Proceedings of the Third International Conference on Artificial Intelligence and Smart Energy (ICAIS 2023)
IEEE Xplore Part Number: CFP23OAB-ART; ISBN: 978-1-6654-6216-7

Emotion Recognition using Deep Stacked Autoencoder with Softmax Classifier

M.Mohana
Research Scholar of Computer Science
Centre for Machine Learning and Intelligence (CMLI)
Avinashilingam Institute
Coimbatore, India.
mohana_cs@avinuty.ac.in

Dr.P.Subashini
Professor of Computer Science
Centre for Machine Learning and Intelligence (CMLI)
Avinashilingam Institute
Coimbatore, India.
subashini_cs@avinuty.ac.in

2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS) | 978-1-6654-6216-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICAIS56108.2023.10073937

Abstract — Deep learning and computer vision research are still quite active in the field of facial emotion recognition (FER). It has been widely applied in several research areas but not limited to human-robot interaction, human psychology interaction detection, and learners' emotion identification. In recent decades, facial expression recognition using deep learning has proven to be effective. This performance has been achieved by a good degree of self-learn kernels in the convolution layer which retains spatial information of images with higher accuracy. Even though, it often leads to convergence in non-optimal local minima due to randomized initialization of weights. This paper introduces a Deep stacked autoencoder in which the output of one autoencoder has given into the input of another autoencoder along with input values. A single autoencoder does not sufficient to extract the complex relationship in features. So, these concatenated features of the stacked autoencoder help to focus on highly active features during training and testing. In addition, this approach helps to solve inefficient data issues. Finally, trained autoencoders have fine-tuned with the Adam optimizer, and emotions are classified by a softmax layer. The outcomes of the proposed methodology on the JAFFE dataset are significant, according to experiments. The proposed method achieved 82% of accuracy, 85% of Precision, 82% of Recall, and 81% of F1-score. Additionally, the performance of the stacked autoencoder has been examined using the reconstruction loss and roc curve.

Index Terms — Facial Emotion Recognition (FER), Stacked Autoencoder, Softmax Classifier, Deep Learning, Computer Vision.

I. INTRODUCTION

Emotion recognition plays a prominent role in human-computer interaction (HCI) [1] and can be applied to online gaming, customer feedback assessment, digital marketing, and healthcare. As humans can understand 55% of the message via feelings and emotions through facial expressions, 7% through spoken words, and the remaining are paralinguistic [2]. Understanding human emotion is typically a difficult task that requires machines to recognize and respond to the user's state. Many research works have been going on regarding facial emotion for the last two decades. There are only six facial expressions that are mentioned as universal emotions [3] such as happy, sad, fear, anger, disgust, and surprise while some researchers add neutral and contempt in universal expressions. These emotions called primary emotions remaining are called secondary facial emotions such as pride, shame, awful, etc., In addition, FER is a challenging task when images/videos have illumination, noise, pose variation, and occlusion [7].

Face detection, pre-processing, feature extraction, and emotion classification are the four stages of a conventional facial recognition system. In the first and second stages, a face detection algorithm was used to detect the face in images, and the detected face was then resized, and unwanted parts of the images were removed. According to the literature, the Viola-jones algorithm is a well-known face detection algorithm that is commonly used in the FER system. It is composed of four simple mechanisms: Haar features, integral images, the Ada-boost algorithm, and the cascade classifier [4]. FER relies heavily on feature extraction in the third stage. The feature extraction method is classified into four types: geometric-based, appearance-based, templated-based, and correlation-based. Combination techniques have been used to improve the accuracy of emotion recognition. The techniques of the histogram of oriented gradients (HOG), scale-invariant feature transform (SIFT), and Local Gabor feature [8] are commonly used for FER feature extraction. Since the complexity of similar facial expressions, the dimensionality of features is usually high. So, it often leads to the misclassification of emotion in real time. Therefore, techniques like local binary pattern (LBP), linear discriminant analyses (LCD), and principal component analyses (PCA) are commonly used to reduce the high dimensionality data into the low dimensional data. Finally, a machine learning classifier is used to classify the different emotions.

In recent decades, deep learning techniques achieved superior performance in FER systems particularly convolutional neural networks (CNN) [9] which extract the relevant information from images automatically and reduce the dimensionality of features with the help of powerful kernels. However, the existing FER has only a few layers, although a deeper model achieved better results on other image classifications. When increasing the number of layers in the CNN network may arise generalization issues. In general, a large number of hidden layers in CNN does not always give the result in greater accuracy, which causes vanishing gradient problems. This is the main reason for the FER system's misclassification of emotions in a challenging environment. Additionally, CNN needs a lot of data and processing power. Later days transfer learning techniques [10] has introduced to increase the performance of CNN.

An unsupervised artificial neural network called an autoencoder [11] is used to convert higher-dimensional data into lower-dimensional space. In simple words, it removes unwanted features during the compression stage. Many researchers have utilized this technique to solve various complex real-world problems such as brain disorders, image denoising, image compression, and so on. In addition, recent studies show that the autoencoder plays a significant role in the FER system [8] [18]. The Viola-Jones method is utilized

in this study to identify facial images and resize them to a 48 × 48 pixel size. Next, a deep sparse stacked autoencoder (DSSA) has been proposed for facial emotion recognition. This autoencoder varies from conventional layer-wise pre-trained stacked autoencoder. The proposed autoencoder has initially trained with unsupervised data then the output of the first autoencoder has given into the input of the second autoencoder along with input values. The concatenated features of two autoencoders have been fine-tuned and finally add with a softmax classifier for multiclass classification. Hence, the performance of the stacked autoencoder analysed with various metrics.

The contributions of the proposed works are:

1. Proposed a stacked autoencoder to overcome the non-optimal local minima and insufficient data issues in the conventional approach.
2. Stacked autoencoder employed for dimensionality reduction by concatenating two autoencoder features. Moreover, emotions are classified by a softmax layer.
3. The proposed method experimented with a popular benchmarks dataset namely Japanese Female Facial Expression (JAFFE)[6].

The paper's organizational structure is as follows: Sec 2 explains the existing emotion recognition works based on state-of-the-art approaches in FER and autoencoders. Sec 3 describes the proposed autoencoder and softmax classifier. Sec 4 analyses the results of the proposed approach with the benchmark dataset. Finally, sec 5 concludes the overall presented work performance and is followed by a list of references.

II. RELATED WORK

Deep learning achieved a greater promising result in automatic facial emotion recognition [12] in recent decades. However, the CNN model often requires a huge number of layers to extract the deep spatial features that good representation of data. In addition, finding an optimal parameter for network configuration is a challenging task and requires many attempts to move toward a possible solution. This section summarizes some important existing work on the FER system using deep learning and unsupervised pre-training techniques.

A. Emotion Recognition using CNN

A convolutional neural network could learn more relevant spatial features from images for classification. This unique power has helped to solve many real-world classification problems such as detecting spam emails, optical character recognition (OCR), object detection, x-ray image analysis, and so on. The spatial features play a crucial role in classification problems of computer vision. Particularly in the area of FER, CNN has already shown impressive state-of-the-art classification results for facial traits such as muscular movements, face shape, and texture. Mehendale, N [14] has proposed two parts of the CNN network, one is used to remove the background of the facial images, and another is to concatenate the extracted facial feature vector of each emotion. On the CK+, CMU, and NIST datasets, this is different from a typical CNN network and increases the accuracy of emotion recognition. On their own facial image data set, Pranav et al. [12] employ a deep convolution neural-based FER system. The author achieves an average of 78.04% accuracy. The reason might be an imbalanced and insufficient dataset. The authors used the ReLu activation function, two convolutional layers, two max-pooling layers, and an Adam optimizer with a 0.001 learning rate. Zadeh et al [13] proposed to train the CNN model based on Gabor filter-based features from facial images. Instead of giving raw images, the extracted features increase the CNN network speed and accuracy. Huang et al [15] introduced a two-attention mechanism-based FER system which to solve long-range inductive bias between various facial regions issues in conventional CNN networks using facial expression images. Low-level feature extraction and high-level semantic representation extraction are done using the two-attention mechanism model, respectively. The author achieved better performance on CK+, FER+, and RAF-DB datasets. Hifny, Y., & Ali, A. [16] have proposed a CNN-LSTM-based speech emotion recognition model which extracts the Spatio-temporal features from audio sequences. To classify the emotion, it has finally connected to a fully connected layer with a softmax layer. The authors achieved an overall 87.2% accuracy in 3 phases of testing.

In the FER system, transfer learning has been applied for improving the performance of emotion recognition accuracy when insufficient data and overfitting problems arise. Akhand et al. [17] have introduced a transfer learning-based FER system to increase the accuracy of emotion prediction Transfer learning replaces the dense upper layer in the base model and is fine-tuned with facial expression data with the corresponding label. On the JAFFE and KDFE image datasets, the authors of this study applied well-known transfer learning methods such as VGG-19 and 16, ResNet-18, ResNet-34, ResNet-50, ResNet-151, DenseNet-161, and Inception-v3. This work achieves better results in both the dataset and pre-trained model after 10-fold cross-validation.

B. Unsupervised Pre-Training

Unsupervised learning always yields greater local minima and generalization on training data, and pre-training neural networks are examples of this. Restricted Boltzmann Machines (RBM) has always been used to pre-train CNN and Deep Belief models. In order to learn the association between the image sequences linked to different facial expressions, Elaiwat et al. [19] presented an RBM-based FER system. This model categorizes pose variations and facial expressions by encoding them into two distinct hidden sets, namely facial and non-facial morphlets. This work demonstrated notable performance using cutting-edge techniques. Deep belief networks, according to Li et al.[20], consistently disregard local features on images that have been shown to be essential for face recognition. A combined deep belief network and local feature-centric model for face recognition has been presented by the authors. The ORL face database was used for this experiment, and the results demonstrate a notable improvement in face recognition rate and pose variation face identification. Autoencoder is used for data dimensionality detection and training the network by a greedy layer-wise approach. It produces an output similar to the given input data with a lower dimension. Lakshmi and staked autoencoder-based face emotion identification using modified HOG and LBP feature extraction approaches have been proposed by Ponnusamy

[8]. On the CK+ dataset, the author obtains 97.66% accuracy.

III. PROPOSED METHOD

The proposed FER system based on a deep-stacked sparse autoencoder consists of four steps. The proposed FER system's architecture is seen in Fig. 1. In the first stage, faces in images are detected using the Viola-Jones algorithm [4] and then cropped into a 48 × 48 size. The next step is used to normalize the cropped face images and flatten them into a 1x2304 feature vector. The third step is used to reduce the high-dimensional features into low-dimensional features using a deep-stacked sparse autoencoder. This method helps to choose more relevant features. Finally, the softmax layer is used to classify the facial expression from the extracted features. JAFFE [6] dataset is used for this experiment. The detailed steps have explained below.

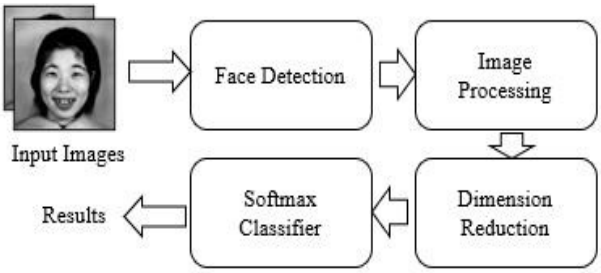


Fig 1. Facial Emotion Recognition Framework

A. Face Detection

First, the face has been detected using the Viola-Jones [4] face detection technique. It is divided into four stages, which are haar features, integral images, Ada-boost, and cascade classifier. Haar features are image features to recognize faces on grayscale images. It is used to identify facial feature locations like eyes, nose, and mouth. An integral image is used to speed up the feature extraction process of haar features. It extracts more than 1,60,000+ features which are all not relevant to identify the face. So, the Ada-boost classifier is used to identify the best features and build a strong classifier for segregating the face and non-face image features out of available huge features. Finally, the cascade classifier is used to ignore the non-face images. Figure 2 depicts the outcome of the viola-jones algorithm. In addition, the detected face has cropped into a 48 x 48 size.

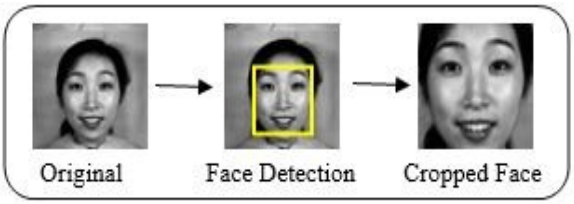


Fig 2. Face detection and cropping

B. Image Processing

Normalization is a vital technique for image processing. It is used to change the pixel intensity values of the images. Sometimes it is called contrast stretching or histogram stretching. In this stage, detected face image pixels are normalized between 0 to 1. This stage is helped to prevent the exploding gradient due to of the high range of the pixels [0-255] and improves the convergence speed. Hence, the cropped and normalized images are flattened into a 1 x 2304 feature vector using the product matrix operator. Finally, these features are fed into the autoencoder.

C. Dimension Reduction

The main goal of this work is to demonstrate how autoencoders reduce high-dimensional data into low-dimensional space in order to learn the most relevant features for identifying target objects.

1) Autoencoders: An autoencoder is an artificial neural network that uses unsupervised learning to encode and decode input data [11]. The aim of the autoencoder is used reconstruct the input data into a meaningful representation with a lower dimension. In an earlier case, principal component analysis (PCA) is used for dimensionality detection for linear data whereas autoencoder is now used for non-linear data reduction. Moreover, the autoencoder is an extended version of the PCA for dimensionality reduction. A simple autoencoder consists of three parts namely encoder, latent space, and decoder. It is called a vanilla autoencoder. Each layer in the autoencoder consists of neuron size, activation function, and padding. Furthermore, the input and output layers always have the same number of neurons, whereas the hidden layer has fewer neurons, making it more efficient than a conventional feed-forward neural network. Spare autoencoder, denoise autoencoder, contractive autoencoder, deep autoencoder, and convolutional autoencoder are the regularized versions of the vanilla autoencoder [5].

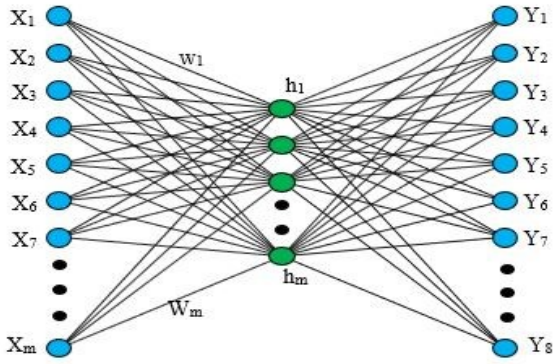


Fig. 3 Architecture of Autoencoder

In the general operation of Autoencoder, first, receive the x input from the user. Then encoder part starts to encode the input data and stored the compressed values in latent space. Next, the decoder starts to reconstruct the input x similar to x'. The primary objective of using an autoencoder is to reconstruct the input feature into a lower dimension for efficient data representation. In addition, this representation can be used for other processes such as data compression, clustering, and feature extraction. The basic operation of an autoencoder can be defined as follows:

$$c = F(w.x + b) \quad (1)$$

$$x' = F(w'.c + b') \quad (2)$$

$$e = min \sum_{i=1}^{n} (x' - x)^2 \quad (3)$$

First, equation (1) is used to encode the input values and get latent space whereas eq. (2) is used to decode the phase to reconstruct the output as similar to input data and Finally, equation (3) is used to fine-tune the whole network and calculate the reconstruction loss between input and output data.

2) Sparse Autoencoder: is used add the penalty on hidden layer or latent space in the autoencoder when the number of hidden layers is large. It is called a sparse autoencoder. So, the overall sparse cost function as defined as follows.

$$J_{sparse}(W,b) = J(W,b) + \beta \sum_{j=1}^{s_2} KL(\rho \| \hat{\rho}_j) \quad (4)$$

Where J(W,b) is the overall cost function,

$$J(W,b) = \left[\frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{2}\left\|h_{w,b}(x^{(i)}) - y^{(i)}\right\|^2\right)\right] + \frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}\left(W_{ji}^{(l)}\right)^2 \quad (5)$$

In equ (5) first term is used to calculate the average sum-of-square error values and the second term is a regularization. Where m, w, b are the training image, weights, bias respectively. λ is a weight decay parameter and n_l defines the number of layers in a sparse autoencoder. s_l defines the number of nodes in layer l , h(x) is a sigmoid function $1/(1+e\hat{}((-x)))$, and y is equal to the input x. KL term denotes the Kullback-Leibler (KL) divergence between ρ and $\hat{\rho}$, s_2 defines the number hidden nodes and β controls the weight of sparsity penalty term which is defined as follows:

$$KL(\rho \| \hat{\rho}_j) = \rho \, log \frac{\rho}{\hat{\rho}_j} + (1-\rho) log \frac{1-\rho}{1-\hat{\rho}_j} \quad (6)$$

where ρ is as sparsity parameter. $\hat{\rho}_j$ is the average activation of hidden unit j:

$$\hat{\rho}_j = \frac{1}{m}\sum_{i=1}^{m}\left[a_j^{(2)}(x^{(i)})\right] \quad (7)$$

$a_j^{(2)}(x^{(i)})$ defines the activation of j[th] hidden node when the input x given into the network.

3) Stacked Autoencoder: Autoencoder can be trained end-to-end or layer-by-layer, then it is latterly called a stacked autoencoder which leads to a deeper encoder with smaller latent space. Fig 1 shows the proposed stacked autoencoder which two autoencoder are arranged cascade manner which helps to learn more relevant information by choosing appropriate features out of the data. The first autoencoder has been trained with unsupervised input data. Then the output of the first autoencoder and input values has both combined and given as input into the second autoencoder which is also trained in an unsupervised manner.

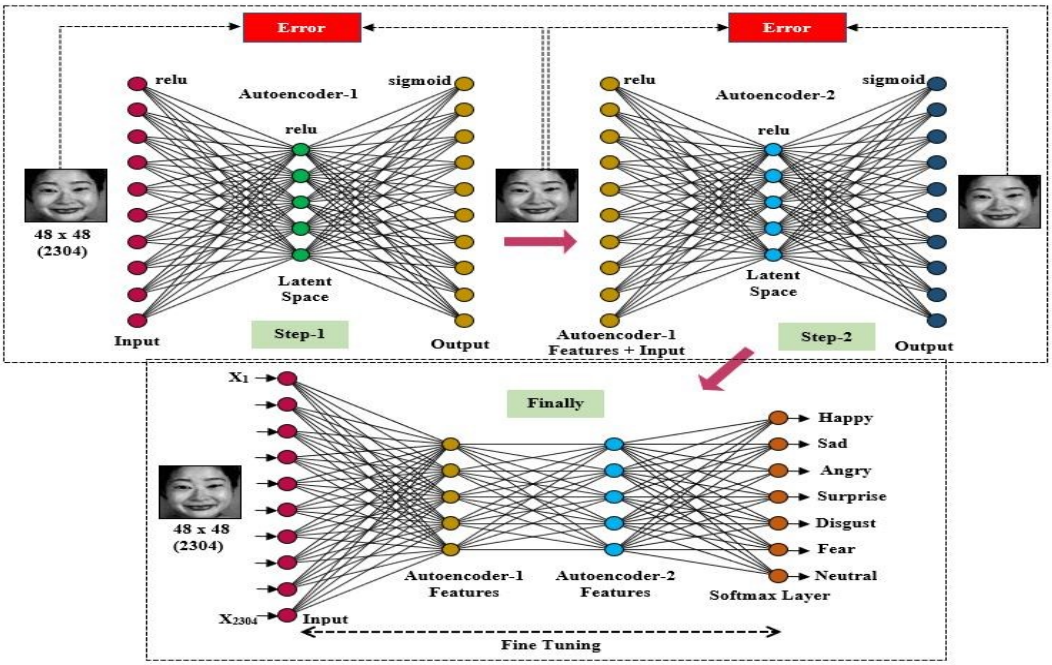


Fig 4. Proposed Deep Stacked Sparse Autoencoder

To assess the performance of the proposed autoencoder, it is fine-tuned by backpropagation. The reconstruction loss is measured by the peak signal to noise ratio (PSNR) and mean squared error (MSE) (see equation 3). Adam optimizer with a learning rate of 0.001 is used for this experimentation. Finally, for emotion classification, these features can be fed into the softmax layer which is activation function for multiclass classification. It has used following the trained stacked autoencoder with input data. The softmax layer is a probability-based sequential classification technique that calculates the probability distribution of input across n dimensions. It employs only the most important

feature, which aids in classification accuracy. Softmax has defined as follows:

$$F(x_i) = \frac{Exp(x_i)}{\sum_{j=0}^{k} Exp(x_j)}, where\ i = 0,1, \dots m \quad (8)$$

IV. RESULTS AND DISCUSSION

A. Dataset

JAFFE [6] facial expression dataset has been taken for this experimental research. Fig 5 shows the sample facial expression of Japanese datasets. It consists of 213 posed facial images of 10 Japanese female subjects. This dataset includes six (happy-31, sad-31, angry-30, surprise-30, disgust-29, and fear-32) basic expressions along with 30 neutral expressions. The resolution of pixel size is 256 x 256, and it is validated by 60 Japanese viewers and is freely available for non-commercial research purposes. Out of this 80% of facial expression, images have been used for training and the remaining 20% for testing.

Fig 5. Sample JAFFE [6] facial expression images

B. Experimental setup

The proposed autoencoder has been implemented using the Keras library and TensorFlow backend in the google colab cloud platform. Furthermore, the studies were carried out using graphics processing units (GPU) equipped with an Intel(R) Core (TM) i5-8400 CPU @ 2.80GHz 2.81 GHz, dell windows 10 64-bit OS desktop. Moreover, table1 and 2 show the parameter setting of the proposed autoencoder.

C. Evaluation Metrics

For this experiment, the following performance metrics are used to evaluate the proposed deep-stacked sparse autoencoder model. TP denotes the True Positive, FN denotes the False Negative, FP denotes the False Positive and FP denotes the False Positive. The summary of four metrics is called a confusion matrix. The value of each metric is calculated from the confusion matrix.

$$Accuracy = \frac{TP+FN}{TP+TN+FP+FN} \quad (9)$$

Accuracy is used to evaluate the detection performance of the whole test set. It can evaluate the positive as positive and the negative as negative.

$$Precision = \frac{TP}{TP+FP} \quad (10)$$

Precision is used to measure the proportion of the real positive samples in the positive samples predicted by the detection model.

$$Recall = \frac{TP}{TP+FN} \quad (11)$$

Eq (8) is used to estimate the exponential of given input x_i and summation of the exponential of all input values and fraction of these results in the output of the softmax function. It returns the probability of each class for multiclassification, where the output class must have the highest probability among all class values.

Recall is used to measure the proportion of the predicted positive case in the whole positive case.

$$F1 - Score = \frac{2TP}{2TP+FP+FN} \quad (12)$$

F1-Score is used to measure the harmonic mean of precision and recall rate and defines the discriminant ability of the designed model for each category.

$$PSNR = 10 \log_{10} \frac{Max_I{}^2}{MSE} \quad (13)$$

The MSE is used to calculate the autoencoder's reconstruction loss, whereas the peak signal-to-noise ratio (PSNR) is used to compare the quality of the original and compressed reconstructed images.

D. Experimental Results

In this section, describes the overall procedure for designing and training of proposed stacked autoencoder. First a deep sparse autoencoder has been designed and trained by unlabeled facial expression Japanese dataset. Table 1 shows the summary of proposed autoencoder with parameter size. This network consists of three encoder dense layer, one latent space dense layer with sparse activation function, three decoder dense layer, one input layer and output layer. In addition, one dense layer has been added for classification using softmax activation function. The hidden layer neuron size is 512, 256, 128 for encoder, 64 for code layer and reversed size for decoder layer. Relu activation function has used in each layer and sigmoid activation function used in output layer. L1 regularization is applied in this proposed network for overcome the overfitting and generalize issues. Moreover, hold-out cross validation method has been used to evaluate the model efficiency. Table 1 shows the optimised parameter setting of deep stacked sparse autoencoder.

Table 1. Summary of Deep Stacked Sparse Autoencoder

Layer (type)	Output shape	Parameters
Input_layer	(None, 2304)	0
Dense_1	(None, 512)	1180160
Dense_2	(None, 256)	131318
Dense_3	(None, 128)	32896
Dense_4	(None, 64)	8256
Dense_5	(None, 128)	8320
Dense_6	(None , 256)	33024
Dense_7	(None, 512)	131584
Dense_8	(None, 2304)	1181951

Table 2. Optimized parameter for proposed model

Parameters	Values
Hidden Layer	5
Neurons	#1 512 #2 256#3 128#4 64#5 7
Activation Function	#1 #2 #3 #4 Relu Sigmoid#5 softmax

Learning Rate	0.001
Loss function	MSE, Categorical_crossentropy, PSNR
Optimizer	Adam
L1 Regularization	le-5
Epochs	100
Batch size	32

Initially presented deep sparse autoencoder has trained with 100 epochs with 32 batch size along with Adam optimizer learning rate 0.001. The mean squared error (MSE) is used to measure the performance of autoencoder. Fig 6 (a) shows the loss curve of first trained autoencoder. The train and validation loss of the first deep sparse autoencoder is 0.0132, 0.0146 respectively. After that the output of the first autoencoder along with input values are concatenated and fed into the input of second autoencoder. The second time stacked autoencoder also trained with 100 epochs ,32 batch size with Adam optimizer with learning rate 0.001. Here, this approach has used to learn more relevant features, reduce dimension of input features and solve the insufficient data problem for training the autoencoder. In addition, Adam optimizer is used for training the proposed autoencoder and for backpropagation learning. It is one of the best optimizers to train the model with less time. Fig 7 shows the train and validation loss of the stacked autoencoder. The loss values are gradually decreased which means the proposed model performed well on the facial expression images. It is one of the important components to measure the error of the neural network. Loss is used to calculate the gradients which are used to update the weights of the network during backpropagation. Fig 6 shows the original images and corresponding proposed autoencoder's generated facial expression images. Most of the facial expressions are reconstructed the same as the original images with lower dimensions. A few of them are slightly different from the original images due to the similarity of facial and emotional expressions. Furthermore, MSE and PSNR are two more commonly used metrics. MSE's limitation is that it is highly dependent on image intensity scaling. PSNR prevents these issues by scaling the MSE according to the image range, and the reconstructed facial expression images achieved 71.62 dB PSNR in the presented network.

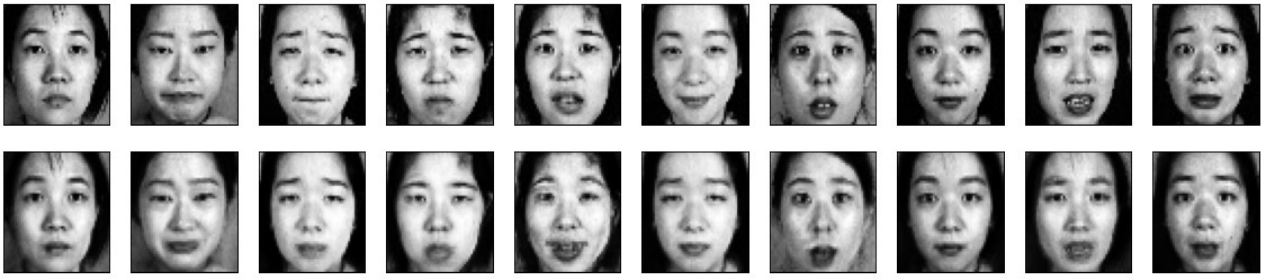
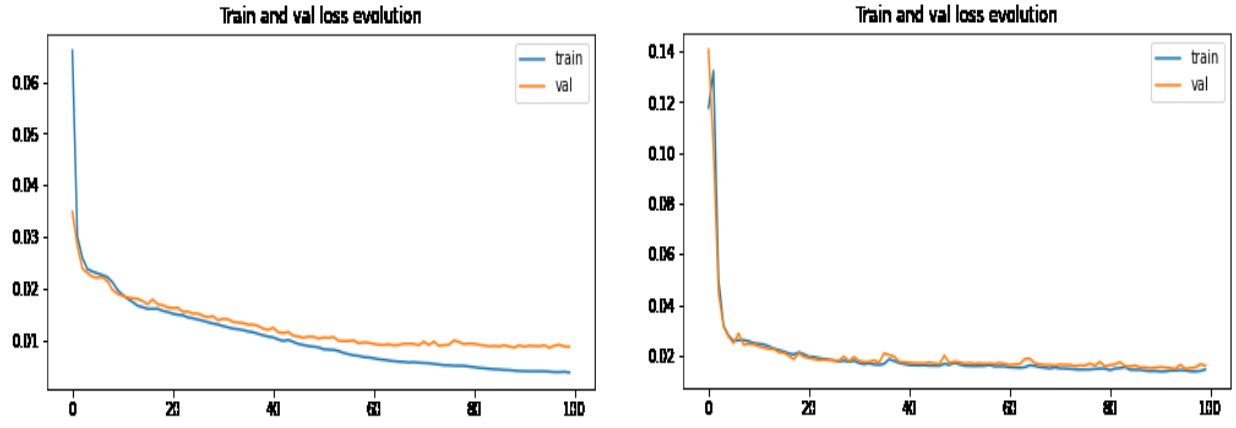
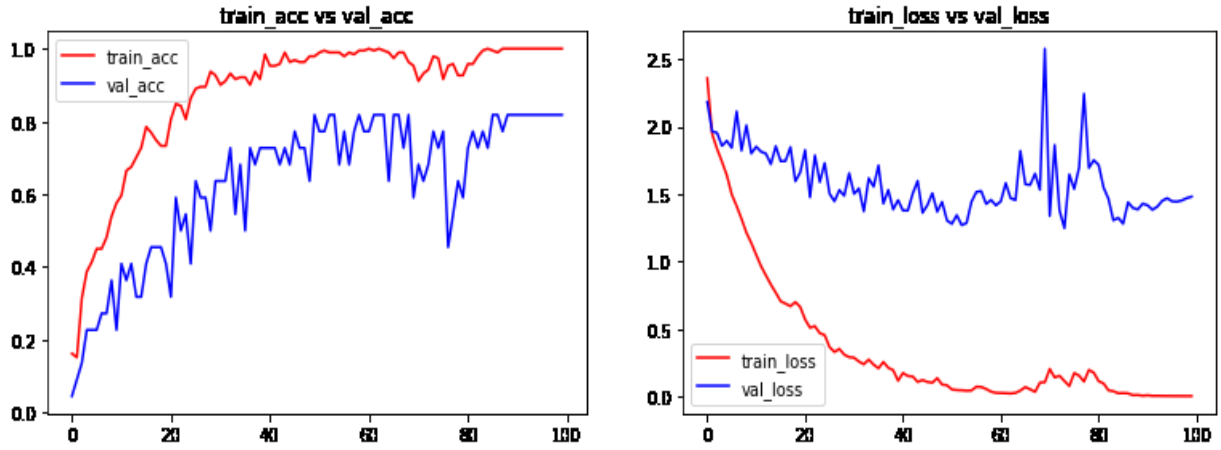


Fig 6. Output of the deep sparse stacked autoencoder, first row shows the original image, and second row shows the proposed autoencoder generated images.



(a) (b)

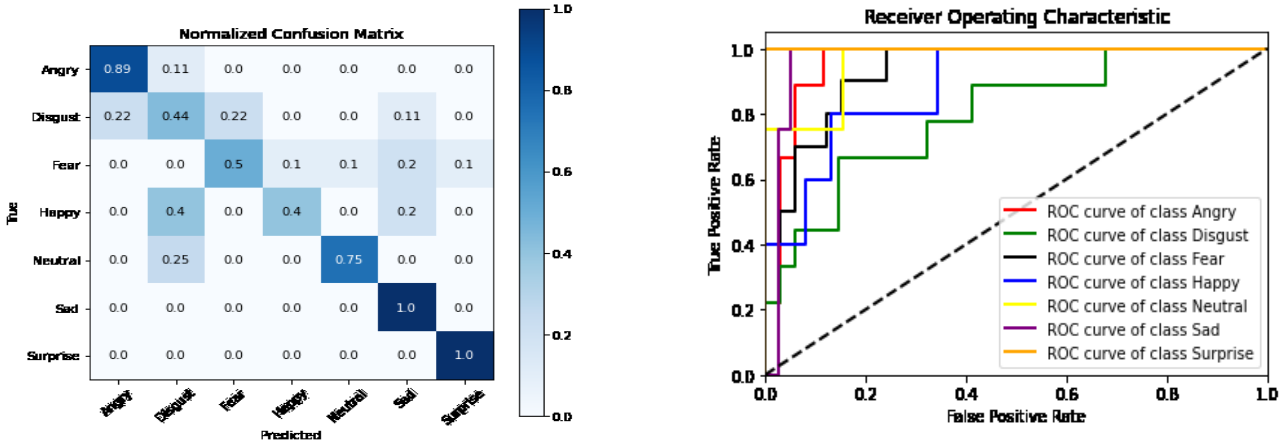
Fig 7. Visual representation of proposed autoencoder loss evaluation curve (a) first deep sparse autoencoder (b) second deep sparse stacked autoencoder



(a) (b)

Fig 8. (a) Training and Validation accuracy of deep sparse stacked autoencoder (b) Training and Validation loss

After the trained stacked autoencoder, the softmax layer has been added, and fine-tuned the whole network with 100 epochs with 32 batch sizes using Adam optimizer and a learning rate of 0.001. Here, Categorical_crossentropy loss function is used to measure the error of the network. This time input images along with corresponding labels have fed into the network for training the whole network in a supervised manner. The train and validation loss of the network is 0.0046, and 1.3666 respectively. Moreover, accuracy and validation accuracies are 0.99% and 0.82% respectively. Fig 8(a) shows the accuracy of the presented network and 8(b) shows the loss evolution of the presented network.



(a) (b)

Fig 9. (a) Normalized Confusion Matrix of deep sparse stacked autoencoder (b) ROC curve of a deep sparse stacked autoencoder

Furthermore, the presented model has been evaluated with precision, recall, f1-score, confusion matrix, and ROC curve which metrics are used to get more details about the accuracy of different facial expressions. Fig 9(a) shows the normalized confusion matrix and 9(b) shows the ROC curve of each facial expression. From this figure, among seven expressions, angry, neutral, sad, and surprise achieved superior performance with the percentage of accuracy 0.89, 0.75, 1.0, 1.0 respectively due to features in the region of eyes and mouth. Meanwhile, the expression of disgust, fear, and happy achieved a satisfactory result while they are easily confused with neutral and sad. In addition, happy with micro expression is easily misclassified as neural. More precisely, anger expression is falsely classified as disgust with 0.11%, and the percentage of disgust falsely classified as angry, fear, and sad are 0.22%, 0.22%, and 0.11% respectively. The percentage of fear that is slightly confused with happy, neutral, sad, and surprise are 0.1%, 0.1%, 0.2%, and 0.1% respectively. Moreover, a happy expression has confused with a disgust expression 0.4% and a sad 0.2%. Typically, an expression is easily confused with another expression due to similarity in shape and appearance of features, and individual variation of the same facial expression.

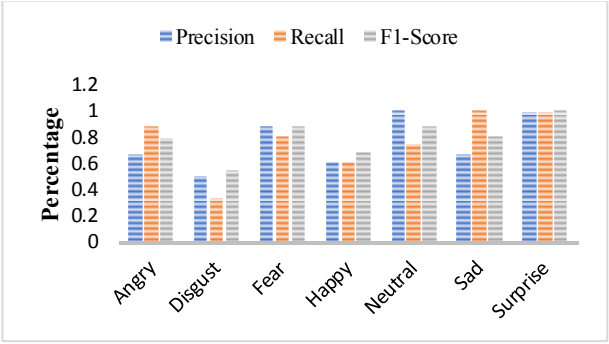


Fig. 10 Recognition performance of each facial expression

Fig 10 shows the precision, recall, and f1-score of each facial expression. From this visual representation presented network has achieved an average of 0.85 % precision, 0.82 % recall, and 0.81 % of f1-score. Moreover, the ROC curve shows the performance the of classification of individual facial expressions with threshold values.

Table 3. Performance comparison of different model

The Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Time
CNN	0.55	0.50	0.56	0.60	5 min
Deep Autoencoder	0.60	0.60	0.58	0.61	4 min
Sparse Autoencoder	0.63	0.60	0.71	0.60	4 min
Proposed Model (DSSA)	**0.82**	**0.85**	**0.82**	**0.81**	**3 min**

Sparse autoencoder is used to impose the penalty in the code layer or in the hidden dense layer. Sparse autoencoder can help to extract the more relevant features for classification. For these experiments, L1 regularization has been used to evaluate the facial expression images. Moreover, the presented deep stacked autoencoder model gives a higher accuracy compared to all other models. This network consists of two network features and sparsity helps to reduce the training time and it restricts the active neurons, which reduces the dependency between features. This allows focusing on the most desirable features. This also proves that this proposed model performance analysis results are reasonable, and it took approximately 3min for training.

V. CONCLUSION

In this paper, a deep sparse stacked autoencoder (DSSA) has been introduced for facial emotion recognition. This model can help to recognize the different facial expressions with higher accuracy. The main aim of the deep sparse stacked autoencoder is designed for overcoming the overfitting and data insufficient problems in network training. Initially, a deep sparse autoencoder has trained with unlabeled data then the output of the first autoencoder and input images are concatenated and fed into the second autoencoder for choosing the most relevant features for training. Next softmax layer has added and fine-tuned the whole network with the labeled dataset. This method achieved a greater performance compared to the conventional CNN-based facial emotion recognition approach and helps to reduce the high dimensional features into the lower dimensional which means ignoring unrelated features while compression. In addition, JAFFE facial expression dataset has used for this experiment and Hold-out cross-validation techniques were employed for evaluated the presented model performance. This method achieved a 0.99% training accuracy and 0.82% validation accuracy.

The performance comparison of the different models has been shown in table 3. All the models have been validated with the CK+ dataset. Only a few studies conducted using autoencoder on the FER system with various facial expression datasets. From the analysis, the CNN model did not give satisfactory results due to overfitting and a small dataset. In addition, it suffered to find optimal maxima when random initialization of weights and it took around 5min for feature extraction. Next, a deep autoencoder with more than five layers has been designed and trained with a facial expression dataset. It slightly gave satisfactory results compared to the conventional CNN model. Typically, autoencoders are used to learn more high-level features from unlabeled data with the help of dimensionality reduction techniques [11]. This method helps to learn more complex features from the facial expression dataset. Thirdly sparse autoencoder has designed and experimented with facial expression datasets.

Moreover, precision, recall, f1-score, confusion matrix, and ROC matrices are used to evaluate the accuracy of individual facial expressions.

ACKNOWLEDGMENT

The author wishes to express their sincere thanks to the Centre for Machine Learning and Intelligence (CMLI) sponsored by the Department of Science and Technology (DST), India for providing resources to conduct this research work.

REFERENCE

[1] S. Brave, and C. Nass," Emotion in human-computer interaction", In *The human-computer interaction handbook,* 2007, 103-118. CRC Press.

[2] A. Mehrabian, and J.A. Russell, "*An approach to environmental psychology",* 1974, the MIT Press.

[3] P. Ekman, and D. Cordaro, "What is meant by calling emotions basic", *Emotion review*, *3*(4), 2011, 364-370.

[4] P.Viola, M.J.Jones, "Robust real-time face detection". *International journal of computer vision*, *57*(2), 2004,137-154.

[5] J. Zhai , S. Zhang, J. Chen, and Q.He, "Autoencoder and its various variants", In *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*,2018, 415-419. IEEE.

[6] The Japanese Female Facial Expression Database. Available Online: https://www.kasrl.org/jaffe_download.html (accessed on 29 July 2022)

[7] S. Poria, N. Majumder, R. Mihalcea, and E. Hovy, "Emotion recognition in conversation: Research challenges, datasets, and recent advances", 7, 2019, 100943-100953. IEEE Access,

[8] D. Lakshmi, and R. Ponnusamy, "Facial emotion recognition using modified HOG and LBP features with deep stacked autoencoders", *Microprocessors and Microsystems*, *82*, 2021, 103834.

[9] Y. Lv, Z. Feng, and C. Xu, "Facial expression recognition via deep learning", In *2014 international conference on smart computing,* 2014, 303-308. IEEE.

[10] H.W. Ng, V.D Nguyen, V. Vonikakis, and S. Winkler, "Deep learning for emotion recognition on small datasets using transfer learning", In *Proceedings of the 2015 ACM on international conference on multimodal interaction,* 2015, 443-449.

[11] D. Bank, N. Koenigstein, and R. Giryes, "Autoencoders", *arXiv preprint arXiv:2003.05991*, 2020E.

[12] Pranav, S. Kamal, C.S. Chandran and M.H. Supriya, M. H, "Facial emotion recognition using deep convolutional neural network", In *2020 6th International conference on advanced computing and communication Systems (ICACCS)*, 2020, 317-320. IEEE.
[13] M.M.T. Zadeh, M. Imani, and B. Majidi, "Fast facial emotion recognition using convolutional neural networks and Gabor filters", In *2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI)* 2019, 577-581. IEEE.
[14] N. Mehendale, "Facial emotion recognition using convolutional neural networks (FERC)". *SN Applied Sciences*, 2(3), 2020, 1-8
[15] Q. Huang, C. Huang , X.Wang, and F. Jiang, "Facial expression recognition with grid-wise attention and visual transformer", *Information Sciences*, *580*, 2021, 35-54.
[16] Y. Hifny, and A. Ali," Efficient arabic emotion recognition using deep neural networks", In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, 6710-6714. IEEE.
[17] M.A.H. Akhand, S. Roy, N. Siddique, M.A.S Kamal, and T.Shimamura, T. "Facial emotion recognition using transfer learning in the deep CNN", *Electronics*, *10*(9), 2021, 1036.
[18] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives", *IEEE transactions on pattern analysis and machine intelligence*, *35*(8), 2013, 1798-1828.
[19] S. Elaiwat, M. Bennamoun , and F. Boussaïd, "A spatio-temporal RBM-based model for facial expression recognition", *Pattern Recognition*, *49*, 2016, 152-161.
[20] C. Li, W. Wei, J. Wang, W. Tang, and S. Zhao, "Face recognition based on deep belief network combined with center-symmetric local binary pattern", In *Advanced multimedia and ubiquitous engineering* , 2016, 277-283. Springer, Singapore.