Int. j. inf. tecnol. (December 2021) 13(6):2419–2430
https://doi.org/10.1007/s41870-021-00803-x

ORIGINAL RESEARCH

An efficient CNN approach for facial expression recognition with some measures of overfitting

Mayank Kumar Rusia[1] · **Dushyant Kumar Singh**[1]

Received: 10 May 2021 / Accepted: 2 September 2021 / Published online: 17 September 2021

Abstract A person's emotion can be represented through facial expressions in non-vocal communication. Nowadays, automatic facial expression recognition systems have attracted myriad interest in applications such as face biometric-based authentication, behavior analysis (psychology), health monitoring (cerebral palsy), recommendation systems, and many others. Deep learning-based solutions have become the most-handy method to solve any image-video processing problems in recent times. Nevertheless, these CNN models include many hidden layers with complex predefined mathematical functions, resulting in increased complexity. Therefore, deep architecture poses a challenge to deal with a large number of learning parameters. This manuscript proposes two customized-CNN models, named Proposed_Model_1 and Proposed_Model_2, to classify universal facial expressions without overfitting. In this paper, the effect of hyper-parameters such as activation function, learning rate, kernel size, and convolutional block are investigated and optimized efficiently. Experimental results reveal that both of our proposed models outperform other existing methods considering all universal facial expressions, with an accuracy of 67.24% and 66.61%, respectively, on the well-known benchmark public dataset (Facial Expression Recognition-2013).

✉ Mayank Kumar Rusia
mayank.qip18@mnnit.ac.in

Dushyant Kumar Singh
dushyant@mnnit.ac.in

[1] Computer Science and Engineering Department, Motilal Nehru National Institute of Technology Allahabad, Prayagraj, Uttar Pradesh, India

Keywords Facial emotion analysis · Convolutional neural network · Computer vision · Deep learning · Image processing · Biometrics

1 Introduction

Automatic recognition of facial expressions creates a great demand worldwide considering many applications such as face biometric-based authentication, health care [1], video surveillance [2, 3], behavior analysis [4], recommendation system [5], human–computer interaction [6, 7] and more [8]. Facial expression recognition consists of three individual words: facial (face), expression, and recognition. The term expression [9] signifies the emotions that a person is currently having. Facial expression is an art to convey emotions related to particular feelings through the muscular movement [10] of facial attributes. More specifically, expressions are behavioral consequences from the influence of emotions, represented through the external part of the body (i.e., the face). However, the response to the same emotion varies enormously from person to person. The term recognition (identification) [11] refers to the similarity of a test face with the faces available in the database.

Numerous facial expressions recognition methods have been proposed to date. Traditional machine learning methods involve handcrafted features, which often leads to a time-consuming feature extraction process with less efficient results. Thus, these methods are not convenient and much effective for our problem. In contrast, deep neural network-based solutions are mostly preferred due to the availability of large datasets and easy access to high computational tools. Convolutional Neural Network (CNN) is a fast, efficient, and well-known customizable architecture for image classification problems [12, 13]. Deep

Table 1 Comparison of various state-of-the-art techniques for facial expression recognition

Reference	Dataset Used/ Expressions covered	Proposed Methodology	Accuracy	Findings	
				Strengths	Weaknesses
Saeed et al. [16]	Cohn– Kanade database (CK)/(06 Expression)	AERS, GSNMF, TPTSR, Gabor filter, LBP	Best accuracy Anger—95%	High robustness More prone to analyzing facial geometry	Covered less than universal classes of facial expressions High computational complexity Poor generalization
Wan. et al. [17]	FER-2013/07 Expressions	AlexNet VGGNet 8-layer-net 11-layer-net	54.8% 63.1% 62.8% 65.3%	Dropout is used to extend convergence time Less chance of overfitting	Training parameters require extra space Heavy network Poor generalization
Liu et al. [18]	FER-2013/07 Expressions	CNN single subnet CNN subnet Ensemble	62.44%, 65.3%	Mini Batch Gradient Descent is used to train the subnet	Poor generalization The error on each CNN subnet is ignored
Agarwal et al. [19]	FER-2013/07 Expressions	CNN(Model-I)	65.23% (fixed kernel fixed filter)	Kernel size and the number of filters are used to obtain good results Human-like accuracy	The structure of Model-II is non-uniform Not suitable for hardware implementation
		CNN(Model-II)	65.77% (fixed kernel variable filter)		
Yu and Zhang [20]	FER-2013 (Training), and SEFW 2.0 (Fine Tuning and Testing)/07 Expressions	Ensemble of three CNN models	61.29%	Cross dataset-based learning process reduces the hinge loss and log-likelihood loss This method can be used for the development of affect-aware games	Overfitting exists Influenced by noise Comparatively less accuracy
Van Huynh et al. [21]	Acted Facial Expressions in the Wild (AFEW)/ 07 Expressions	MLCNN and EyeGaze	51.96%	Deep learning is applied for the extraction of facial emotion Eye movement-based features are extracted through eye-landmarks and gaze features	Poor accuracy An open-source framework is used to extract eye-gaze features Authenticity issue of these open-source frameworks
Chiranjeevi et al. [22]	CK + and Internally collected dataset SRID-1/SRID-2/04 Expressions	Action unit-based Key emotions. Constrained Local Model (CLM)	66% (SRID-2) 72% (CK +) Both with key emotion tracking	Face varies in both pose and scale over time. Thus, the main focus was on neutral facial emotion detection	Not generalized adequately Only a Neutral class of facial expressions exists
Perveen, et al. [23]	JAFFE database/07 Expressions	Multi-level backpropagation and Neural Network classifier (Statistical and Spatial features)	70%	A hybrid approach is applied to combine both the statistical features and the facial spatial features Best utilization of the traditional method	Only 70 images are considered here for testing Only 22 images are considered for each class of facial expressions. Thus, this method is not suitable for deep learning applications

AERS Automated Expression Recognition System, *GSNMF* Graph-preserving Sparse Nonnegative Matrix Factorization algorithm, *TPTSR* Two-Phase Test Sample Representation technique, *CNN* Convolutional Neural Network, *LBP* Local Binary Pattern, *FER* Facial Expression Recognition, *CK +* Extended Cohn-Kanade, *JAFFE* Japanese Female Facial Expression architecture can better understand the non-linearity in feature space and relatively classify different emotions [14]. Thus, CNN can effectively learn intricate features using convolution and pooling operations. However, the CNN architecture also poses a significant challenge of overfitting when dealing with large-scale data and complex network architectures. Overfitting [15] is the effect of weights initially assigned or automatically adopted by the network during the training process. Sometimes, these weights are fine-tuned to fit well with our training data but perform poorly for the unseen data. To solve this issue, we conducted several experiments and found some significant measures to reduce the problem of overfitting (a.k.a. data memorization) (see Sect. 3). Another issue with existing methods of facial expression recognition is the classification accuracy, which is highly influenced by various factors, including the experimental setup, as listed below:

- How many classes of facial expressions have been considered for the experiments?
- How can we be sure that the recorded emotions are expressed deliberately or elicited naturally?
- What types of props and underlying models have been used to create a natural environment to capture real emotions?
- What are the measures adopted for constrained and unconstrained experimental setup?

Thus, we considered the well-known FER-2013 dataset for our use case experiments to achieve remarkable accuracy taking into account these factors. However, we also examined the performance of other recent facial expressions datasets to validate our results. All these factors motivate us to complete this research work. We aim to propose a novel solution capable of detecting and classifying all seven universal classes of facial expressions (i.e., Anger, Disgust, Fear, Happy, Sad, Surprise, and Neutral) without overfitting. We have designed a relatively simple architecture consisting of a small number of hidden layers with fine-tuning of hyperparameters.

This manuscript is organized into five sections. Section 1 delineates a brief introduction with strong motivation to do this work. Section 2 presents the literature review with a comparative analysis of various state-of-the-art approaches to facial expression recognition. Section 3 details the proposed methodology of data preprocessing, feature extraction, and classification. Section 4 demonstrates the experimental results for both of our proposed models, followed by the analysis of the results. Section 5 summarizes the entire work with inspiration for future work.

2 Literature review

The face recognition system always seeks to render the desired faces without expression to provide adequate results. However, this is often not the case in real life. Different facial expressions can positively affect the accuracy of a face recognition system. Thus, systems need to be upgraded to detect and recognize facial expressions efficiently. However, many researchers have provided appropriate solutions to identify facial expressions; most of them confront the issues of overfitting and generalization. Saeed et al. [16] represented a comparative study of various feature extraction methods such as Automated Expression Recognition System (AERS), Graph-preserving Sparse Nonnegative Matrix Factorization (GSNMF) algorithm, Two-Phase Test Sample Representation (TPTSR) technique, Gabor filter, and image sequencing-based methods for detecting facial expressions. Wan et al. [17] proposed four models influenced by the VGGNet and AlexNet for the FER-2013 dataset. Liu et al. [18] proposed

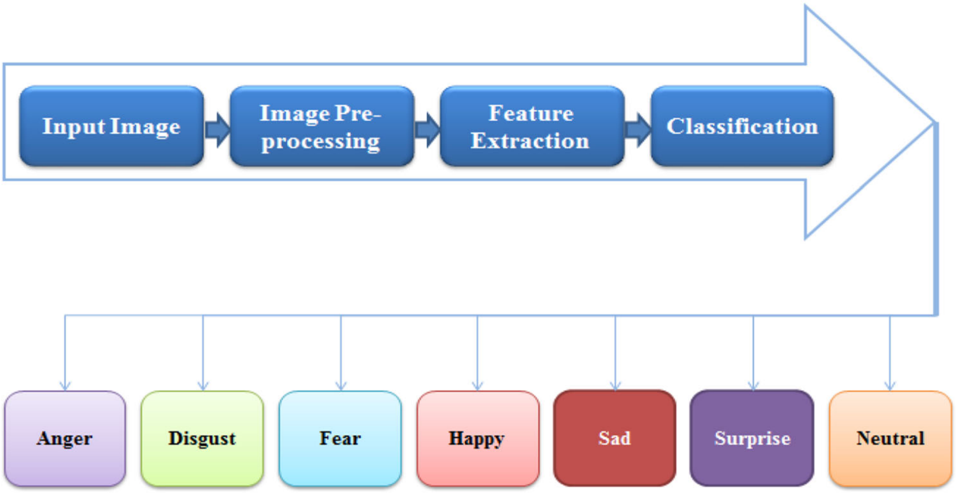


Fig. 1 The process flow of the proposed work

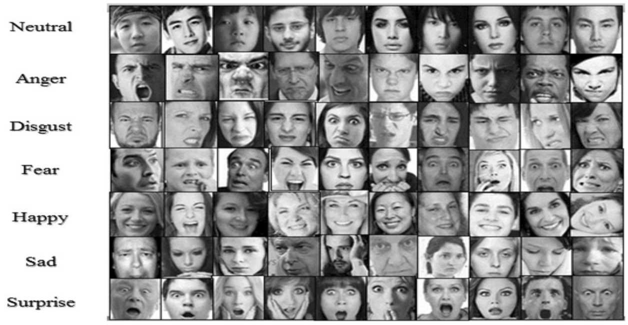


Fig. 2 The sample images of the FER-2013 database

three different subnet-based CNN architectures for identifying facial expressions. Furthermore, these subnets are ensembled to achieve better accuracy of 65% on the FER-2013 dataset. Agarwal et al. [19] proposed two novel configurable CNN architectures using customized parameters such as kernel size and the number of filters to obtain good classification accuracy for the FER-2013 database. Yu and Zhang [20] proposed a facial expression identification method based on static images for Emotion Recognition in the Wild Challenge (EmotiW) 2015. Here, an ensemble of three different deep convolutional neural network methods has been deployed to explore the seven basic expressions. These three CNN models are first pre-trained on the FER-2013 dataset, followed by fine-tuning with the training set of SFEW 2.0. Van Huynh et al. [21] proposed a combined approach that deals with both facial and eye movement information to improve the model's performance. A total of fifty-one features have been extracted to classify facial expressions for Acted Facial Expressions in the Wild (AFEW) dataset. Chiranjeevi et al. [22] proposed a lightweight emotion classification engine as a pre-processor for other conventional supervised learning methods. The performance of the model has been measured based on Key Emotion (KE). Perveen et al. [23] proposed a hybrid approach consists of multiple statistical features (i.e., kurtosis, skewness, entropy, energy, moment, mean, variance, and standard deviation) and facial spatial features to predict the class of expressions for the JAFFE database. Table 1 summarizes the state-of-the-art work concerning the dataset used, the result obtained, and significant findings, such as the strengths and weaknesses of the earlier works.

A study of enucleated literature on facial expression recognition clarifies that some datasets do not include all classes of universal facial expressions. Thus, regardless of overall performance, the classification results of the model are represented based on specific class-wise expressions. Some techniques give good results but are not generalized enough and have an overfitting problem with large datasets. Therefore, we emphasize the overall performance of facial expression recognition for the FER-2013 database. However, other recent facial expression datasets are also

Table 2 Comparison of parameter specification for proposed models

Proposed Model	Parameters						
	Data Augmentation	Learning Rate	Dropout	No. of Epochs	Kernel Initialization	Activation function	Average inference time per epoch
Proposed_Model_1	R-40,HS -0.2, VS- 0.2, SR-0.2, ZR-0.2, HF-True	0.0005	25% (HL) 50% (DL)	65	he_normal	ELU (CL and FCL)	40 s 89 ms (Google colab)
Proposed_Model_2	HF- True	0.0001	25% (DL)	130	-NA-	ReLU (CL and FCL)	41 s 91 ms (Google colab)

R Rotation, *HS* Height_Shift, *WS* Width_Shift, *SR* Shear Range, *ZM* Zoom Range, *HF* Horizontal Flip, *HL* Hidden Layers, *DL* Decision Layer, *CL* Convolution Layer, *DL* Dense Layer, *FCL* Fully Connected Layer

Table 3 Constituent layer

Conv2D ⇒ Nf × K_1 × K_2 ⇒ Padding ⇒ K_Init ⇒ Strides ⇒ BN ⇒ Activation(ELU/ReLU) ⇒ Pooling ⇒ Dropout
Flatten
Dense1 ⇒ Nf × K_1 × K_2 ⇒ Padding ⇒ K_Init ⇒ Strides ⇒ BN ⇒ Activation(ELU/ReLU) ⇒ Pooling ⇒ Dropout
Dense2 ⇒ Nf × K_1 × K_2 ⇒ Padding ⇒ K_Init ⇒ Strides ⇒ BN ⇒ Activation(ELU/ReLU) ⇒ Pooling ⇒ Dropout
SOFTMAX

Nf Num_filters, K_1 Kernel$_1$, K_2 Kernel$_2$, *K_Init* Kernel_Initialization, *BN* Batch Normalization examined to validate the performance of our proposed methods (see Sect. 4).

3 Proposed methodology

We have proposed two customized-CNN architectures best suited for the classification of universal facial expressions. This section includes image preprocessing and CNN-based feature extraction and classification methods. We also discussed the essential measures to reduce the most common problem of overfitting in this section. Figure 1 depicts the complete process flow of the proposed work for facial expression recognition.

3.1 Image Preprocessing

In deep learning-based solution approaches, image preprocessing is preferred as a necessary step before the feature extraction process to achieve high classification accuracy. Here, the image preprocessing is applied to the collected benchmark database (i.e., FER-2013). The FER-2013 [24] database contains grayscale images with a fixed image size of 48 × 48 pixels. Each grayscale image represents a specific class of universal facial expressions such as Anger, Disgust, Fear, Happy, Sad, Surprise, and Neutral. The FER-2013 database includes 35,887 grayscale images, out of which 28,709 are used for training purposes, and the remaining 7178 are used for testing purposes. Sample images from the FER-2013 database for all seven facial emotion categories are depicted in Fig. 2.

Image preprocessing involves two efficient scaling methods, namely normalization, and standardization to improve the quality of images that may have poor contrast due to glare. The normalization method is used to rescale the pixel value of the image data to a predefined range. At the same time, standardization is used to rectify the signal intensity variation that occurs due to unstable lighting problems during the acquisition process. Here, a transformation of a grayscale image into a standard grayscale image is performed. This method is also known as min–max scaling. Our task is to scale the image in such a way that it represents zero mean and unit variance. Next, we implemented the image data augmentation process to enrich the capacity of the sample data by applying various transformation functions such as zooming, shearing, shifting, rotation, and flipping. The details of the parameters for image data augmentation are displayed in Table 2.

3.2 Feature extraction and classification (convolutional neural network)

This manuscript proposed two customized-CNN models named Proposed_Model_1 and Proposed_Model_2 to predict the class of facial expression using the benchmark FER-2013 dataset. The architecture of both our models has the same number of convolution blocks. However, these models differ slightly concerning hyper-parameters (learning rate, activation function) and regularization (dropout) tuning. These factors create a significant difference in the complexity and accuracy of our proposed models. Our proposed CNN models follow an intuitive process to extract low, middle, and high-level features. The initially hidden layers of our models are responsible for extracting low-level and middle-level features. In comparison, higher layers are used to match the input image with the stored images using the pattern (template) matching process. Each layer aims to make the best use of the applied weights and parameters from the preceding layer. The softmax loss/activation function is utilized here for classification. The configuration of the parameters used for both of our proposed models is shown in Table 3.

The layer-wise functionality of our proposed CNN models is described below:

Convolution layers are the primary building blocks that determine how learnable parameters can be used to process information. The convolution layer converts a two-dimensional image into positively correlated weighted neurons. As the learning process progresses more deeply, these weighted neurons learn the most important features or

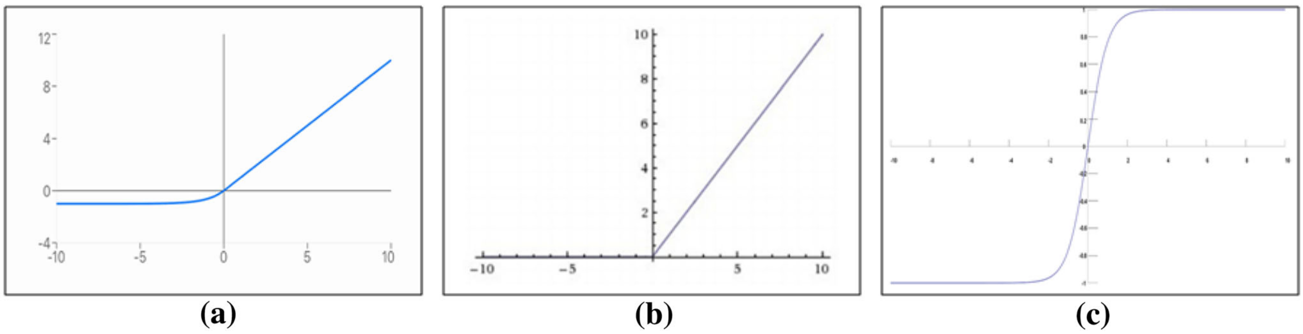


Fig. 3 Activation functions **a** ELU **b** ReLU **c** Softmax

patterns. The size of the kernel determines how the feature map will look after convolution. We implemented four convolution operations (i.e., 64, 128, 512, 512) for both of our proposed models with a kernel size of 3 × 3 (except the second convolution block, which consists of 5 × 5). Here, different kernel sizes are used to generate significant feature maps with spatial correlations between pixels to detect different features or identify unique patterns. The formula for generating a convoluted image from the input and feature vector is shown in (1). The formula for calculating the output height and width is shown in (2) and (3).

$$(A * w)_{i,j} = \sum_{i'=1}^{K}\sum_{j'=1}^{K} A\left(i+i', j+j'\right), w(i', j') \tag{1}$$

$$H_o = \frac{(H_i - k + 2p)}{s} + 1 \tag{2}$$

$$W_o = \frac{(W_i - k + 2p)}{s} + 1 \tag{3}$$

where, H_o, W_o are the output height and width H_i, W_i are the input height and width, p is the padding size, s is the stride value, and k is the kernel.

The activation function introduces non-linearity and also confirms whether a neuron will fire. This manuscript has deployed three activation functions: Exponential Linear Unit (ELU), Rectified Linear Unit (ReLU), and softmax function. Our Proposed_Model_1 has been evaluated with ELU, while the Proposed_Model_2 has been examined with ReLU for the hidden layers. The softmax function is utilized for multi-class classification purposes in the dense layer of both models to predict the expression class of the input image. Rectified Linear Unit (ReLU) is the default function and is the most preferred activation function in CNNs. Here, ReLU converts all negative values to a minimum (i.e., zero) to check the non-linearity of the function while positive values remain unchanged (i.e., original) to represent the linearity, as shown in (4). These activated neurons fire only when a specific threshold is attained.

$$\varnothing(z) = \begin{cases} 0 & if \quad z < 0 \\ z & if \quad z \geq 0 \end{cases} \tag{4}$$

Exponential Linear Unit (ELU), also known as Scaled Exponential Linear Unit (SELU), eliminates the sharp smoothing condition of ReLU for negative inputs. During

Table 4 Proposed architecture of Proposed_Model_1 and Proposed_Model_2

Model Layer	Proposed_Model_1 Size	Proposed_Model_2 Size	Model Layer	Proposed_Model_1 Size	Proposed_Model_2 Size
1Conv2D	64	64	4Kernel_size	3 × 3	3 × 3
1Kernel_size	3 × 3	3 × 3	4Kernel_ Init	he_normal	–
1Kernel_ Init	he_normal	–	4Activation	ELU	ReLU
1Activation	ELU	ReLU	4BN	Yes	Yes
1BN	Yes	Yes	4Max Pooling	(2, 2)	(2, 2)
1Max Pooling	(2, 2)	(2, 2)	4Dropout	25%	–
1Dropout	25%	–	Flatten()	–	–
2Conv2D	128	128	1Dense	256	256
2Kernel_size	5 × 5	5 × 5	5Kernel_size	3 × 3	3 × 3
2Kernel_ Init	he_normal	–	5Kernel_ Init	he_normal	–
2Activation	ELU	ReLU	5Activation	ELU	ReLU
2BN	Yes	Yes	5BN	Yes	Yes
2Max Pooling	(2, 2)	(2, 2)	5Dropout	50%	25%
2Dropout	25%	–	2Dense	512	512
3Conv2D	512	512	6Kernel_size	3 × 3	3 × 3
3Kernel_size	3 × 3	3 × 3	6Kernel_ Init	he_normal	–
3Kernel_ Init	he_normal	–	6Activation	ELU	ReLU
3Activation	ELU	ReLU	6BN	Yes	Yes
3BN	Yes	Yes	6Dropout	50%	25%
3Max Pooling	(2, 2)	(2, 2)	Classification	7	7
3Dropout	25%	–	7Kernel_size	3 × 3	3 × 3
4Conv2D	512	512	7Activation	Softmax	Softmax

K_Init Kernel_Initialization, *BN* Batch Normalization, *ELU* Exponential Linear Unit, *ReLU* Rectified Linear Unit the training process, we observed the dying ReLU problem, where large gradients flow down. Therefore, the output of the ReLU activation function remains constant to zero value for all input data samples, indicating that it is not contributing to the model. Thus we applied the ELU activation function to fix this problem. ELU is considered to be a perfect combination of ReLU and Leaky ReLU, which provides a slight improvement in the results. The function for ELU is shown in (5).

$$\varnothing(z) = \begin{cases} \alpha(e^{z} - 1) & if \quad z < 0 \\ z \quad if & z \geq 0 \end{cases} \tag{5}$$

We applied the probability-based activation function (i.e., Softmax) to classify the category (i.e., class) of basic facial expressions in both of our proposed models. The formula used to calculate probability using Softmax is displayed in (6). Figure 3 depicts a graphical representation of all the activation functions used in our experiment.

$$\sigma(\overrightarrow{Z})_i = \frac{e^{Z_i}}{\sum_{j=1}^{K} e^{Z_j}} \tag{6}$$

Kernel initialization is responsible for managing the weights to obtain a more accurate recognition rate. We found in the CNNs literature that random weights produced poor results. Thus, we applied the ‘he_normal’ kernel initialization method on Proposed_Model_1, which resulted in slightly better accuracy over a relatively reduced number of epochs.

The Pooling layer is aimed to downsample the feature space caused by the non-linearity involved in the network while preserving important information. Thus, the complexity of the model and the computation time required to accomplish the task is reduced, indicating improved accuracy. We implemented a max-pooling function of size (2 × 2) for our use case experiments.

A Fully connected layer is responsible for extracting high-level features through complex network architectures. Here, the neuron of one layer is connected to all subsequent layers for both the feed-forward and the backpropagation to improve the learning weights significantly.

3.3 Overfitting and measures to reduce overfitting

Overfitting problem is most common when dealing with machine learning and deep learning approaches, leading to less efficient results. Overfitting occurs when the training algorithm memorizes the training data due to low weights or sometimes over-tuning of the weights. Thus, the obtained results are good enough for training data but reflect poor performance on the unseen data. To reduce the impact of overfitting, we have adopted the following protocols:

- Use of a smaller CNN architecture with fewer weights to restrict model memorization.
- We implemented fine-tuning of hyperparameter through callback functions, dropout, and data augmentation.

Smaller CNN architecture is always a good choice to reduce the problem of overfitting. We emphasized this concept and proposed a simple CNN model with two different variants, including distinct parameters specifications. Table 4 demonstrates the layer-wise description of our proposed model.

Regularization and hyper-parameters tuning provide an alternate way to reduce the effects of overfitting. We efficiently utilized hyper-parameters and regularization such as activation function, dropout, early stopping, data augmentation, facilitated learning rate, and kernel initializer. Dropout is used to drop a random number of nodes from the hidden and dense (i.e., visible) layers in the forward and backward propagation. We applied 25% dropout in hidden layers and 50% dropout in dense layers for Proposed_Model_1. In contrast, Proposed_Model_2 contains 25% dropout in the dense layers only. We found in the literature that dropout is the best choice to reduce

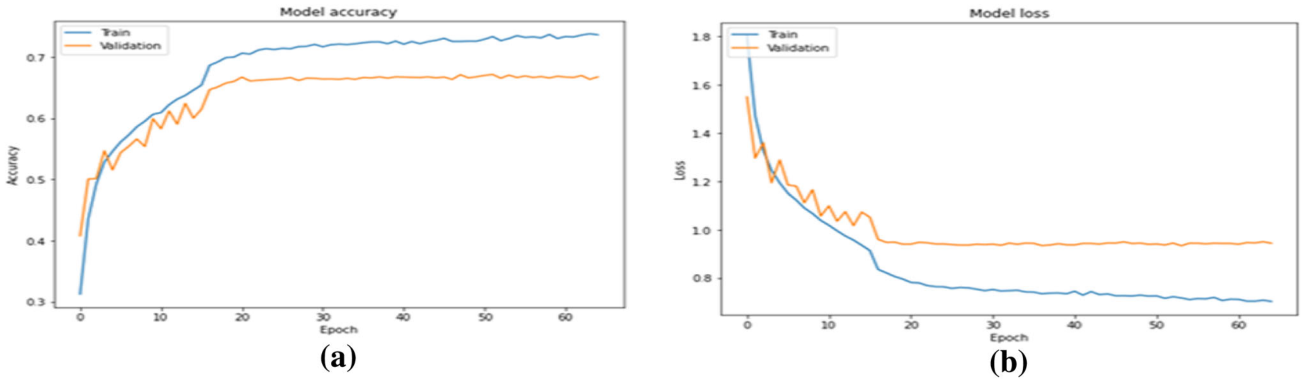


Fig. 4 Accuracy and loss curve with respect to epochs for Proposed_Model_1 **a** Accuracy **b** Loss

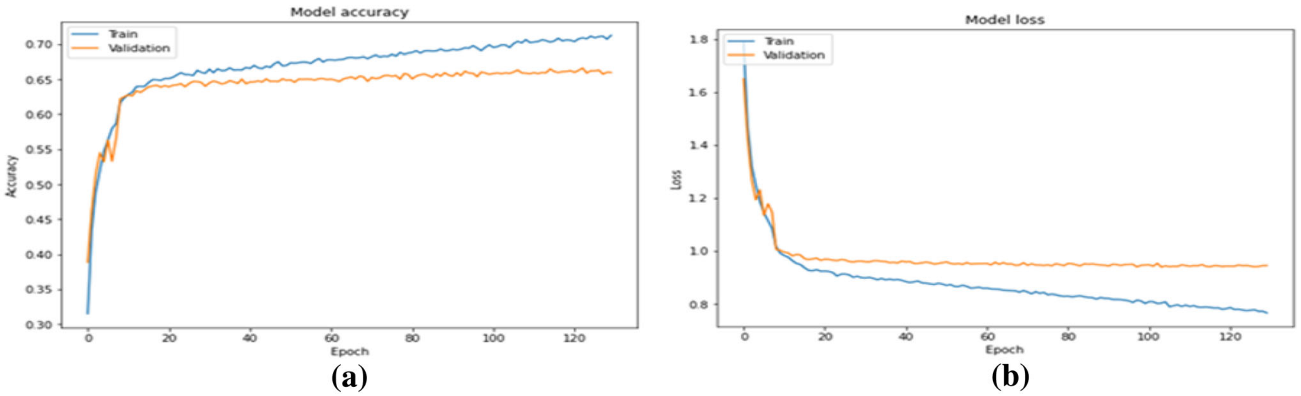
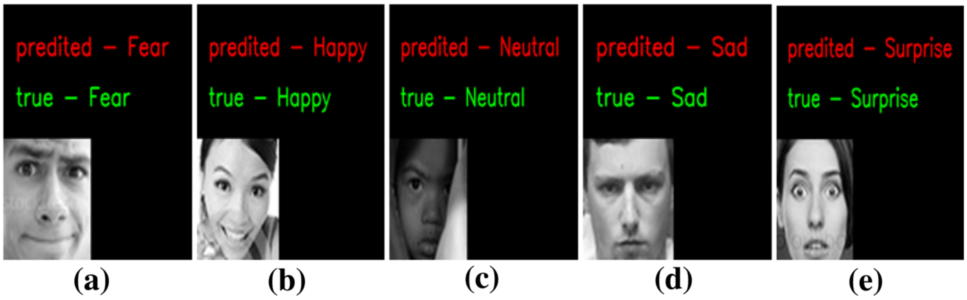


Fig. 5 Accuracy and loss curve with respect to epochs for Proposed_Model_2 **a** Accuracy **b** Loss

Fig. 6 Correct classification examples (**a**–**e**)



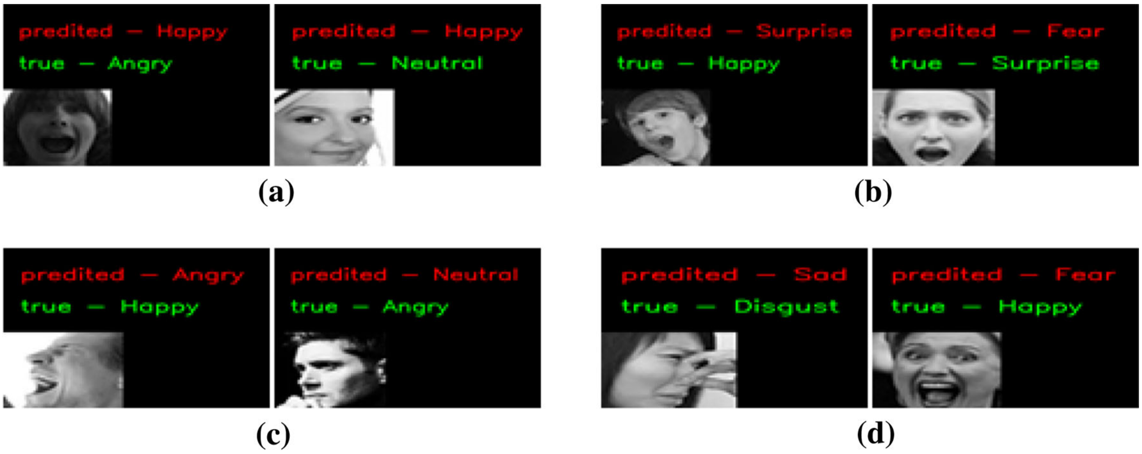


Fig. 7 Incorrect classification examples row **a**–**c** interrelated expressions, **d** other cases

overfitting through creating a strong bonding among neurons to learn robust and meaningful features. Dropout increases the convergence time during the training process. The term epoch refers to a pass through the entire training dataset. After each epoch, we have some sophisticated sets of weights to make the weights even better. Early stopping is call back function from the Keras library, which allows us to stop the training at any stage of the process, when we do not receive further upgrades in validation accuracy or a reduction in validation loss after particular predefined epochs (i.e., patience value). Reducing the learning rate to a plateau is another callback function that automatically decreases the learning rate after a specific predefined value based on the relative improvement in model's performance.

The network architecture for Proposed_Model_1 and Proposed_Model_2 is identical except for some additional fine-tuned parameters, applied separately on Proposed_Model_1 to investigate the effect of these parameters to achieve better accuracy. The detailed specifications for both of our proposed models are displayed in Table 4.

Table 5 Comparison of various state-of-the-art datasets used for facial expression recognition

Dataset	Year	Sample Images	Type	Color Channel	Elicitation Method	Expression Distribution	Annotation Method	Access Link	Ref
AffectNet	2018	450,000 (I)	MSE	RGB	P & S	07 (U)	01 annotator per image	http://mohammadmahoor.com/databases-codes/	[27]
RAF-DB	2017	29,672 (CS)	Crowd-sourced	RGB	P & S	07 (U) 12 (C)	40 labellers per image	http://www.whdeng.cn/RAF/model1.html	[26]
4DFAB	2012–17	1.8 million 3D faces (79 LMs)	HR 3D face	3D-RGB	P & S	06 (U)	Five year span for data collection	http://www.di3d.com	[29]
EmotionNet	2016	1,000,000 (I)	RT Image	RGB	P & S	06 (U) 17 (C)	10% (M) and 90% (A)	http://cbcsl.ece.ohio-state.edu/dbformemotionet.html	[25]
ExpW	2015	91,793	OWI	RGB	P & S	07 (U)	100% (M)	http://mmlab.ie.cuhk.edu.hk/projects/socialrelation/index.html	[28]
FER-2013	2013	35,887 images	RT Image	Grayscale (48 × 48)	S	07 (U)	Image search API	https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge	[24]

P Pose, *LMs* Landmarks, *S* Spontaneous, *M* Manual, *A* Automatically, *U* Universal, *C* Compound, *CS* Crowd-Sourced, *I* Internet, *MSE* Multi-Search Engine, *RT* Real-Time, *HR* High- Resolution, *3D* Three-dimensional, *OWI* Original Web Image

Table 6 Comparison of proposed models with state-of-the-art methods

Reference	Baseline architecture	Highest accuracy	Expression classes covered	Parameters	Year
Saeed S et al.[30]	HOG and Support Vector Machine (cubic kernel)	57.17%	Universal	0.37M	2018
Talegaonkar et al.[31]	CNN	60.13%	Universal	5.0M	2019
Liu (Single subnet best) [18]	VGGNet	62.44%	Universal	84M	2016
Liu (Ensemble subnet) [18]	VGGNet	65.30%	Universal	84M	2016
Agrawal et al. [19] (2)	CNN	65.23%	Universal	0.46M	2020
Wan et al. [17]	AlexNet, VGGNet, 8-layer-net, 11-layer-net	65.34%	Universal	14M	2016
Agrawal et. al. [19] (1)	CNN	65.77%	Universal	0.93M	2020
Proposed_Model_2	Customized- CNN	**66.61%**	Universal	4.4M	2021
Proposed_Model_1	Customized- CNN	**67.24%**	Universal	4.4M	2021

(1) Model 1 of reference, (2) Model 2 of reference

4 Experimentation and results analysis

In this section, we discussed the experimental setup requirement. Subsequently, the outcomes of these experiments are evaluated and comparatively analyzed for all of our proposed models.

4.1 Experimental setup

Experiments have been performed on an interactive python notebook (i.e., Google Colaboratory) with free GPU access support on a cloud-based environment. The Colab environment consists of open-source services of NVIDIA-SMI 460.67, Driver Version—460.32.03. The access 'Tesla K80' GPU device with CUDA version 11.2 has been

Table 7 Comparison of model's performance on various state-of-the-art datasets

Dataset	Training Samples (80%)	Testing Samples (20%)	Random Images Taken From Test Samples	Average Model Accuracy	
				Proposed_Model_1	Proposed_Model_2
AffectNet [27]	3,60,000	90,000	70 (07 expressions)	59.40%	56.80%
RAF-DB [26]	23,738	5934	70 (07 expressions)	66.00%	62.5%
EmotionNet [25]	800,000	200,000	60 (06 expressions)	66.12%	63.36%
ExpW [28]	73,434	18,359	70 (07 expressions)	61.80%	65.30%
FER-2013 [24]	28,709	7178	70 (07 expressions)	67.24%	66.61%

considered for faster processing of datasets during training. The programming is done in Python 3.7.10 version with Tensorflow 2.4.1 and Keras 2.4.3. Our proposed model can take inputs in the form of an image or video and generate the output simultaneously.

4.2 Visualization of results

The trade-off between the validation accuracy and training accuracy and validation loss and training loss for Proposed_Model_1 and Proposed_Model_2 are depicted in Figs. 4 and 5, respectively. The results clearly show that Proposed_Model_1 achieves slightly better accuracy in fewer epochs. It is also observed that sometimes validation accuracy exceeds the training accuracy due to the dropout applied to this network. Dropout eliminates random nodes to participate in the training process, creating a stronger connection between neurons to learn more meaningful patterns/features without memorizing the weights.

We obtained a validation accuracy of 67.24% for Proposed_Model_1 and 66.61% for Proposed_Model_2. Both of our proposed models exhibited slightly better accuracy than humans (i.e., ~ 65%) and other existing methods for the FER-2013 database.

Figure 6 indicates some results for different facial expressions with a correct prediction. In contrast, Fig. 7 represents the incorrect classification cases, including similar expressions.

Our proposed model fails to differentiate between similar expressions such as Angry-Happy, Happy-Neutral, Surprise-Happy, Fear-Angry, Neutral-Angry, Sad-Disgust, and vice-versa. There are two critical reasons for incorrect classifications: the unavailability of a sufficient number of training samples. The second is that facial muscle motions (behavioral changes) may be the same for another expression. More specifically, the facial response to certain universal expressions may be the same for another expression for a different person. During the experiments, it was observed that the kernel initialization with parameter 'he_normal' provides remarkable accuracy over a small number of epochs. However, this does not directly affect the accuracy of the model. The slight improvement in the accuracy of our Proposed_Model_1 is due to the effect of regularization and fine-tuning of parameters. Since there is no mechanism or a predefined method to select the best hyperparameters, thus we performed several experiments iteratively and observed the impact on the results relatively. Only then do we finalize the hyperparameters with adequate control for our use case.

4.3 Performance comparison

We considered some other state-of-the-art datasets for facial expression recognition such as EmotionNet [25], RAF-DB [26], AffectNet [27], ExpW [28], and 4DFAB [29] to prove the effectiveness of our use case dataset, i.e., FER-2013. The comparative analysis of these datasets is shown in Table 5.

The above table pointed out that the FER-2013 dataset is more promising for real-time applications, where small computational power-based devices with less storage space are implanted to get a quick response. For instance, in a health (cerebral palsy) monitoring system, rapid response/assistance can be provided based on the patient's facial expressions. Thus, the FER-2013 dataset, consisting of small-sized (48 × 48) grayscale images representing all universal spontaneous facial expressions, can be easily deployed in these situations. Two other factors that make the FER-2013 superior to other recent datasets are; spontaneous expressions (natural emotions) and image search API-based annotation method (automated). These two factors can significantly increase the recognition rate and efficiency of the model. In contrast, other recent datasets containing massive high-resolution images and videos can take a long time to process the image, resulting in poor performance.

To further inspect the efficiency of the suggested model, it was compared with other contemporary methods implemented on the same benchmark FER-2013 dataset to classify all seven universal classes of facial expressions. In this paper, the classification accuracy is confirmed as the base parameter of comparison. Table 6 represents a comparison of our Proposed_Model_1 and Proposed_Model_2 with several other state-of-the-art methods, including CNN-based models. The comparison indicates that both of our proposed models outperform the existing models concerning accuracy. It is visible that both of our proposed models, i.e., Proposed_Model_1 and Proposed_Model_2 have less complexity in their structure and require fewer training parameters. However, both of our proposed models require slightly higher parameter space than one of the comparison models [19].

Here, we also examined the performance of our proposed models on various recent datasets to validate our results, as shown in Table 7. The randomly selected images from a test set of the recent datasets are taken into account to evaluate the performance of our proposed models considering universal facial expressions.

The results reveal that our proposed methods with the FER-2013 dataset outperform the other existing datasets.

5 Conclusion

With the impetus to develop reliable, efficient, and robust facial expression recognition for real-time applications, we proposed two new customized-CNNs-based models named Proposed_Model_1 and Proposed_Model_2 to classify seven different categories of universal facial expressions. The experiments have been conducted on the benchmark public FER-2013 dataset. The size of the image samples has been kept constant and fixed to its original size, i.e., 48 × 48. The Proposed_Model_1 and the Proposed_Model_2 include simple deep network architectures; the only difference between these two models is the variations of different hyper-parameters and regularization methods to accelerate the overall performance of the model. It has been observed from the state-of-the-art approaches that overfitting is a significant cause of poor recognition rate in any model. Also, we found that regularization techniques such as data augmentation, dropout, early stopping, and reducing learning rate may effectively improve the performance of the model if it is tuned and controlled adequately. Thus, we conducted several experiments to finalize the values of the hyperparameters. The more efficient activation function ELU is utilized in our Proposed_Model_1 to overcome the dying ReLU condition. We obtained an accuracy of 67.24% for Proposed_Model_1, and 66.61% for Proposed_Model_2. Our proposed approaches are ahead in terms of accuracy obtained as compared to other existing methods. It slightly surpasses human-like accuracy to recognize facial expressions. In the near future, we will design a new algorithm for classifying of micro-expressions, such as eye-blinking, with improved accuracy.

References

1. Borovac B, Gnjatović M, Savić S, Raković M, Nikolić M (2016) Human-like robot marko in the rehabilitation of children with cerebral palsy. New trends in medical and service robots. Springer, pp 191–203
2. Kant Verma K, Singh BM, Dixit A (2019) A review of supervised and unsupervised machine learning techniques for suspicious behavior recognition in intelligent surveillance system. Int J Inf Technol. https://doi.org/10.1007/s41870-019-00364-0
3. Ansari MA, Singh DK (2021) Monitoring social distancing through human detection for preventing/reducing COVID spread. Int J Inf Technol 13:1255–1264. https://doi.org/10.1007/s41870-021-00658-2
4. Van Thang, D., Mangla, M., Satpathy, S., Pattnaik, C. R., & Mohanty SN (2021) A fuzzy-based expert system to analyse purchase behaviour under uncertain environment. Int J Inf Technol 13:997–1004. https://doi.org/10.1007/s41870-021-00615-z
5. Choi IY, Oh MG, Kim JK, Ryu YU (2016) Collaborative filtering with facial expressions for online video recommendation. Int J Inf Manage 36:397–402
6. Singh DK (2015) Recognizing hand gestures for human computer interaction. In: In 2015 International Conference on Communications and Signal Processing (ICCSP). IEEE, 0379–0382
7. Ansari MA, Singh DK (2019) An approach for human machine interaction using dynamic hand gesture recognition. In: In 2019 IEEE Conference on Information and Communication Technology. IEEE, 1–6
8. Rahman A, Mohammed M, Beg S (2018) Face sketch recognition: an application of Z-numbers. Int J Inf Technol 11:541. https://doi.org/10.1007/s41870-018-0178-0
9. Shah R, Lewis M (2003) Locating the neutral expression in the facial-emotion space. Vis cogn 10:549–566
10. Sharan P, Sandhya KV, Barya R, Bansal M, Upadhyaya AM (2021) Design and analysis of moems based displacement sensor for detection of muscle activity in human body. Int J Inf Technol 13:397–402
11. Joormann J, Gotlib IH (2006) Is this happiness I see? Biases in the identification of emotional facial expressions in depression and social phobia. J Abnorm Psychol 115:705
12. Vijayalakshmi M, Peter VJ (2021) CNN based approach for identifying banana species from fruits. Int J Inf Technol 13:27–32
13. Gupta M, Calvin MR, Desai B, Bhoir MS (2021) Chest Disease Detection through X-Ray using Machine Learning. Int J Inf Technol 7:46–49
14. Sharma R, Pachori RB, Sircar P (2020) Automated emotion recognition based on higher order statistics and deep learning algorithm. Biomed Signal Process Control 58:101867
15. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15:1929–1958
16. Saeed S, Mahmood MK, Khan YD (2018) An exposition of facial expression recognition techniques. Neural Comput Appl 29:425–443
17. Wan W, Yang C, Li Y (2016) Facial Expression Recognition Using Convolutional Neural Network. A Case Study of the

Relationship Between Dataset Characteristics and Network Performance. Stanford University Reports Stanford, Stanford

18. Liu K, Zhang M, Pan Z (2016) Facial expression recognition with CNN ensemble. In: 2016 international conference on cyberworlds (CW), 163–166
19. Agrawal A, Mittal N (2020) Using CNN for facial expression recognition: a study of the effects of kernel size and number of filters on accuracy. Vis Comput 36:405–412
20. Yu Z, Zhang C (2015) Image based static facial expression recognition with multiple deep network learning. In: In Proceedings of the 2015 ACM on international conference on multimodal interaction, 435–442
21. Van Huynh T, Yang H-J, Lee G-S, Kim S-H, Na I-S (2019) Emotion recognition by integrating eye movement analysis and facial expression model. In: Proceedings of the 3rd International Conference on Machine Learning and Soft Computing. pp 166–169
22. Chiranjeevi P, Gopalakrishnan V, Moogi P (2015) Neutral face classification using personalized appearance models for fast and robust emotion detection. IEEE Trans Image Process 24:2701–2711
23. Perveen N, Guptaand S, Verma K (2012) Facial expression classification using statistical, spatial features and neural network. Int J Adv Eng Technol 4:424
24. Mollahosseini A, Chan D, Mahoor MH (2016) Going deeper in facial expression recognition using deep neural networks. IEEE Winter Conf Appl Comput Vision. https://doi.org/10.1109/WACV.2016.7477450
25. Benitez-Quiroz CF, Srinivasan R, Martinez AM (2016) EmotioNet: An accurate, real-time algorithm for the automatic annotation of a million facial expressions in the wild. Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit. https://doi.org/10.1109/CVPR.2016.600
26. Li S, Deng W (2019) Reliable crowdsourcing and deep locality-preserving learning for unconstrained facial expression recognition. IEEE Trans Image Process 28:356–370. https://doi.org/10.1109/TIP.2018.2868382
27. Mollahosseini A, Hasani B, Mahoor MH (2019) AffectNet: a database for facial expression, valence, and arousal computing in the wild. IEEE Trans Affect Comput 10:18–31. https://doi.org/10.1109/TAFFC.2017.2740923
28. Zhang Z, Luo P, Loy CC, Tang X (2018) From facial expression recognition to interpersonal relation prediction. Int J Comput Vis 126:550–569. https://doi.org/10.1007/s11263-017-1055-1
29. Cheng S, Kotsia I, Pantic M, Zafeiriou S (2018) 4DFAB: a large scale 4D database for facial expression analysis and biometric applications. Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit. https://doi.org/10.1109/CVPR.2018.00537
30. Saeed S, Baber J, Bakhtyar M, Ullah I, Sheikh N, Dad I, Sanjrani AA (2018) Empirical evaluation of svm for facial expression recognition. Int J Adv Comput Sci Appl 9:670–673
31. Talegaonkar I, Joshi K, Valunj S, Kohok R, Kulkarni A (2019) Real time facial expression recognition using deep learning. In: In Proceedings of International Conference on Communication and Information Processing (ICCIP)