shown in Fig. 8 that shows for the binary cross-entropy loss function, and the performance is better. Hence, the binary cross-entropy loss function is considered for further experiment.

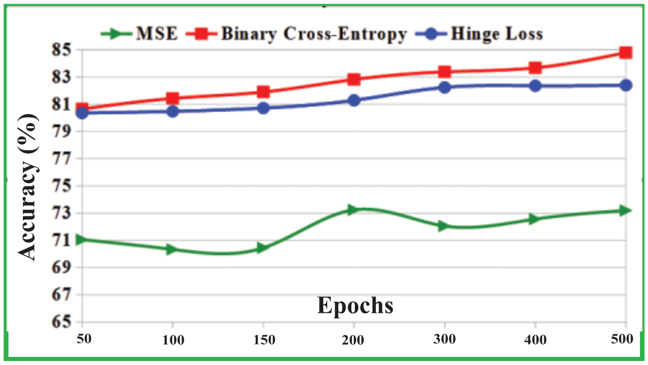
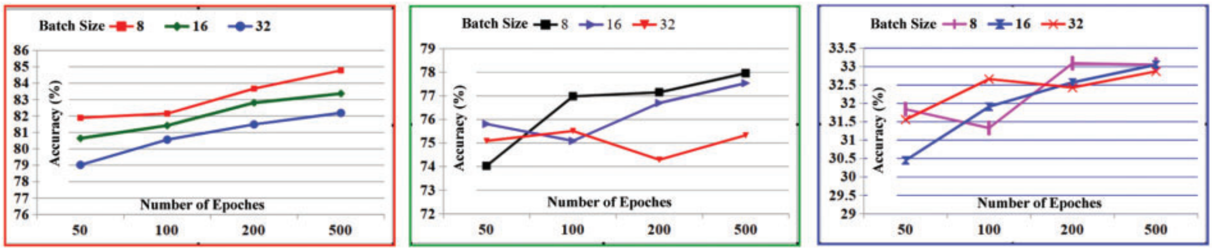


Figure 8: Effectiveness of different loss functions on the performance of CNN_1 models for CK+ dataset

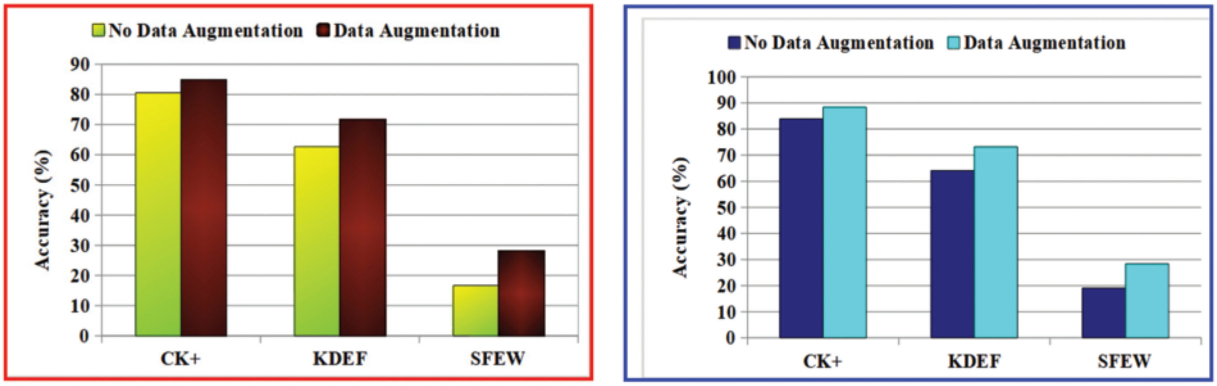
- **Batch *vs.* epoch impact:** In this work, it is seen that the recognition of the proposed model also improves due to the variation of $\{8, 16, 32\}$ batch sizes with corresponding $\{50, 100, 200, 500\}$ epochs. Fig. 9 demonstrates the effectiveness of batch sizes and the number of epochs over the performance of the proposed model due to CNN_1 models for CK+, KDEF, and SFEW datasets. From this figure, it has been observed that the performance improves with the increase of epochs employed for learning the trained model, while the batch size is more or less effective over the performance of the proposed model. For this work, it has been observed that for batch size 8, the performance of the FER model is much better for CK+, KDEF, and SFEW datasets. Hence for further experiments, we have employed eight batch sizes of training samples with 500 epochs for learning the parameters of CNN_1 and CNN_2 architectures.



(a) (b) (c)

Figure 9: Effectiveness of trade-off between batch sizes and number of epochs on the performance of CNN_1 models: (a) CK+, (b) KDEF, and (c) SFEW dataset

- **Data augmentation impact:** The effectiveness of data augmentation on the performance of the proposed model is depicted in Fig. 10. It is found that the data augmentation techniques have increased the performance of the proposed model. Hence, for further implementation of the proposed model, data augmentation is implemented on each training sample to increase the training sample's size for better learning of the CNN models.



(a) (b)

Figure 10: Effectiveness of data augmentation over the performance of the proposed model due to: (a) CNN_1 and (b) CNN_2 models

- **Multiscaling and Multiresolution impact:** Hence, the recognition performance of the proposed model is reported in Table 4 where the usefulness and effectiveness of multiscaling and multiresolution of images (progressive image resizing) with variable sizes such as $\mathscr{F}_{48\times48}$, $\mathscr{F}_{64\times64}$, and $\mathscr{F}_{96\times96}$ has been shown. In this experiment, we used the Mini-Batch Gradient Descent optimization technique [56] with batch sizes such as 8 and the number of epochs is 500 for reporting the performance. From the Table 4, it has been observed that for CK+, KDEF, and SFEW datasets, the performance of the proposed model increases with increasing the image size in both the CNN architectures and also the performance is slightly better due to CNN_2 than CNN_1 model. So, the proposed model attains the highest performance, 95.89% for CK+, 78.27% for KDEF, and 35.31% for the SFEW dataset due to the CNN_2 model and using the CNN_1 model, the proposed model attains 93.41% for CK+, 77.76% for KDEF, and 33.05% for SFEW dataset.

Table 4: Performance due to CNN_1 and CNN_2 models in terms of accuracy (%) with varying image sizes

	CNN_1		
Image-size	CK+	KDEF	SFEW
$\mathscr{F}_{48\times48}$	84.78	71.57	28.14
$\mathscr{F}_{64\times64}$	87.32	74.34	29.89
$\mathscr{F}_{96\times96}$	93.41	77.76	33.05
	CNN_2		
Image-size	CK+	KDEF	SFEW
$\mathscr{F}_{48\times48}$	88.34	73.19	28.31
$\mathscr{F}_{64\times64}$	92.16	75.85	29.78
$\mathscr{F}_{96\times96}$	95.89	77.27	35.31

- **Fine tuning impact:** Here, the performance of the proposed CNN_1 and CNN_2 architectures are improved by applying the method of fine-tuning to tune the hyper-parameters of the trained model. This fine-tuning method considers the trained CNN_1 and CNN_2 network model, initializes its trained weight, and re-trained the whole network by freezing some of the layers to reduce the computational overhead of training hyper-parameters of the trained model. Hence the impact of fine-tuning for the proposed FERS has been shown in Fig. 11.

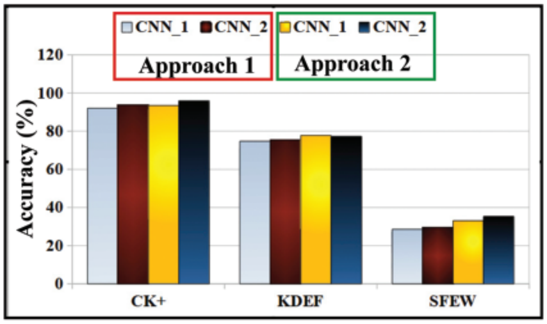


Figure 11: Impact of fine tuning the hyper-parameters of the trained CNN_1 and CNN_2 models on the performance of the proposed FERS

- **Scores fusion impact:** To adapt the effectiveness of both CNN models, the performance of the proposed model has been fused such that the scores due to CNN_1 and CNN_2 models have been fused to derive a final decision for the proposed model. Here score level fusion techniques such as sum-rule and product-rule-based methods have been used. Here the sum-rule based score level fusion is defined as $s = s_i + s_j$, whereas the product-rule based score level fusion is defined as $s = s_i \times s_j$, s_i and s_j be the scores for a test sample due to CNN_1 and CNN_2 models, respectively. The fused performance of the proposed system due to CNN_1 and CNN_2 models have been shown in Table 5 concerning each employed facial expression dataset. This table shows that each dataset has attained better performance after fusion, and the product-rule has achieved better performance than the sum-rule-based score level fusion technique. Hence, for CK+, KDEF, and SFEW datasets, the proposed model has obtained 96.89%, 82.35%, and 41.73% accuracy, respectively. For these performances, the confusion matrix performance for CK+, KDEF, and SFEW datasets has been shown in Fig. 12 for a better understanding of the classification of each test sample in its corresponding class.

Table 5: Effectiveness of score fusion on the performance of CNN_1 and CNN_2 in terms of accuracy (%)

Method	CK+	KDEF	SFEW
$\text{Fusion}_{\text{sum}}$	96.18	80.45	39.09
$\text{Fusion}_{\text{product}}$	96.89	82.35	41.73

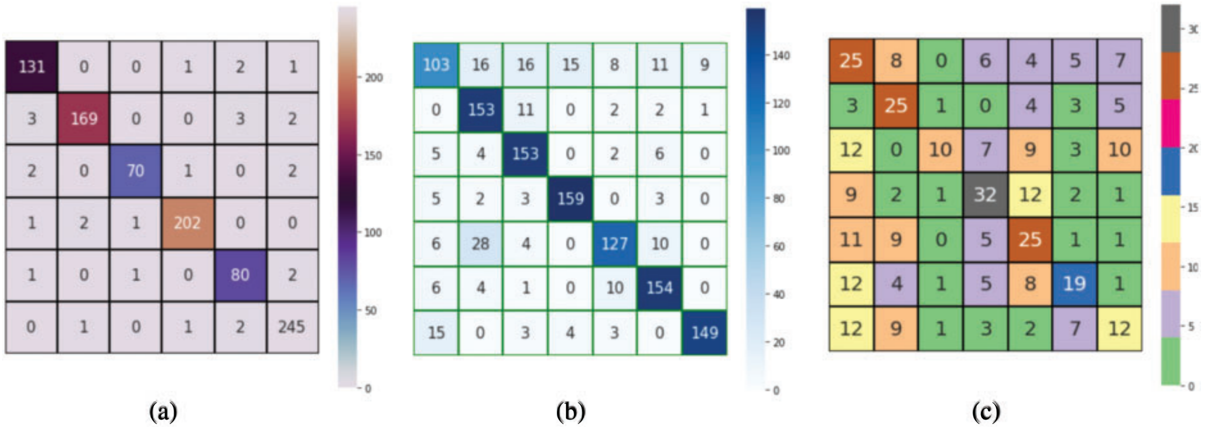


Figure 12: Confusion matrix performance for (a) CK+, (b) KDEF, and (c) SFEW dataset due to the fused performance of CNN_1 and CNN_2 models

3.2 Comparisons

Here, during comparison with other existing CNN models, the input to these CNN models is the same facial region as used by the proposed system. Also, the same data augmentation techniques have been employed for all the compering methods employed here. Hence, the performance comparisons reported herewith have been made under the same training-testing protocol used by the proposed methodology. Table 6 shows the performance of analysis of Res-Net50 [57], Inception-v3 [58], Sun et al. [59], and the proposed model on CK+ dataset. It is found that the proposed model achieves better performance with 96.89% performance. Table 7 shows KDEF dataset analysis, and it is found that the proposed model shows an average 82.35% improvement over the existing models. Table 8 shows the comparative analysis on SFEW dataset. It is found that the proposed model achieves better performance than the existing models by showing an average enhancement of 41.73% over the competitive models.

Table 6: Comparison of performance for CK+ dataset (CV is cross validation)

Model	Accuracy (%)	Remarks
Sun et al. [23]	94.67	10-fold CV with 510 images for 6 expression classes
ResNet50 [57]	91.87	10-fold CV with 981 images for 6 expression classes
Inception-v3 [58]	94.07	10-fold CV with 981 images for 6 expression classes
Vgg16 [60]	76.90	10-fold CV with 981 images for 6 expression classes
Fard et al. [61]	95.51	10-fold CV with 981 images for 6 expression classes
Hussein et al. [62]	96.71	10-fold CV with 981 images for 6 expression classes
Proposed	**96.89**	10-fold CV with 981 images for 6 expression classes

Table 7: Comparison of performance for KDEF dataset (CV is cross validation)

Model	Accuracy (%)	Remarks
Vgg16 [60]	65.08	10-fold CV with 980 images for 7 expression classes
Rao et al. [63]	74.05	10-fold CV with 720 images for 7 expression classes
Inception-v3 [58]	75.04	10-fold CV with 980 images for 7 expression classes
ResNet50 [57]	72.32	10-fold CV with 980 images for 7 expression classes
Zavarez et al. [64]	72.55	10-fold CV with 980 images for 7 expression classes
Sun et al. [59]	82.24	10-fold CV with 490 images for 7 expression classes
Fard et al. [61]	80.76	10-fold CV with 981 images for 7 expression classes
Hussein et al. [62]	81.87	10-fold CV with 981 images for 7 expression classes
Proposed	**83.27**	10-fold CV with 980 images for 7 expression classes

Table 8: Comparison of performance for SFEW dataset (here competing models used same training-testing protocols)

Model	Accuracy (%)
ResNet50 [57]	24.98
Vgg16 [60]	24.78
Liu et al. [65]	26.14
Inception-v3 [58]	29.52
Dhall et al. [66]	39.13
Fard et al. [61]	33.45
Hussein et al. [62]	37.31
Proposed	**41.73**

Apart from these, the proposed deep fusion model is used to control the music player. Depending upon the human emotions, the music player is controlled. Based upon the userâs emotion, a song is selected from the given class. The proposed model can be better used for disabled persons to change their moods. During the real-time testing, it was found that on a computer with 2.4 GHz, the proposed model can predict 28 frames per second. Therefore, the proposed model can be used for other human-computer interface-based applications.

4 Conclusion

A facial expression recognition model was proposed under controlled and uncontrolled imaging environments. The images considered here are captured in the unconstrained environment, such as motion blurred, hazy, rotated, pose invariant, moving at a distance, and off-angle. The implementation of the proposed model was divided into three components: (i) image preprocessing, (ii) feature learning with classification, and (iii) performance fusion. The face region was extracted during image preprocessing as this is the region of interest for the proposed model. The extracted face region undergoes feature learning with classification tasks. Here for feature learning with classification task, two convolutional neural networks (CNNs) have been proposed where each CNN was learned with the facial regions of the training samples. In contrast, the learned CNN model was employed to obtain the classification performance using the facial region of testing samples. Finally, the performances obtained from both the CNN models were fused to build the final recognition model. Several factors affecting the CNN performance, such as data augmentation, fine-tuning the hyper-parameters, and multi-resolution with progressive image sizing, were also performed during experimentation. The proposed model was verified on three well-known datasets, i.e., CK+, KDEF, and SFEW. Comparative analysis revealed that the proposed model outperforms the state-of-the-art models in various performance metrics. Finally, the proposed deep fusion model was utilized to control the music player using the recognized emotions of the user.

Funding Statement: This work was supported by the Researchers Supporting Project (No. RSP-2021/395), King Saud University, Riyadh, Saudi Arabia.

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

References

1. Jaimes, A., Sebe, N. (2007). Multimodal human–computer interaction: A survey. *Computer Vision and Image Understanding, 108(1–2),* 116–134. DOI 10.1016/j.cviu.2006.10.019.
2. Ekman, P., Matsumoto, D., Friesen, W. V. (1997). Facial expression in affective disorders. In: *What the face reveals: Basic and applied studies of spontaneous expression using the facial action coding system (FACS),* vol. 2, pp. 331–342.
3. Tanaka, J. W., Kaiser, M. D., Butler, S., Le Grand, R. (2012). Mixed emotions: Holistic and analytic perception of facial expressions. *Cognition & Emotion, 26(6),* 961–977. DOI 10.1080/02699931.2011.630933.
4. Hupont, I., Cerezo, E., Baldassarri, S. (2010). Sensing facial emotions in a continuous 2D affective space. *2010 IEEE International Conference on Systems, Man and Cybernetics*, Istanbul, Turkey, IEEE.
5. Cohn, J. F., de la Torre, F. (2015). Automated face analysis for affective computing. In: Calvo, R. A., D'Mello, S. K., Gratch, J., Kappas, A. (Eds.), *The Oxford handbook of affective computing*, pp. 131–150. Oxford University Press.
6. Chen, J., Chen, Z., Chi, Z., Fu, H. (2016). Facial expression recognition in video with multiple feature fusion. *IEEE Transactions on Affective Computing, 9(1),* 38–50. DOI 10.1109/TAFFC.2016.2593719.
7. Bi, Y., Xue, B., Zhang, M. (2022). Using a small number of training instances in genetic programming for face image classification. *Information Sciences, 593,* 488–504. DOI 10.1016/j.ins.2022.01.055.
8. Bi, Y., Xue, B., Zhang, M. (2021). Multi-objective genetic programming for feature learning in face recognition. *Applied Soft Computing, 103,* 107152. DOI 10.1016/j.asoc.2021.107152.
9. Ekman, P., Friesen, W. V. (1971). Constants across cultures in the face and emotion. *Journal of Personality and Social Psychology, 17(2),* 124–129. DOI 10.1037/h0030377.
10. Friesen, E., Ekman, P. (1978). Facial action coding system: A technique for the measurement of facial movement. *Palo Alto, 3,* 1–20.
11. Ko, B. C. (2018). A brief review of facial emotion recognition based on visual information. *Sensors, 18(2),* 401. DOI 10.3390/s18020401.
12. Castrillón-Santana, M., de Marsico, M., Nappi, M., Riccio, D. (2017). Meg: Texture operators for multi-expert gender classification. *Computer Vision and Image Understanding, 156,* 4–18. DOI 10.1016/j.cviu.2016.09.004.
13. Lee, K., Lee, E. C. (2019). Comparison of facial expression recognition performance according to the use of depth information of structured-light type RGB-D camera. *Journal of Ambient Intelligence and Humanized Computing, 10,* 1–17. DOI 10.1007/s12652-019-01278-2.

14. Yan, Y., Zhang, Z., Chen, S., Wang, H. (2020). Low-resolution facial expression recognition: A filter learning perspective. *Signal Processing, 169,* 107370. DOI 10.1016/j.sigpro.2019.107370.
15. Sadeghi, H., Raie, A. A. (2019). Histogram distance metric learning for facial expression recognition. *Journal of Visual Communication and Image Representation, 62,* 152–165. DOI 10.1016/j.jvcir.2019.05.004.
16. Makhmudkhujaev, F., Abdullah-Al-Wadud, M., Iqbal, M. T. B., Ryu, B., Chae, O. (2019). Facial expression recognition with local prominent directional pattern. *Signal Processing: Image Communication, 74,* 1–12.
17. Gao, G., Yang, J., Jing, X. Y., Shen, F., Yang, W. et al. (2017). Learning robust and discriminative low-rank representations for face recognition with occlusion. *Pattern Recognition, 66,* 129–143. DOI 10.1016/j.patcog.2016.12.021.
18. Erhan, D., Courville, A., Bengio, Y., Vincent, P. (2010). Why does unsupervised pre-training help deep learning? *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, Sardinia, Italy.
19. Krizhevsky, A., Sutskever, I., Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems, 25,* 1–9.
20. Schuster, M., Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing, 45(11),* 2673–2681. DOI 10.1109/78.650093.
21. Socher, R., Lin, C. C. Y., Ng, A. Y., Manning, C. D. (2011). Parsing natural scenes and natural language with recursive neural networks. *ICML*, pp. 129–136. Bellevue, Washington, USA.
22. Ye, Y., Zhang, X., Lin, Y., Wang, H. (2019). Facial expression recognition via region-based convolutional fusion network. *Journal of Visual Communication and Image Representation, 62,* 1–11. DOI 10.1016/j.jvcir.2019.04.009.
23. Sun, X., Xia, P., Zhang, L., Shao, L. (2020). A ROI-guided deep architecture for robust facial expressions recognition. *Information Sciences, 522,* 35–48. DOI 10.1016/j.ins.2020.02.047.
24. Lai, Z., Chen, R., Jia, J., Qian, Y. (2020). Real-time micro-expression recognition based on resnet and atrous convolutions. *Journal of Ambient Intelligence and Humanized Computing, 11,* 1–12. DOI 10.1007/s12652-020-01779-5.
25. Yu, M., Zheng, H., Peng, Z., Dong, J., Du, H. (2020). Facial expression recognition based on a multi-task global-local network. *Pattern Recognition Letters, 131,* 166–171. DOI 10.1016/j.patrec.2020.01.016.
26. Martinez, B., Valstar, M. F. (2016). Advances, challenges, and opportunities in automatic facial expression recognition. *Advances in Face Detection and Facial Image Analysis*, pp. 63–100. Switzerland: Springer International Publishing.
27. Sardar, A., Umer, S., Pero, C., Nappi, M. (2020). A novel cancelable facehashing technique based on non-invertible transformation with encryption and decryption template. *IEEE Access, 8,* 105263–105277. DOI 10.1109/Access.6287639.
28. Umer, S., Dhara, B. C., Chanda, B. (2018). An iris recognition system based on analysis of textural edgeness descriptors. *IETE Technical Review, 35(2),* 145–156. DOI 10.1080/02564602.2016.1265904.
29. Hossain, S., Umer, S., Asari, V., Rout, R. K. (2021). A unified framework of deep learning-based facial expression recognition system for diversified applications. *Applied Sciences, 11(19),* 9174. DOI 10.3390/app11199174.
30. Vedaldi, A., Zisserman, A. (2016). *VGG convolutional neural networks practical*, vol. 66, pp. 1–14. Department of Engineering Science, University of Oxford.
31. Targ, S., Almeida, D., Lyman, K. (2016). Resnet in resnet: Generalizing residual architectures. arXiv preprint arXiv:1603.08029.
32. Liang, X., Wang, X., Lei, Z., Liao, S., Li, S. Z. (2017). Soft-margin softmax for deep classification. *International Conference on Neural Information Processing*, pp. 413–421. Long Beach, California, Springer.
33. Ioffe, S. (2017). Batch renormalization: Towards reducing minibatch dependence in batch-normalized models. *Advances in Neural Information Processing Systems, 30,* 1–9.

34. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research, 15(1),* 1929–1958.
35. Hernández-García, A., König, P. (2018). Further advantages of data augmentation on convolutional neural networks. *International Conference on Artificial Neural Networks*, Rhodes, Greece, Springer.
36. Perez, L., Wang, J. (2017). The effectiveness of data augmentation in image classification using deep learning. arXiv preprint arXiv:1712.04621.
37. Paris, S., Kornprobst, P., Tumblin, J., Durand, F. (2009). *Bilateral filtering: Theory and applications.* Boston-Delft: Now Publishers Inc.
38. Polesel, A., Ramponi, G., Mathews, V. J. (2000). Image enhancement via adaptive unsharp masking. *IEEE Transactions on Image Processing, 9(3),* 505–510. DOI 10.1109/83.826787.
39. Pardo-Igúzquiza, E., Chica-Olmo, M., Atkinson, P. M. (2006). Downscaling cokriging for image sharpening. *Remote Sensing of Environment, 102(1–2),* 86–98. DOI 10.1016/j.rse.2006.02.014.
40. Asano, T., Bitou, S., Motoki, M., Usui, N. (2007). In-place algorithm for image rotation. *International Symposium on Algorithms and Computation*, pp. 704–715. Sendai, Japan, Springer.
41. Wu, R., Yan, S., Shan, Y., Dang, Q., Sun, G. (2015). Deep image: Scaling up image recognition. arXiv preprint arXiv:1501.02876.
42. Tanter, M., Touboul, D., Gennisson, J. L., Bercoff, J., Fink, M. (2009). High-resolution quantitative imaging of cornea elasticity using supersonic shear imaging. *IEEE Transactions on Medical Imaging, 28(12),* 1881–1893.
43. Battiato, S., Gallo, G., Stanco, F. (2002). A locally adaptive zooming algorithm for digital images. *Image and Vision Computing, 20(11),* 805–812.
44. de Queiroz, R. L. (2000). On data filling algorithms for mrc layers. *Proceedings 2000 International Conference on Image Processing (Cat. No. 00CH37101)*, vol. 2. Vancouver, BC, Canada, IEEE.
45. Iliyasu, A. M., Le, P. Q., Dong, F., Hirota, K. (2012). Watermarking and authentication of quantum images based on restricted geometric transformations. *Information Sciences, 186(1),* 126–149.
46. Fan, H., Zheng, L., Yan, C., Yang, Y. (2018). Unsupervised person re-identification: Clustering and fine-tuning. *ACM Transactions on Multimedia Computing, Communications, and Applications, 14(4),* 1–18.
47. Shin, H. C., Roth, H. R., Gao, M., Lu, L., Xu, Z. et al. (2016). Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE Transactions on Medical Imaging, 35(5),* 1285–1298.
48. He, M., Horng, S. J., Fan, P., Run, R. S., Chen, R. J. et al. (2010). Performance evaluation of score level fusion in multimodal biometric systems. *Pattern Recognition, 43(5),* 1789–1800.
49. Lucey, P., Cohn, J. F., Kanade, T., Saragih, J., Ambadar, Z. et al. (2010). The extended cohn-kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*, San Francisco, CA, IEEE.
50. Lundqvist, D., Flykt, A., Öhman, A. (1998). The karolinska directed emotional faces (KDEF). *CD ROM from Department of Clinical Neuroscience, Psychology Section, Karolinska Institutet, 91(630),* 2.
51. Dhall, A., Goecke, R., Lucey, S., Gedeon, T. (2011). Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark. *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pp. 2106–2112. Barcelona, IEEE.
52. Chollet, F. (2015). keras, GitHub. https://github.com/fchollet/keras.
53. Ye, A. (2022). *A deep dive into keras. Modern deep learning design and application development*, pp. 1–48. Berkeley, CA: Springer, Apress.
54. Liu, D., Zhao, J., Wu, J., Yang, G., Lv, F. (2022). Multi-category classification with label noise by robust binary loss. *Neurocomputing, 482,* 14–26.
55. Janthakal, S., Hosalli, G. (2022). A granular parakeratosis classification using svm hinge and cross validation. *Journal of Applied Science and Engineering, 26(1),* 35–42.

56. Ioffe, S., Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167.
57. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z. (2016). Rethinking the inception architecture for computer vision. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826. Las Vegas, NV, USA.
58. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A. A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. *Thirty-First AAAI Conference on Artificial Intelligence*, San Francisco, California USA.
59. Sun, Z., Hu, Z. P., Wang, M., Zhao, S. H. (2017). Discriminative feature learning-based pixel difference representation for facial expression recognition. *IET Computer Vision, 11(8),* 675–682. DOI 10.1049/iet-cvi.2016.0505.
60. Simonyan, K., Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
61. Fard, A. P., Mahoor, M. H. (2022). Ad-corre: Adaptive correlation-based loss for facial expression recognition in the wild. *IEEE Access, 10,* 26756–26768. DOI 10.1109/ACCESS.2022.3156598.
62. Hussein, H. I., Dino, H. I., Mstafa, R. J., Hassan, M. M. (2022). Person-independent facial expression recognition based on the fusion of hog descriptor and cuttlefish algorithm. *Multimedia Tools and Applications, 81(8),* 11563–11586. DOI 10.1007/s11042-022-12438-6.
63. Rao, Q., Qu, X., Mao, Q., Zhan, Y. (2015). Multi-pose facial expression recognition based on surf boosting. *2015 International Conference on Affective Computing and Intelligent Interaction (ACII)*, pp. 630–635. Xi'an, China, IEEE.
64. Zavarez, M. V., Berriel, R. F., Oliveira-Santos, T. (2017). Cross-database facial expression recognition based on fine-tuned deep convolutional network. *2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pp. 405–412. Niterói, Brazil, IEEE.
65. Liu, M., Li, S., Shan, S., Chen, X. (2013). Au-aware deep networks for facial expression recognition. *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, pp. 1–6. Shanghai, China, IEEE.
66. Dhall, A., Ramana Murthy, O., Goecke, R., Joshi, J., Gedeon, T. (2015). Video and image based emotion recognition challenges in the wild: Emotiw 2015. *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pp. 423–426. Seattle, Washington, USA.