International Journal of Pattern Recognition
and Artificial Intelligence
Vol. 37, No. 8 (2023) 2356008 (27 pages)

DOI: 10.1142/S0218001423560086

World Scientific
www.worldscientific.com

Emotion Recognition from Facial Expression Using Hybrid CNN–LSTM Network

M. Mohana*, P. Subashini† and M. Krishnaveni‡

Department of Computer Science
Centre for Machine Learning and Intelligence (CMLI)
Avinashilingam Institute (Deemed University)
Coimbatore, India
**mohana_cs@avinuty.ac.in*
†*subashini_cs@avinuty.ac.in*
‡*krishnaveni_cs@avinuty.ac.in*

Received 16 September 2022
Accepted 29 January 2023
Published 7 July 2023

Facial Expression Recognition (FER) is a prominent research area in Computer Vision and Artificial Intelligence that has been playing a crucial role in human–computer interaction. The existing FER system focuses on spatial features for identifying the emotion, which suffers when recognizing emotions from a dynamic sequence of facial expressions in real time. Deep learning techniques based on the fusion of convolutional neural networks (CNN) and long short-term memory (LSTM) are presented in this paper for recognizing emotion and identifying the relationship between the sequence of facial expressions. In this approach, a hyperparameter tweaked VGG-19 skeleton is employed to extract the spatial features automatically from a sequence of images, which avoids the shortcoming of the conventional feature extraction methods. Second, these features are given into bidirectional LSTM (Bi-LSTM) for extracting spatiotemporal features of time series in two directions, which recognize emotion from a sequence of expressions. The proposed method's performance is evaluated using the CK+ benchmark as well as an in-house dataset captured from the designed IoT kit. Finally, this approach has been verified through hold-out cross-validation techniques. The proposed techniques show an accuracy of 0.92% on CK+, and 0.84% on the in-house dataset. The experimental results reveal that the proposed method outperforms compared to baseline methods and state-of-the-art approaches. Furthermore, precision, recall, $F1$-score, and ROC curve metrics have been used to evaluate the performance of the proposed system.

Keywords: Artificial intelligence; Bi-LSTM; computer vision; CNN–LSTM; facial expression recognition; human–computer interaction; spatiotemporal.

1. Introduction

Emotion recognition plays a prominent role in human–computer interaction. Humans can express emotions in different ways such as, through speech, facial

*,† Corresponding authors.

expressions, and body language. Facial expression analysis is the most significant and active research area in recent decades among those concerned with emotion recognition. Mehrabian's[29] works show that 55% of the message can be obtained through feelings and attitudes in facial expressions, 7% is expressed through speech, and the rest by paralinguistic. Ekman and Friesen[11,12] proposed six basic emotions: happy, sad, surprise, fear, anger, and disgust. Sometimes researchers add neutral and contempt in this category. These are known as universal expressions or primary emotions. In the fields of computer vision, deep learning, and pattern recognition, the study of facial expression recognition (FER) is given more consideration. It has also been extensively used in a variety of fields, including education, human–robot interaction, the healthcare system, autism spectrum disorder, and sophisticated driver assistance systems.

Researchers have been working on FER using machine learning algorithms for the past two decades. The objective of FER is to distinguish and categorize the significant movements of various facial musculature into meaningful diverse emotions. Numerous research on human–computer interaction has been conducted.[27] However, the FER system comprises four main steps: face detection, pre-processing, feature extraction, and emotion classification. First, face detection is a serious processing step to identify or locate whether the images/video frames contain a face or not. Second, pre-processing methods are utilized to highlight the features of the image. The third stage involves extracting appropriate features from the detected face to identify the considerable emotion from facial expressions. Finally, the classifier has been trained to categorize emotions based on the target label.

Conventional FER commonly adopts various handcrafted feature extraction techniques (e.g. HOG, SIFT, LBP) to extract the feature from facial images.[14,37,51] Later, geometric-based and appearance-based feature extraction approaches were used to automate the FER system. According to certain studies,[2,26,30] the geometric-based feature extraction approach retrieves geometric characteristics connected to face action units. Both the active appearance model (AAM) and the active shape model (ASM) extract geometric features depending on the form and location of the facial expression.[36] The appearance technique is more resistant to noise and feature extraction than geometric-based methods. In a few situations, hybrid-based feature extraction approaches were applied, resulting in higher detection performance. These approaches are appropriate only in a clinical situation.

In real time, detecting emotion on the face has several challenges such as complex backgrounds, occlusion, illumination, and spontaneous expression.[26] As people vary by culture, this spontaneous expression differs from the normal expression by subtle expression. In such instances, typical feature extraction algorithms cause significant computing costs, learning time, and poor real-time performance. Furthermore, the complex image necessitates strong memory power and the investigation of discriminative visual features to distinguish facial emotions and connect with a person's related emotional state.

Deep learning has gained prominence in the computer vision sector in recent years because of increased computational power such as graphics processing unit (GPU) and tensor processing unit (TPU) support for massive data training. It showed significant improvement in image recognition and detection. Mainly, the convolutional neural network (CNN)[32] has attained remarkable achievements in the FER system due to its strong expressive power for the feature extraction method. It is widely used in static images which extract features based on the appearance of the images. However, these features could not define emotion entirely because static images lacking in dynamic sequences related to facial expressions.[1] Many two-dimensional (2D) CNN models fail to recognize temporal features in images. As a result, numerous studies integrated both feature extraction approaches, such as CNN with long short-term memory (LSTM)[17] and CNN–recurrent neural network (RNN).[7,19] These techniques are used to extract the dynamic sequence of features from images.

Inspired by different CNN–LSTM-based algorithms, this paper presents fusion feature extraction techniques from a sequence of facial expression images. This study proposes (1) A CNN–LSTM fusion model for the analysis of the sequence of facial expressions, (2) Data augmentation techniques used for generating various illumination, noise, and different angles of the facial image for improving the emotion recognition power of the system in the real-time scenario, (3) A hyperparameter tweaked skeleton of VGG-19 used for extracting spatial feature, which overcomes the shortcoming of conventional feature method, (4) The designed CNN–LSTM-based Bi-LSTM method is used to extract spatiotemporal features which classify each emotion accuracy based on feature vector sequence, and (5) The proposed system's performance is compared to the benchmark dataset and state-of-the-art methods.

This paper is presented as follows: Section 2 describes the related work regarding deep learning techniques for FER systems. Section 3 explains the dataset collection and pre-processing methods. Section 4 describes the proposed CNN with the LSTM network. Section 5 explains the effectiveness of the proposed approach. Section 6 presents the discussions and limitations of this study. Finally, Sec. 7 presents the efficiency of the proposed method.

2. Background and Related Work

2.1. *Facial expression recognition*

The majority of previous research looked at facial expressions from static images, which provide spatial information based on appearance. Facial expressions, on the other hand, are produced by the relaxation and contraction of the facial musculature. Therefore, it is efficient for extracting both spatial and temporal features in facial expressions. This paper analyzed some existing works based on FER using CNN and its limitations.

2.2. *Deep neural network*

In order to address the difficulties in facial expression detection, many researchers have proposed various deep learning techniques based on CNN and the integration of two or three networks in recent decades. The representation of CNN's architecture and the use of a specialized temporal feature extraction approach while processing visual sequences are the key differences between each network.

Mehendale[28] proposed new techniques for facial emotion detection by combining the two-CNN network. First, CNN removed the background of the image. The second one concentrated on facial descriptor vector extraction which works based on different emotions and achieved 96% accuracy on CMU and NIST datasets. Zadeh *et al.* proposed deep CNN with Gabor feature extraction techniques.[53] It increased the speed and accuracy of the FER system. Chowdary *et al.* used transfer learning ResNet-50,[16] VGG-19,[43] Inception V3,[46] and MobileNet to compare performance with the CK+ dataset.[8] Usually, transfer learning achieves better performance on popular benchmark datasets over state-of-the-art techniques. Haddad *et al.* introduced the three-dimensional (3D)-CNN network over a sequence of frames.[15] This proposed method improved the results when tested with CK+ and Oulu-CASIA datasets. A few studies used 3D-CNN in facial emotion recognition. A hybrid CNN–RNN network was suggested by Bai and Goecke and used to retrieve spatiotemporal information from video sequences.[3] Also, this method combines transfer learning ResNet-50[16] for training and used the VGGFace2[54] dataset to enhance the usefulness of the suggested approach. The Grad-CAM heat maps visualization has been used here to visualize the before and after training samples. Sepas-Moghaddam *et al.* proposed VGG-16 with a Bi-LSTM network for extracting spatiotemporal features.[39] The spatial descriptor from image sequences is first extracted by the VGG-16. Then, spatial–angular features are learned using the Bi-LSTM RNN. This method experimented with the IST-EURECOM Light Field Face database. A temporal relational network (TRN) was proposed by Pise *et al.* for recognizing changes in emotions on a student's face in an online learning environment.[34] In addition, MLP has been used as a base classifier for emotion recognition. The proposed framework achieved better results on DISFA+ datasets. Jaiswal and Valstar designed an e-learning system using the CNN network. This method has been mainly proposed for with and without learning disabilities students.[20] It achieved better results in CK+ and JAFFE datasets. In order to jointly learn spatial features and temporal features for FER, Liang *et al.* presented a Bi-LSTM network.[23] This method combined deep spatial network (DSN) and deep temporal network (DTN) for extracting spatial features from image sequences. After that those features are combined and given into the Bi-LSTM network. This combined method achieved better performance in CK+, Oulu-CASIA, and MMI datasets. The CNN–LSTM network was introduced by Li *et al.* to extract spatial–temporal features.[22] Finally, transfer learning has been used to boost the performance of the suggested approach.

Bargal *et al.* proposed CNN with support vector machine (SVM) for emotion classification from a video sequence.[4] After a fully connected layer, SVM was associated to classify the emotions. Xiao *et al.* proposed a malware classification framework based on a CNN–SVM network which is employed to extract the features automatically.[50] Finally, the SVM classifier is used to classify malware according to features with an accuracy of 0.997%. Ruiz-Garcia *et al.* have presented CNN–SVM facial emotion recognition framework for socially assistive robots.[38] This approach combines self-supervised feature extraction methods and SVM for emotion classification. The author achieved 96.26% of accuracy in KDEF datasets. In addition, feature extraction effectiveness was compared using CNN and Gabor filters. Donahue *et al.* presented a long-term RNN to learn spatiotemporal features from long video sequences for activity recognition.[10] Fan *et al.* proposed a hybrid network of RNNs with C3D.[13] First, RNN extracts the features of the image sequence based on appearance while the 3D convolution network combines features map of both image and audio for emotion classification. Jaiswal and Valstar proposed a combination of CNN with Bi-LSTM to learn spatiotemporal features from an image sequence.[20] This approach achieved a great performance in FERA 2015 dataset. Zahara *et al.* proposed an IoT-based facial emotion recognition using a CNN with the help of the OpenCV library.[54] It was mainly proposed to detect micro-expression on the face in real-time facial expression. This method has used FER-2013 for training a neural network and achieved 65.97%.

In this study, CNN is employed to extract spatial information and is trained on labeled facial image sequences. The extracted features are then fed into the designed Bi-LSTM network, which captures the facial expression's temporal and contextual information. Finally, each emotion is classified by the softmax layer. Furthermore, the performance is compared to benchmark datasets and state-of-the-art methods.

3. Methods and Materials

The entire proposed system of numerous stages of the FER is depicted in Fig. 1. The pre-processing pipeline runs on raw video frames of the facial images. In the pre-processing pipelines, histogram equalization, bilateral filtering, flipping, rotating, and normalization are performed for enhancing features and increasing dataset size. The pre-processed dataset consists of three parts: training validation and testing, which consists of different samples. Following that, the CNN–LSTM (Bi-LSTM) network is trained and employed to extract spatiotemporal features from the dynamic sequence for classifying facial expressions. Furthermore, hold-out cross-validation techniques are used to compute the model's accuracy and loss for each epoch. The performance of the proposed approach is measured using the following metrics: confusion matrix, accuracy, precision, recall, and $F1$-score. In addition, the proposed model is evaluated by state-of-the-art techniques.

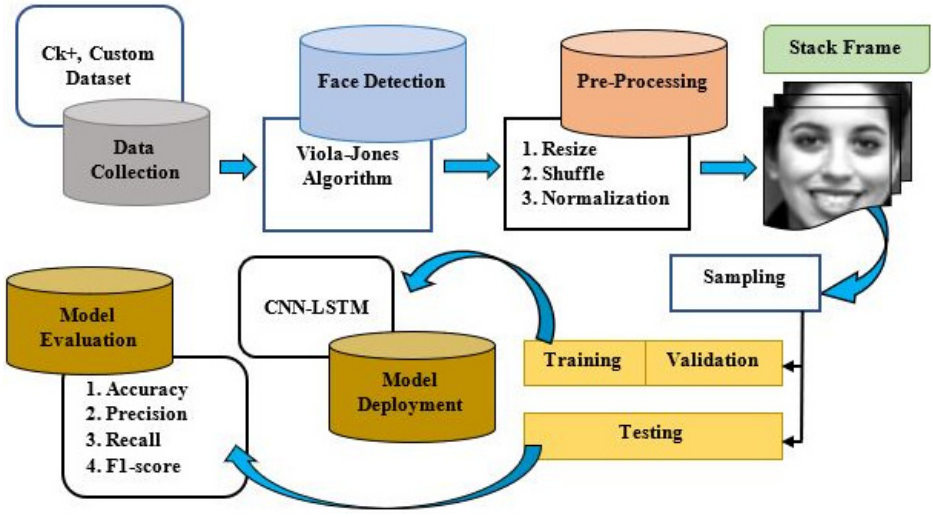


Fig. 1. Overall design of the proposed CNN–LSTM FER system.

3.1. *Dataset collection methods and description*

3.1.1. *CK+ dataset*

A facial expression dataset is a set of static images and short video clips with a range of emotions. The CK+ dataset[25] has 593 image sequences from 123 different people. Out of 593 images, 327 have been labeled for seven facial expressions such as anger, happy, sad, surprise, fear, contempt, and disgust. Figure 2 shows the sample facial expression of the CK+ dataset. Each expression in this database begins with a neutral expression and ends with a peak expression. For this study, the last three frames have been selected for incorporating the spatiotemporal features. Out of seven

Fig. 2. Sample CK+ facial expression images[26]: (a) surprise, (b) disgust, (c) happy.

emotions, only five emotions are considered for this study. The expression of contempt and disgust is less for training and testing. For training, 80% of data has been taken and the remaining 20% has been used for testing where peak images have been validated for the evaluation process.

3.1.2. *In-house dataset*

For analysis, the effectiveness of the proposed system, with a sequence of facial expressions in which in-house data has been captured in a lab setting for evaluating the spatiotemporal features. For that, Raspberry Pi and RGB camera modules are utilized to capture real-time spontaneous facial expressions from subjects in an unconstrained environment. The Raspberry Pi is a credit card-sized computer that usually connects to a TV and monitor. The camera port connects the camera with the Raspberry Pi for image and video processing in the computer vision field. Moreover, a 5 MP camera with Raspberry Pi 3/4 Model B has been employed in video capturing for this investigation. The goal of developing this dataset is to distinguish subjects' emotions during learning, which include happy, surprise, sleepy, and neutral. These emotions are recorded in a controlled laboratory setting with a free hand and head movements. The subject consent form is obtained before the study, and participants are asked to make subjective judgments of their feelings after their facial expressions are recorded. It is often called a self-annotation technique.

In detail, there are 40 female subjects aged from 21 to 26 years, with varying amounts of light, occlusion, and positions. This in-house dataset contains a sequence of facial expressions of individuals who consented to the use of their facial expressions for research purposes. Figure 3 depicts the experimental setup. The entire procedure takes place in the Centre for Machine Learning and Intelligence (CMLI). On the other hand, obtaining facial expressions is a difficult task. The video sequence is more useful for emotion recognition rather than static facial expression which does not contain temporal information. In a total of 1600 video frames, four emotions are captured. Happy represents pride and delight when studying, neutral denotes an inactive state of learning, sleepy signals low energy while learning, and surprise denotes a more extreme sense of enjoyment. Each clip stretched for 10–15 s, and the frame rate is 10 frames per second. Furthermore, the frames range from three

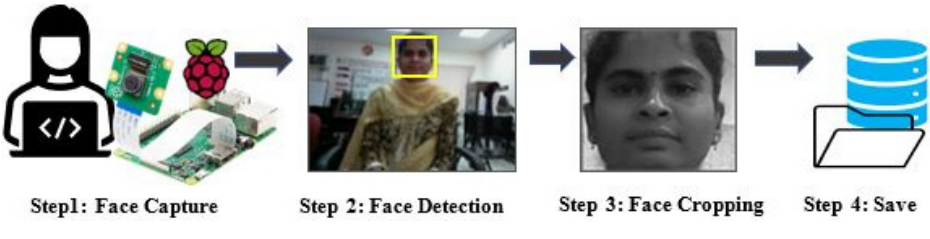


Fig. 3. Framework for facial emotion dataset collection using IoT kit.

peak levels. The algorithm for the facial expression dataset is collected using the following steps:

Step 1. Start the Raspberry camera to capture the subject's spontaneous facial expression.

Step 2. Viola–Jones[48] face detection algorithms are employed to detect the subject's facial expression. This approach is made up of four methods: Haar features, Integral images, Ada-boost, and Cascade classifier. Haar features are utilized to extract face information using line, edge, and four rectangular kernels. Integral images are employed to speed up the Haar feature extraction process, and an Ada-boosting classifier is used to build a strong feature to detect face and nonface in video frames. Finally, a Cascade classifier is utilized to eliminate the nonface region from the video frame.

Step 3. Captured video frames are converted to grayscale and scaled to 48×48 pixels during image pre-processing. The resized frames' probability density function is determined as

$$P(G_M) = \frac{N_M}{N}, \tag{1}$$

where G_M is the number of grayscale video frames in one emotion, N_M denotes the number of frames that occur, and N is the total times of pixels in one frame.

Step 4. Finally, 10 video frames of four emotions are saved in the appropriate folder.

3.2. *Pre-processing*

The term "image pre-processing" describes the transformation of the images before sending them to machine learning and deep learning techniques. The collected facial expression frames are made up of varied lighting conditions and image noise. In addition, grayscale images are widely used rather than color scales which contain less information about facial expressions. Therefore, using RGB photos is not required. So, histogram equalization[24] is done across grayscale video frames to equalize visual contrast. Figure 4 depicts the results. The blue color represents the original pictures,

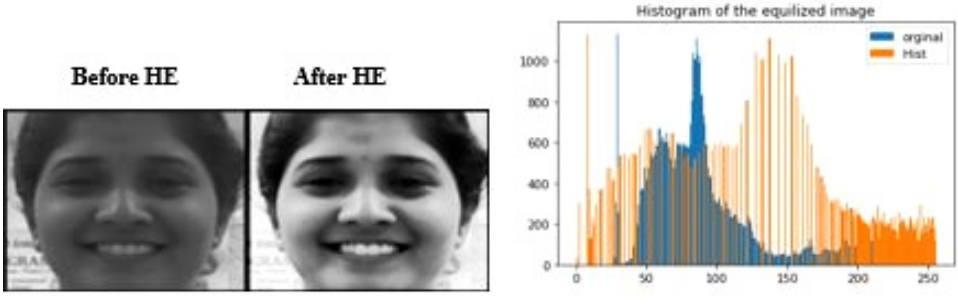


Fig. 4. Histogram equalization.

whereas the orange represents the results after HE. A bilateral filter[55] is used to eliminate noise from video frames. It is one of the image-smoothing filters that is nonlinear, edge-preserving, and noise-reducing. The bilateral filter reduced noise in a video frame more effectively than other filters such as the median filter and Gaussian blur filter. Figure 5 depicts the results of the filter, and the final facial expression after all the processing procedures. Finally, each pixel is divided by 255 to achieve normalization.

3.2.1. *Data augmentation*

Data augmentation[42] is an efficient technique used in deep learning for increasing the quantity of a dataset to increase the model performance. Deep learning-based data augmentation has been the main focus of recent image processing research.[31] The primary goal of this is to prevent overfitting issues caused by insufficient training data. In addition, it helps to generate possibilities of data in a real-world environment. To address these challenges, deep learning will employ data augmentation. The facial images are captured with a straight neck and upward direction, as illustrated in Fig. 5. However, the FER system takes these factors into account when it is built, because the facial position may change in real time depending on the camera's position or the person's posture. There are several techniques such as flipping, rotating, color space, cropping, translation, and noise injection available in data augmentation. Moreover, there are no particular rules for applying data augmentation techniques, it will differ depending on the kind of dataset.[42] For this study, the following technique is applied to the training dataset. Initially, the video frames of the left and right faces are horizontally flipped. Second, each emotion's video frame is rotated from -20° to $+20^\circ$, which is the safest rotation for facial images.[42]

Fig. 5. Collected emotion databases are happy (row 1), neutral (row 2), sleepy (row 3), surprise (row 4).

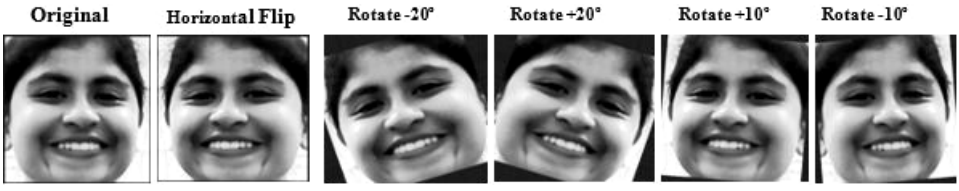


Fig. 6. After data augmentation.

The rotated frames are then horizontally flipped. The data size is expanded by generating synthetic frames by rotating and horizontal flipping, which improved the effectiveness of the proposed model. Figure 6 depicts the outcomes of enhancement procedures.

3.3. *Proposed model combining CNN and LSTM*

The proposed technique aims to discover the relationship between the sequence of facial expressions from their corresponding labels. As previously mentioned, the combination of the contraction and relaxation of one or more facial musculature produces facial expressions; hence, this study has focused on both spatial and temporal features.

3.3.1. *CNN model*

CNN is typically employed to extract spatial information and provides state-of-the-art performance for various computer vision tasks. It has four basic layers: a convolution layer, a pooling layer, a rectified linear unit (ReLU) layer, and a fully connected layer. Figure 7 depicts CNN's core architecture. CNN has been used in a variety of applications[27,32] and has performed admirably in areas such as medical image analysis, image segmentation, object detection, and image classification. The aim of CNN is to extract local descriptors from the top layer and transmit them to the lower level to extract complicated descriptors.

The convolution layer is made up of filters that determine the tensor of each convolution block's feature map. It extracts unique attributes from the given input images.

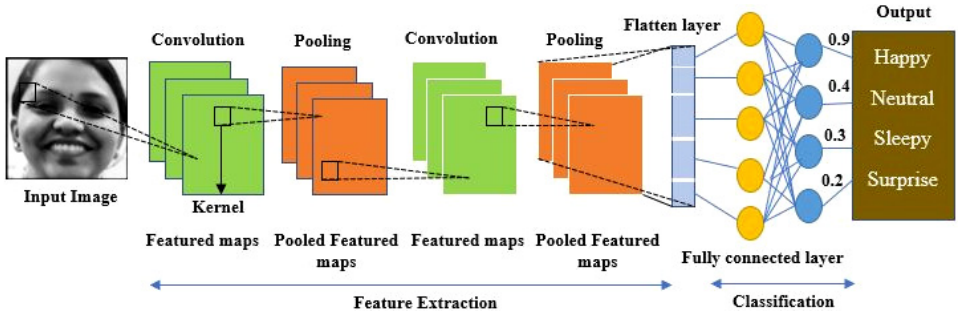


Fig. 7. Basic architecture of CNN.

The kernels (filters) are applied over the input images and use up "stride(s)" such that the result volume size comes to be a numeric matrix. The output of the image dimension is minimized after the striding procedure. Therefore, padding is necessary to pad an input volume with zero while maintaining the size of input images with low-level features. The convolution layer's mathematical process is as follows:

$$F(i,j) = (I * K)(i,j) = \sum I(i+m, j+n)K(m,n), \tag{2}$$

where i denotes the input matrix, K denotes the kernel size $m \times n$, and F denotes the feature map output. I^*K implies the operation between the convolution layer and the kernel. ReLU activation layer is commonly applied to reduce nonlinearity in the output of CNN feature maps. The exponential linear unit (Elu)[9] activation function is utilized in this experiment instead of ReLU as it overcomes the dying problem of the ReLU activation function. It is stated numerically as defined

$$\mathrm{Elu}(x) = \begin{cases} x, & x > 0, \\ \alpha(e^x - 1), & x < 0. \end{cases} \tag{3}$$

The pooling layer down samples incoming data to minimize the dimensionality of feature maps. It decreases the dimension of features to learn as well as the computation time performed for each block of the convolution operation. The most frequent approach is max pooling, which creates the largest value in an input area. Next, the pooling layer, a purposeful dropout layer[44] is employed to generalize the network. It aids in preventing the model from overfitting. Finally, the fully connected layer serves as a classifier, making a judgment based on the basic information gathered from the convolution and pooling layers.

3.3.2. *LSTM model*

LSTM is an extended version of the RNN algorithm that has difficulties of learning high-dimensional data. The conventional RNN architecture has been giving promising results in a shorter length of image sequences, whereas it is giving a poor performance in longer sequences of images due to the vanishing/exploding gradient issues.[17] However, unlike typical RNN units, LSTM incorporates a memory block and was designed to tackle such an issue by offering memory for retaining and forgetting past information over a lengthy period of time. Figure 8 depicts the LSTM's basic structure. It is made up of memory units and three control gates: forget gate, input gate, and output gate, where x_t represents the current input and C_t and C_{t-1} represent the new and prior cell states. Furthermore, h_t and h_{t-1} refer to the new and previous outputs, respectively.[45]

The implementation of various gates aids in the retention of earlier information depending on the network's dependencies. The following diagram depicts the LSTM input gate principle. The input gate (i_t) stores and updates new information about the current state.

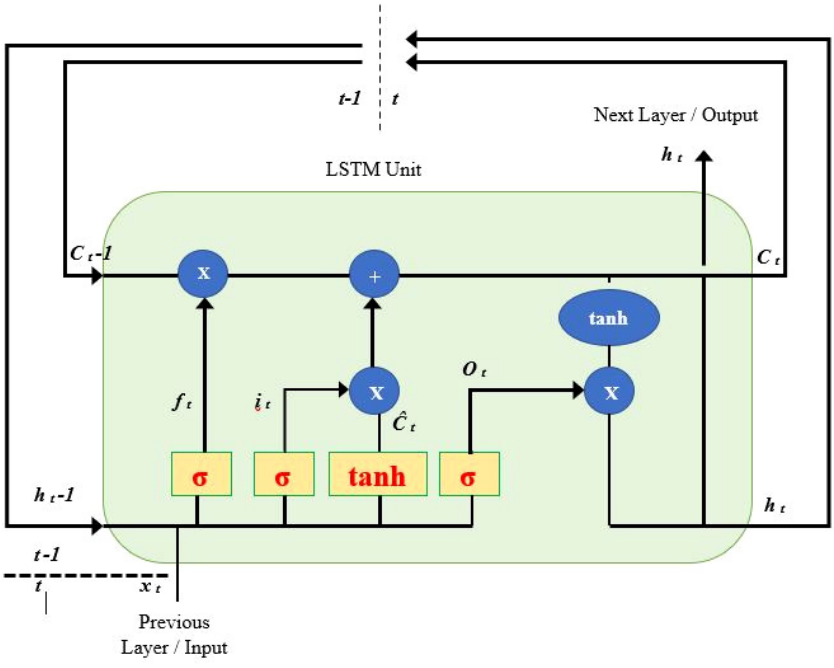


Fig. 8. Basic architecture of LSTM.

$$i_t = \sigma(W_i \cdot [h_{t-1} \cdot x_t] + b_i), \tag{4}$$

$$\hat{C}_t = \tanh(W_C \cdot [h_{t-1} \cdot x_t] + b_C), \tag{5}$$

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t, \tag{6}$$

where (4) is employed to pass the h_{t-1} and x_t via the sigmoid activation layer to determine which portion of the data needs to be added. Following that, (5) is utilized to acquire the new knowledge after h_{t-1} and x_t, which is then sent by the tanh layer. In LSTM units, information transmission is determined by the sigmoid function and the dot product. It values from 0 to 1. If the value of the sigmoid is 1, then information must be transferred. Otherwise, it may not be transferred. $\hat{C}_t$, C_{t-1} refer the long-term memory information and current moment information are joined in (6), where W_C denotes the sigmoid output and $\hat{C}_t$ denotes tanh output. In addition, W_C denotes weight matrix, and input gate bias of LSTM is b_i. The forget gate (f_t) is used in (4) to forget or keep previous information based on network dependencies, where W_f is the weight matrix and b_f is the offset.

$$f_t = \sigma(W_f \cdot [h_{t-1} \cdot x_t] + b_f). \tag{7}$$

The results are generated via the output gate (O_t). Using (7), this identifies the states that must be continued by the h_{t-1} and x_t inputs (8). The final values are derived by

passing the new information, C_t, through the tanh layer using a state decision vector.

$$O_t = \sigma(W_O \cdot [h_{t-1} \cdot x_t] + b_O), \tag{8}$$

$$h_t = O_t * \tanh(C_t), \tag{9}$$

where b_O and W_O are the weighted matrix and LSTM bias output gates, respectively.

3.3.3. *Fusion of CNN–LSTM model*

After analysis, the fusion of CNN and LSTM is feasible for extracting the spatiotemporal features from a sequence of facial images. The next step is to develop a CNN–LSTM combined approach (see Fig. 9) and improve efficiency by making the network stronger. To avoid the complexity and sensitivity of conventional feature extraction approaches, CNN layer is employed to extract the sequence of spatial features from corresponding labels based on frame sequence. After that, extracted feature vector sequence is fed into the designed Bi-LSTM model, which is incorporated with a forward and backward pass to extract spatiotemporal features. The function of LSTM is that it obtains information about the cells preceding a given cell. However, it is unable to access the data from the preceding cell. As a result, the model Bi-LSTM could perform better when processing time series data. Finally, the softmax layer is used to classify each emotion.

Figure 9 illustrates the proposed network of CNN–LSTM. Initially, increasing the number of layers for improving the efficiency of the network, the VGG-19 skeleton is used for extracting the spatial features from a sequence of images. VGG-19, on the other hand, relied on its deeper network structure and feature extraction ability, which reduced FER and caused overfitting issues due to the layer's size. In addition, the vanishing gradient problem slows down the process. subsequently, the layer has been fine-tuned based on the dataset and dimension of the image sequences. The proposed network consists of 21 layers: 8 TimeDistributed layers, 2 dropout layers, 2 batch normalization, 1 flatten layer, 4 pooling layers, 1 FC layer, and 2 LSTM layers with softmax function for categorical classification. Each convolution block is followed by a dropout layer or batch-normalization layer. The 3×3 kernel convolutional layer is utilized for feature extraction and is activated by the Elu activation function. The max-pooling layer is 2×2 in size and is used to minimize the size of the input dimension. The batch-normalization layer[18] is utilized instead of the dropout layer to normalize the output of the activation map, increasing network performance even more. The feature map vector is assigned to the Bi-LSTM layer after the architecture to extract time sequence information. It combines forward and backward LSTM to obtain hidden information from the past and future. The current input points' sequence of features is calculated by the first LSTM, and the reverse sequence features are read and added by the second LSTM. The forward and backward propagation of neurons updates the interaction between neurons in the two states. Therefore, it improves the ability to extract spatiotemporal features. The input form for the LSTM layer became (4,691,903), and Table 1 gives a

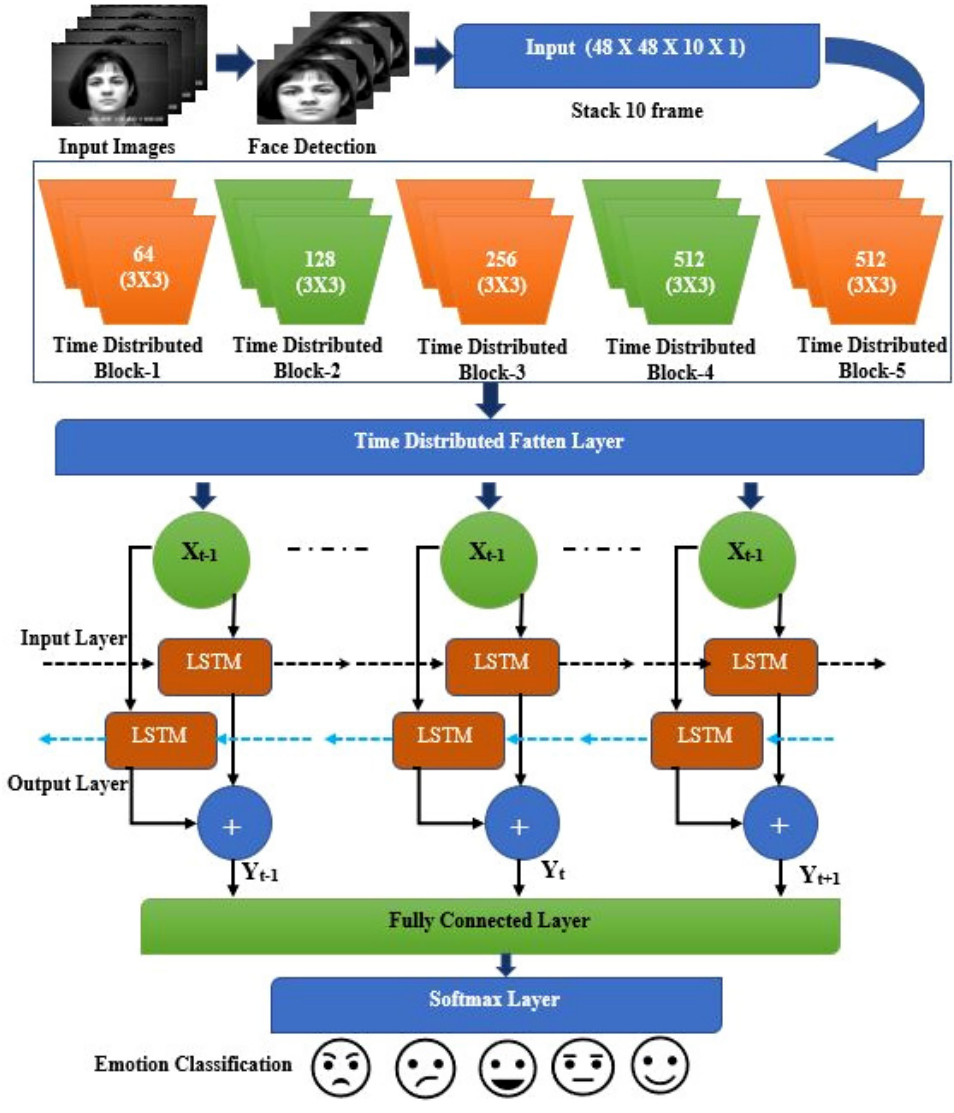


Fig. 9. Enhanced CNN–LSTM (Bi-LSTM) network.

summary of the proposed techniques. The softmax layer then uses labels from the input and the feature map that it has learned to classify and predict each expression. Softmax has been used to efficiently classify the nonlinear function for multi-class classification. It does, however, boost model generality. The softmax layer equation has given in (10). To avoid overfitting issues,[52] regularization techniques, such as dropout, an early stopping method, and kernel_initializer, are applied.

$$P(x) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}. \tag{10}$$

Table 1. Summary of CNN–LSTM (Bi-LSTM) network.

#	Type	Kernel Size	Stride	Kernel/Size	Kernel_Initializer	Parameter
	Input layer					$3 \times 48 \times 48 \times 1$
1	TimeDistributed	3×3	1	64	he_normal	640
2	TimeDistributed	3×3	1	64	he_normal	36,926
3	Pool	2×2	2	—		0
4	Dropout	—	—	0.45		0
5	TimeDistributed	3×3	1	128	he_normal	73,856
6	TimeDistributed	3×3	1	128	he_normal	147,584
7	Pool	2×2	2	—		0
8	BatchNormalization	—	—	—		128
9	TimeDistributed	3×3	1	256	he_normal	2,195,168
10	TimeDistributed	3×3	1	256	he_normal	590,048
11	Pool	2×2	2	—		0
12	Dropout	—	—	0.45		0
13	TimeDistributed	3×3	1	512	he_normal	1,180,160
14	TimeDistributed	3×3	1	512	he_normal	2,359,808
15	Pool	2×2	2	—		0
16	BatchNormalization	—	—	—		2048
17	TimeDistributed	—	—	—		4,691,903
18	LSTM	—	—	512		656,384
19	LSTM			64		164,352
20	FC	—	—	128		16,512
21	Output	—	—	5		645

The parameters setting of the proposed CNN with the LSTM (Bi-LSTM) network is shown in Table 2. The dataset is divided into a training set and a test set, each in an 8:2 ratio, for the aim of obtaining features from the sequence of images. For each round of training, the key model parameters namely the batch size, input size, hidden units, and dropout are monitored regularly. The effectiveness of FER is significantly influenced by the number of hidden units. So, the early stopping method, batch normalization, and dropout are utilized to avoid overfitting and generalize the proposed network. Due to the limited computing capability, the proposed model is trained with 100 iterations and 10 batch sizes. Furthermore, the Adam optimizer[21] incorporates the features of the AdaGrad and RMSProp algorithms to solve the

Table 2. Optimized parameter for proposed method.

Parameter Name	Value
Input size	$3 \times 48 \times 48 \times 1$
Activation function	Elu
Kernel size	3×3
Learning rate	0.001
Batch size	10
Dropout rate	0.3
Iteration	100
Optimizer	Adam

network's sparse gradient and noise challenges. As a result, the Adam optimizer is used to train this network, with a learning rate of 0.001.

4. Results and Analysis

4.1. *Experimental setup*

The FER datasets are partitioned into training and testing parts of 80% and 20%, respectively, for this experiment. The hold-out cross-validation method is employed to obtain the performance data. As can be seen in Table 1, the experimentally used presented network has nine TimeDistributed convolution layers with learning rates of 0.001 and 100 epochs. In order to avoid the overfitting problem, an early stopping method has been adopted in this experiment. The CNN, LSTM, and CNN with LSTM networks, as well as the proposed networks, have been evaluated on Google Colab environments using Python and Kera's package with TensorFlow backends. Furthermore, the studies were carried out using GPU) equipped with an Intel(R) Core (TM) i5-8400 CPU @ 2.80GHz 2.81 GHz Dell desktop.

4.2. *Performance evaluations*

The following performance metric is used to gauge the proposed system's effectiveness. TP indicates the True Positive of accurately predicted emotions. FP denotes the False Positive of misclassified classes. The True Negative of correctly recognized emotions is denoted by TN, whereas the False Negative of the FER system is denoted by FN, which is misclassified emotions. The following equations are used to calculate the accuracy, precision, recall, and $F1$-measure:

$$\text{Accuracy} = (\text{TP} + \text{TN})/(\text{FP} + \text{FN} + \text{TP} + \text{TN}), \tag{11}$$

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP}), \tag{12}$$

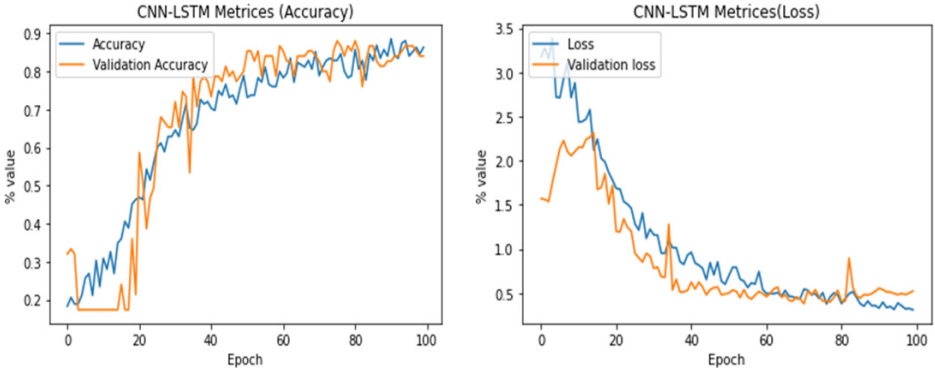
$$\text{Recall} = \text{TP}/(\text{TP} + \text{FN}), \tag{13}$$

$$F1\text{-measure} = 2 \times (\text{Precision} \times \text{Recall})/(\text{Precision} + \text{Recall}). \tag{14}$$

Accuracy is defined in this case as the ratio of correctly classified test frames to total frames. Precision is defined as the ratio of the number of correct frames in the test set for that emotion to the number of correct frames in the emotion recognition results. Recall refers to the ratio of the number of correctly identified frames in that emotion to the total number of frames in those emotions. Finally, the $F1$-score provides the proposed system's average accuracy and recall measures.

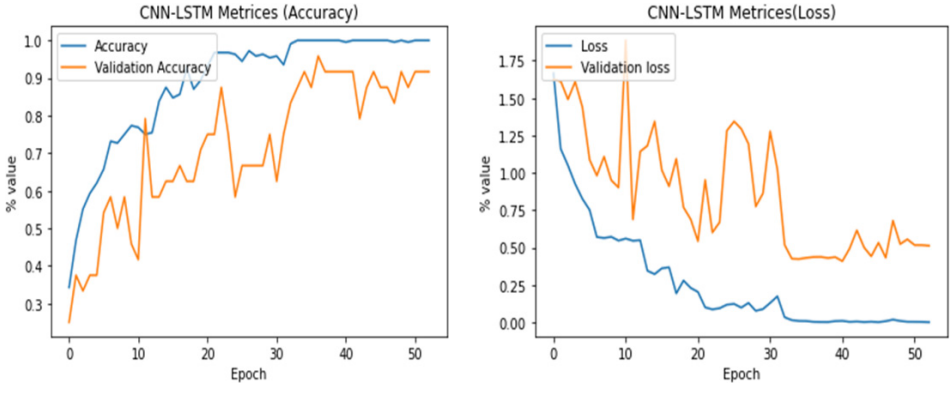
4.3. *Results on proposed network*

Figures 10 and 11 show the accuracy and cross-entropy (loss) performance evaluations of the proposed model on the CK+ and in-house datasets during the training and testing phases. In the CK+ dataset, at epoch 100, the training and testing accuracy are, respectively, 0.91% and 0.84%. Similar to this, the in-house dataset's



(a) (b)

Fig. 10. Timing process of CNN–LSTM in CK+ dataset (a) accuracy (b) loss.

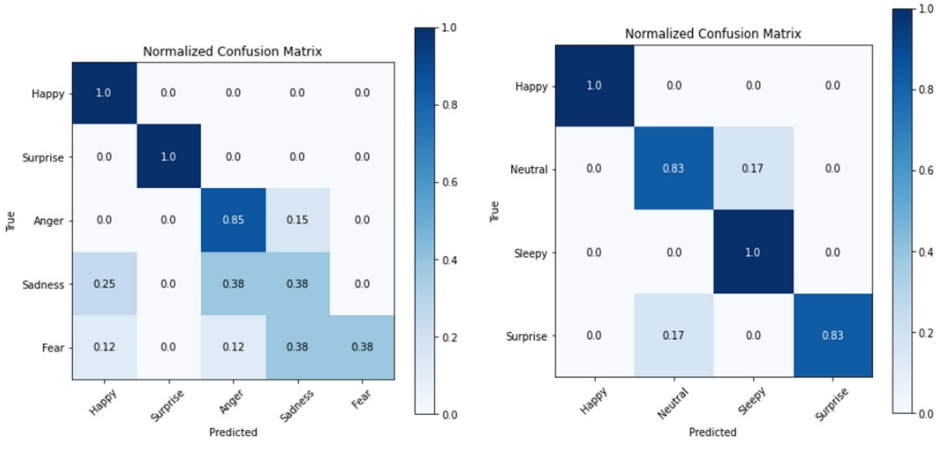


(a) (b)

Fig. 11. Timing process of CNN–LSTM in in-house dataset (a) accuracy (b) loss.

training and testing accuracies are 0.99% and 0.92%, respectively, at epoch 50. The training's initial stage's slower convergence speed could be seen, but it then picked up and converged at epochs 90–100 (see Fig. 10(a)), proving that the model's learning efficiency is satisfactory. The slope of the training curve is reduced during model training. Same as in Fig. 11(a), training accuracy gradually increased with slight fluctuation. Furthermore, the CK+ and in-house datasets have training and testing losses of 0.84 and 0.61, respectively.

Figure 12 illustrates the confusion matrix of the proposed model on both datasets. Figure 12(a) shows the performance on CK+ dataset. Among 750 image sequences, 21 were misclassified due to the similarity of facial appearances, with five emotions. From Fig. 12(a), happy and surprise achieved superior performance due to the sequence of facial features and time series information, whereas, anger, sad, and fear



(a) (b)

Fig. 12. (a) Confusion matrix of CK+. (b) Confusion matrix of in-house dataset.

are misclassified as other emotions slightly. Anger is falsely classified as sad, sad is falsely classified as happy and anger. Fear is falsely classified as sad, anger, and happy. The precision of four emotions varied from 0.83 to 1.0, the recall ranged from 0.83 to 1.0, and the $F1$-score ranged from 0.83 to 1.0, respectively, while the average precision and recall were 0.92 and 0.91, respectively (see Fig. 13). From Fig. 12(b), it is seen that happy and sleepy achieved a greater performance compared to neutral and surprise. Similarly, neutral overlapped sleepy expression, and surprise slightly overlapped with a neutral expression. The precision of five emotions varied from 0.38 to 1.0, the recall ranged from 0.38 to 1.0, and the $F1$-score ranged from 0.38 to 1.0, respectively, while the average precision and recall were 0.85 and 0.84, respectively (see Fig. 14). Due to similarities in the shape and appearance of the facial features and individual variations of the same facial expression, an expression is typically easily confused with another expression. The proposed CNN with LSTM yields notable results, more reliable true positive and true negative values, fewer false negative and false positive values, and more consistent true positive and true negative values. As a result, the proposed model is able to efficiently categorize the sequence of emotions.

Additionally, the receiver operating characteristic (ROC) curves between true positive rate and false positive rate are presented in Fig. 15 to assess the overall performance. The ROC curve clearly reveals that the suggested model performance is determined to be 0.92 in the in-house dataset and 0.84 in the CK+ dataset. The primary goal of this research is to discover the relationship between sequences of images of human facial expressions and the labels that correspond to them. In addition, it aims to extract the spatiotemporal features using a CNN architecture with LSTM (Bi-LSTM). From Fig. 15(a), happy and sleepy have been correctly identified, however, neutral and surprise have been slightly misclassified with 0.17%. From Fig. 15(b), happy and surprise are classified with a higher rate, fear and anger are

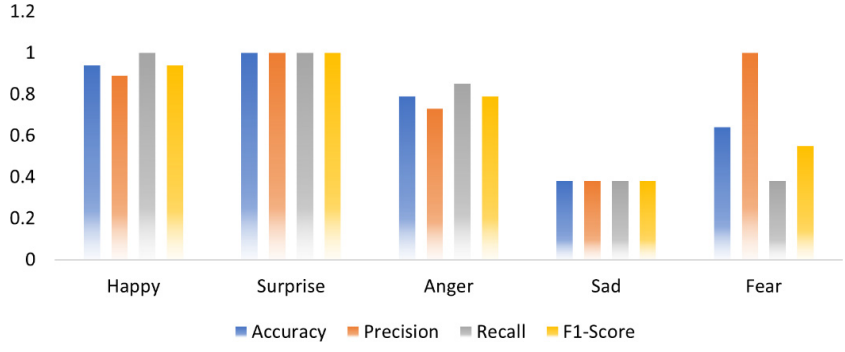


Fig. 13. Recognization accuracy of proposed model on CK+ dataset.

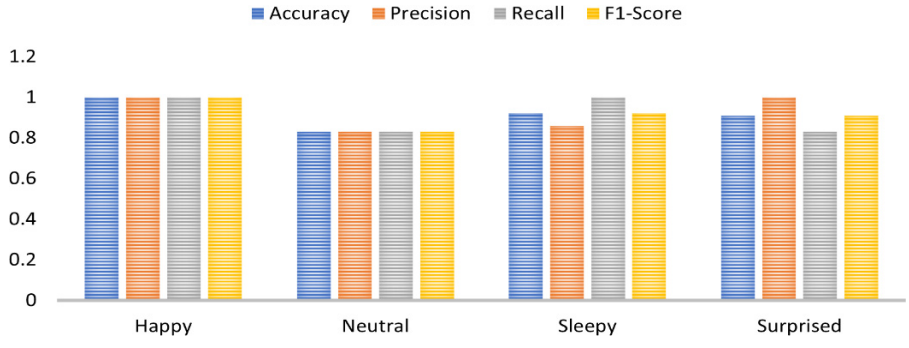
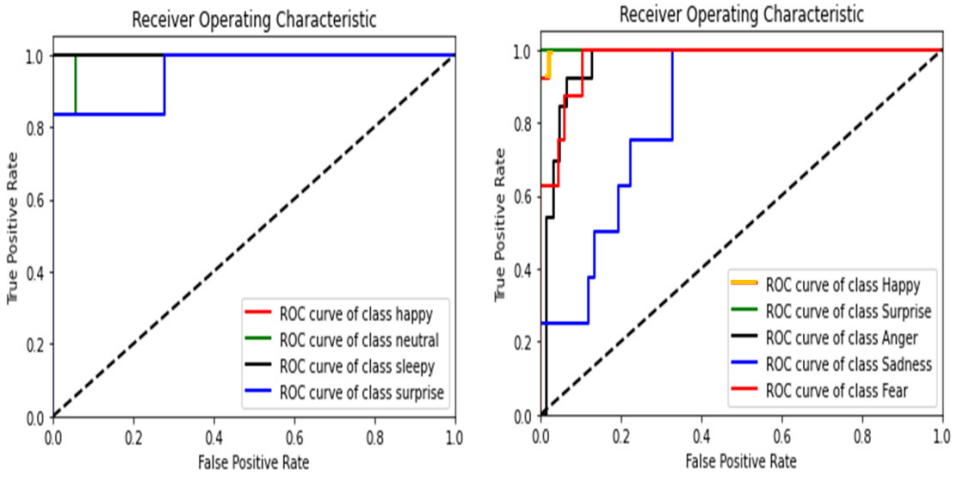


Fig. 14. Recognization accuracy of proposed model on in-house dataset.



(a) (b)

Fig. 15. ROC curve of the proposed model (a) in-house dataset (b) CK+ dataset.

given satisfiable results, and sad is given little lower results in the CK+ dataset. Furthermore, the primary goal of this research is to achieve satisfactory results in recognizing facial emotion from a dynamic sequence of facial expressions. The results of the experiment show that the proposed method outperformed the competitive state-of-the-art methods (see Tables 4 and 5).

4.4. *Comparison of baseline network and state-of-the-art FER models*

To evaluate the performance and generalization of the proposed model, hold-out cross-validation techniques are used to compare it to another baseline network such as CNN, LSTM, and CNN–LSTM. The above-mentioned model which has the same structure as the proposed model has been designed and validated. Table 3 summarizes the training parameters for each model. All models have the same learning rate, loss function, epoch, batch size, optimizer, and activation function; these parameters influence model performance while training. The gradient calculation is heavily influenced by the loss function and optimizer. The Learning rate is used to regulate the parameters which are updated with respect to the gradient. In most cases, CNN and LSTM networks use the ReLU activation function. In this study, the Elu activation function has been used instead of ReLU to address the dying ReLU problem. Moreover, Elu outperformed LeakyReLU. When compared to the other architectures in Table 3, the proposed model has the fewest parameters. The proposed model and all other models have been trained and tested with CK+ and an in-house dataset, shown in Table 4. The proposed model performance clearly boosted from 69.85% to 84.87% on CK+, and 74.56–92.84% on the in-house dataset due to time series features. From Table 4, CNN architecture is given poor performance with time sequence information whereas CNN–LSTM (only 78.34 on CK+, 84.32% on in-house) has given slightly superior results compared to general CNN (only 69.84 on CK+, 74.56% on in-house) and LSTM (only 50.78 on CK+, 79.38% on in-house) network. In addition, the computational time is also calculated for each model. Compared to

Table 3. Parameter settings of each baseline model.

Methods	Learning Rate	Epoch	Batch Size	Loss Function	Optimizers	Parameters
CNN	0.001	100	10	Cross-entropy	Adam	76 M
LSTM	0.001	100	10	Cross-entropy	Adam	57 M
CNN + LSTM	0.001	100	10	Cross-entropy	Adam	49 M
Proposed model	0.001	100	10	Cross-entropy	Adam	46 M

Table 4. Comparison between different models on CK+ and in-house dataset.

Methods	Accuracy CK+ (%)	Time	Accuracy in-House Dataset (%)	Time
CNN	69.84	8 min	74.56	12 min
LSTM	50.78	10 min	79.38	16 min
CNN + LSTM	78.34	7 min	84.32	10 min
Proposed model	84.87	5.5 min	92.84	7.5 min

Table 5. Performance analysis of the proposed model with state-of-the-art models.

Approach	Dataset	Accuracy (%)
SVM + NN[47]	CK+	80.00
PCA + KNN[48]	CK+	77.29
CNN + KNN[49]	CK+	80.30
	JAFFE	76.74
PCA + NMF + LNMF[50]	CK+	81.40
CNN[51]	In-house	78.04
Topographic Context + LDA[52]	CK+	82.68
HOSVD[53]	CK+/JAFFE	73.30
AAM + AUs + SVM[54]	CK+	54.47
Proposed method	CK+	**84.87**
	In-house	**92.84**

other baseline models proposed model took less time for training the model. The amount of the dataset and the complexity of the image sequences always affect the model's training time and accuracy. Additionally, the proposed model has been examined using current state-of-the-art methodologies, as shown in Table 5. From Table 5, it is found that some of the existing systems[5,6,33,35,40,41,47,49] obtained slightly lower accuracy in the range of 54.47–76.74%. The moderately reasonable accuracy of 77.29%, 78.04%, 80.00%, 80.30%, 81.40%, and 82.68% are found in Refs. 5, 33, 35, 41, and 49, respectively. The proposed model's 2.19% accuracy is increased compared to the state-of-the-art methods.

5. Discussion

The proposed model has been compared with baseline models such as CNN, LSTM, and CNN with LSTM (VGG-19) network. The collected in-house datasets have various challenging conditions such as illumination, partial occlusion, and noise. The proposed hyperparameter-tuned CNN with LSTM performed efficiently in this dataset as well as the benchmark CK+ dataset. This integrated model has been investigated experimentally on CNN and LSTM networks individually before being combined for extracting spatiotemporal features from a sequence of images. The majority of the facial expression datasets were created using a high-quality camera in a controlled setting. In this work, an emotion dataset has been captured by a low-quality 5 MP camera module attached to a Raspberry Pi to analyze the FER system's performance with challenging images. First, a FER has been trained with CNN, which yielded unsatisfactory performance testing results because CNN is incapable of extracting spatiotemporal properties from image sequences.[1] Nevertheless, this network has been working effectively in 2D images. Second, the LSTM model has been used to evaluate the performance of emotion recognition from the feature vector from the image sequence. This network did not give efficient results in images, because, when flattening the image sequence $48 \times 48 \times 3 \times 1$, it could not handle the massive size of the features. As a result, the performance of LSTM is insufficient for

developing a FER system model. Finally, features have been extracted using the VGG-19 architecture and fed into the LSTM network. Due to the number of layers, the FER model suffered from the vanishing gradient problem.[16] As a result, this model has been optimized by neuron size, layer count, batch normalization, dropout, and learning rate.

Furthermore, the proposed model is examined with pre-processing, which boosted the recognition rate to 0.92, and without pre-processing, which dropped the recognition rate to 0.80. When CNN and LSTM models are evaluated individually, the performance suffered due to overfitting and misclassification of sleepy and neutral emotions. Similarly, the VGG-19 architecture model has been used to extract spatiotemporal features from the sequence of images without tweaking the layer size, and activation function before entering the LSTM network. This combined network has given better results compared to the general CNN and LSTM network, even though, a hypermeter-tuned CNN with an LSTM model improved the recognition accuracy and classification outcomes rather than the VGG-19 architecture. Table 4 depicts the effectiveness of the proposed system with different baseline architectures. In addition, the proposed model marginally suffered in generalization when training due to insufficient sample size. Moreover, it does not possess the ability to recognize unseen emotions which did not fall in training. In the future, more subjects' facial samples will be collected for each expression from neutral to peak for analyzing the performance of the proposed system. However, when compared to the other state-of-the-art techniques, the proposed model outperforms them.

6. Conclusion

In the last two decades, emotion recognition is an active research area due to its application in many fields. The contraction and relaxation of some facial muscles produce continuous facial expressions. Hence, it is necessary to consider the dynamic and temporal features when recognizing facial emotion from facial expressions. The proposed model extracts spatiotemporal features from a sequence of frames, whereas existing CNN–LSTM works on FER with single images for emotion recognition. Initially, CNN is used to extract spatial features, while Bi-LSTM is used to identify time information in order to classify emotions based on labels. For this study, the in-house dataset has been collected with a small 5 MP camera with Raspberry Pi. The collected dataset consisted of high illumination and noise. Those challenges have been tried to minimize by pre-processing before entering the CNN model. This proposed system is trained from scratch; moreover, data augmentation, batch normalization, and dropout improve the proposed system's efficiency. In addition, the proposed model evaluated by the benchmark dataset achieved 0.84% and 0.92% on the in-house dataset. Furthermore, baseline model efficiency on the FER dataset is also compared with the proposed CNN–LSTM (Bi-LSTM) which efficiently learns the relationship between sequences of facial expression for identifying emotions. This

study will be expanded in the future to recognize accurate facial emotion recognition while combining it with other multimodal aspects.

Acknowledgment

This work has been supported by the CMLI funded by the Department of Science and Technology (DST-CURIE).

References

1. M. Asim, Z. Ming and M. Y. Javed, CNN-based spatiotemporal feature extraction for face anti-spoofing, in *2017 2nd Int. Conf. Image, Vision, and Computing (ICIVC)* (IEEE, 2017), pp. 234–238.
2. M. A. Azizan *et al.*, Development of real-time emotion recognition system based on machine learning algorithm, in *Human-Centered Technology for a Better Tomorrow* (Springer, Singapore, 2022), pp. 101–114, https://doi.org/10.1007/978-981-16-4115-2_8.
3. M. Bai and R. Goecke, Investigating LSTM for micro-expression recognition, in *Companion Publication of the 2020 Int. Conf. Multimodal Interaction* (Association for Computing Machinery, 2020), pp. 7–11, https://doi.org/10.1145/3395035.3425248.
4. S. A. Bargal, E. Barsoum, C. C. Ferrer and C. Zhang, Emotion recognition in the wild from videos using images, in *Proc. 18th ACM Int. Conf. Multimodal Interaction* (Association for Computing Machinery, 2016), pp. 433–436, https://doi.org/10.1145/2993148.2997627.
5. M. S. Bilkhu, S. Gupta and V. K. Srivastava, Emotion classification from facial expressions using cascaded regression trees and SVM, in *Computational Intelligence: Theories, Applications and Future Directions - Volume II* (Springer, Singapore, 2019), pp. 585–594.
6. I. Buciu and I. Pitas, Application of non-negative and local nonnegative matrix factorization to facial expression recognition, in *Proc. 17th Int. Conf. Pattern Recognition, 2004. ICPR 2004*, Vol. 1 (IEEE, 2004), pp. 288–291.
7. Q. Cao, L. Shen, W. Xie, O. M. Parkhi and A. Zisserman, VGGFace2: A dataset for recognising faces across pose and age, in *2018 13th IEEE Int. Conf. Automatic Face & Gesture Recognition (FG 2018)* (IEEE, 2018), pp. 67–74, https://doi.org/10.1109/FG.2018.00020.
8. M. K. Chowdary, T. N. Nguyen and D. J. Hemanth, Deep learning-based facial emotion recognition for human–computer interaction applications, *Neural Comput. Appl.* (2021) 1–18, https://doi.org/10.1007/s00521-021-06012-8.
9. D. A. Clevert, T. Unterthiner and S. Hochreiter, Fast and accurate deep network learning by exponential linear units (ELUs), preprint (2015), arXiv:1511.07289.
10. J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko and T. Darrell, Long-term recurrent convolutional networks for visual recognition and description, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (IEEE, 2015), pp. 2625–2634.
11. P. Ekman and W. V. Friesen, Constants across cultures in the face and emotion, *J. Pers. Soc. Psychol.* **17**(2) (1971) 124.
12. P. Ekman and W. V. Friesen, Facial action coding system, *Environ. Psychol. Nonverbal Behav.* (1978).
13. Y. Fan, X. Lu, D. Li and Y. Liu, Video-based emotion recognition using CNN-RNN and C3D hybrid networks, in *Proc. 18th ACM Int. Conf. Multimodal Interaction* (Association

for Computing Machinery, 2016), pp. 445–450, https://doi.org/10.1145/2993148.2997632.
14. X. Feng, M. Pietikäinen and A. Hadid, Facial expression recognition based on local binary patterns, *Pattern Recognit. Image Anal.* **17**(4) (2007) 592–598, https://doi.org/10.1134/S1054661807040190.
15. J. Haddad, O. Lézoray and P. Hamel, 3D-CNN for facial emotion recognition in videos, in *Int. Symp. Visual Computing* (Springer, Cham, 2020), pp. 298–309, https://doi.org/10.1007/978-3-030-64559-5_23.
16. K. He, X. Zhang, S. Ren and J. Sun, Deep residual learning for image recognition, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (IEEE, 2016), pp. 770–778.
17. S. Hochreiter and J. Schmidhuber, Long short-term memory, *Neural Comput.* **9**(8) (1997) 1735–1780.
18. S. Ioffe and C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in *Int. Conf. Machine Learning* (PMLR, 2015), pp. 448–456.
19. N. Jain, S. Kumar, A. Kumar, P. Shamsolmoali and M. Zareapoor, Hybrid deep neural networks for face emotion recognition, *Pattern Recognit. Lett.* **115** (2018) 101–106.
20. S. Jaiswal and M. Valstar, Deep learning the dynamic appearance and shape of facial action units, in *2016 IEEE Winter Conf. Applications of Computer Vision (WACV)* (IEEE, 2016), pp. 1–8, https://doi.org/10.1109/WACV.2016.7477625.
21. D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, preprint (2014), arXiv:1412.6980.
22. T. H. S. Li, P. H. Kuo, T. N. Tsai and P. C. Luan, CNN and LSTM based facial expression analysis model for a humanoid robot, *IEEE Access* **7** (2019) 93998–94011.
23. D. Liang, H. Liang, Z. Yu and Y. Zhang, Deep convolutional BiLSTM fusion network for facial expression recognition, *Vis. Comput.* **36**(3) (2020) 499–508.
24. L. Lu, Y. Zhou, K. Panetta and S. Agaian, Comparative study of histogram equalization algorithms for image enhancement, *Proc. Mobile Multimedia/Image Processing, Security, and Applications*, Vol. 7708 (SPIE, 2010), pp. 337–347.
25. P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar and I. Matthews, The extended Cohn-Kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression, in *2010 IEEE Computer Society Conf. Computer Vision and Pattern Recognition — Workshops* (IEEE, 2010), pp. 94–101.
26. B. Martinez and M. F. Valstar, Advances, challenges, and opportunities in automatic facial expression recognition, in *Advances in Face Detection and Facial Image Analysis* (Springer, Cham, 2016), pp. 63–100, https://doi.org/10.1007/978-3-319-25958-1_4.
27. V. Mayya, R. M. Pai and M. M. Pai, Automatic facial expression recognition using DCNN, *Procedia Comput. Sci.* **93** (2016) 453–461.
28. N. Mehendale, Facial emotion recognition using convolutional neural networks (FERC), *SN Appl. Sci.* **2**(3) (2020) 1–8.
29. A. Mehrabian, Communication without words, in *Communication Theory* (Routledge, 2017), pp. 193–200.
30. P. Michel and R. El Kaliouby, Real-time facial expression recognition in video using support vector machines, in *Proc. 5th Int. Conf. Multimodal Interfaces* (Association for Computing Machinery, 2003), pp. 258–264, https://doi.org/10.1145/958432.958479.
31. A. Mikołajczyk and M. Grochowski, Data augmentation for improving deep learning in image classification problem, in *2018 Int. Interdisciplinary PhD Workshop (IIPhDW)* (IEEE, 2018), pp. 117–122, https://doi.org/10.1109/IIPHDW.2018.8388338.
32. K. O'Shea and R. Nash, An introduction to convolutional neural networks, preprint (2015), arXiv:1511.08458.

33. M. Peter, J. L. Minoi and I. H. M. Hipiny, 3D face recognition using kernel based PCA approach, in *Computational Science and Technology* (Springer, Singapore, 2019), pp. 77–86.
34. A. Pise, H. Vadapalli and I. Sanders, Facial emotion recognition using temporal relational network: An application to E-learning, *Multimedia Tools Appl.* **81** (2022) 26633–26653, https://doi.org/10.1007/s11042-020-10133-y.
35. E. Pranav, S. Kamal, C. S. Chandran and M. H. Supriya, Facial emotion recognition using deep convolutional neural network, in *2020 6th Int. Conf. Advanced Computing and Communication Systems (ICACCS)* (IEEE, 2020), pp. 317–320.
36. M. S. Ratliff and E. Patterson, Emotion recognition using facial expressions with active appearance models, in *Proc. HRI* (Citeseer, 2008), pp. 1–6.
37. F. Ren and Z. Huang, Facial expression recognition based on AAM–SIFT and adaptive regional weighting, *IEEJ Trans. Electr. Electron. Eng.* **10**(6) (2015) 713–722, https://doi.org/10.1002/tee.22151.
38. A. Ruiz-Garcia, M. Elshaw, A. Altahhan and V. Palade, A hybrid deep learning neural approach for emotion recognition from facial expressions for socially assistive robots, *Neural Comput. Appl.* **29**(7) (2018) 359–373.
39. A. Sepas-Moghaddam, A. Etemad, F. Pereira and P. L. Correia, Facial emotion recognition using light field images with deep attention-based bidirectional LSTM, in *2020 IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)* (IEEE, 2020), pp. 3367–3371, https://doi.org/10.1109/ICASSP40776.2020.9053919.
40. M. Sert and N. Aksoy, Recognizing facial expressions of emotion using action unit specific decision thresholds, in *Proc. 2nd Workshop Advancements in Social Signal Processing for Multimodal Interaction* (Association for Computing Machinery, 2016), pp. 16–21.
41. K. Shan, J. Guo, W. You, D. Lu and R. Bie, Automatic facial expression recognition based on a deep convolutional-neural-network structure, in *2017 IEEE 15th Int. Conf. Software Engineering Research, Management, and Applications (SERA)* (IEEE, 2017), pp. 123–128.
42. C. Shorten and T. M. Khoshgoftaar, A survey on image data augmentation for deep learning, *J. Big Data* **6**(1) (2019) 1–48.
43. K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, preprint (2014), arXiv:1409.1556.
44. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, *J. Mach. Learn. Res.* **15**(1) (2014) 1929–1958.
45. R. C. Staudemeyer and E. R. Morris, Understanding LSTM — A tutorial into long short-term memory recurrent neural networks, preprint (2019), arXiv:1909.09586.
46. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, Rethinking the inception architecture for computer vision, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (IEEE, 2016), pp. 2818–2826.
47. H. Tan, Y. Zhang, H. Cheri, Y. Zhao and W. Wang, Person-independent expression recognition based on person-similarity weighted expression feature, *J. Syst. Eng. Electron.* **21**(1) (2010) 118–126.
48. P. Viola and M. J. Jones, Robust real-time face detection, *Int. J. Comput. Vis.* **57**(2) (2004) 137–154.
49. J. Wang and L. Yin, Static topographic modeling for facial expression recognition and analysis, *Comput. Vis. Image Underst.* **108**(1–2) (2007) 19–34.

50. G. Xiao, J. Li, Y. Chen and K. Li, MalFCS: An effective malware classification framework with automated feature extraction based on deep convolutional neural networks, *J. Parallel Distrib. Comput.* **141** (2020) 49–58.
51. X. Xu, C. Quan and F. Ren, Facial expression recognition based on Gabor wavelet transform and histogram of oriented gradients, in *2015 IEEE Int. Conf. Mechatronics and Automation (ICMA)* (IEEE, 2015), pp. 2117–2122.
52. X. Ying, An overview of overfitting and its solutions, *J. Phys., Conf. Ser.* **1168**(2) (2019) 022022.
53. M. M. T. Zadeh, M. Imani and B. Majidi, Fast facial emotion recognition using convolutional neural networks and Gabor filters, in *2019 5th Conf. Knowledge Based Engineering and Innovation (KBEI)* (IEEE, 2019), pp. 577–581, https://doi.org/10.1109/KBEI.2019.8734943.
54. L. Zahara, P. Musa, E. P. Wibowo, I. Karim and S. B. Musa, The facial emotion recognition (FER-2013) dataset for prediction system of the micro-expressions face using the convolutional neural network (CNN) algorithm-based Raspberry Pi, in *2020 Fifth Int. Conf. Informatics and Computing (ICIC)* (IEEE, 2020), pp. 1–9.
55. M. Zhang and B. K. Gunturk, Multiresolution bilateral filtering for image denoising, *IEEE Trans. Image Process.* **17**(12) (2008) 2324–2333.

M. Mohana is Ph.D. candidate in the Department of Computer Science, CMLI at the Avinashilingam Institute, Coimbatore, India. She is now working in the field of Mulsemedia and Affective Computing using Deep Learning Techniques. She has qualified for the UGC NET (National Eligibility Test for Assistant Professor) with JRF (Junior Research Fellowship) in June 2020. She received her M.Phil., MCA, and B.Sc. degree in Computer Science from the Bharathiar University in 2020, 2018, and 2015, respectively. She has an experience of one year in teaching in programming languages. Her research has spanned Artificial Intelligence, Facial Emotion Recognition, Computer Vision, Machine Learning, and Deep Learning. She also extended her contribution towards various international collaborations with universities from USA and UK.

P. Subashini is working for the Department of Computer Science, the Avinashilingam University for Women, Tamil Nadu, India, since 1994. She is also the Coordinator of the CMLI sanctioned by the Department of Science and Technology. Her research has spanned many disciplines like Image analysis, Pattern Recognition, Neural Networks, and Computational Intelligence. She has authored and co-authored four books, four book chapters, one monograph, and 145 research papers both at international and national levels. She has 10 sponsored research projects worth more than 2.54 crores from various government funding agencies. She also extended her contribution towards various international collaborations with universities from USA, Germany, and Morocco.

M. Krishnaveni is Assistant Professor in the Department of Computer Science, the Avinashilingam University for Women, Coimbatore, Tamil Nadu, India. She has research experience in Defense projects and worked in disciplines like IoT, Image Processing, Speech Processing, Data Mining, and Computational Intelligence. She has published four books, six book chapters, one monograph, and 86 research papers at both national and international levels. She has research projects under various funding agencies and acts as an active member of the CMLI and coordinates the AI Start-up program (Product Development Lab) for the student. She has received awards such as the best young teacher award IASTE 2017, the best NSS program officer award, NYLP 2016, Government of India.