ISSN: 0976-9102 (ONLINE)
DOI: 10.21917/ijivp.2021.0359

FACIAL EMOTION RECOGNITION USING CONVOLUTIONAL NEURAL NETWORK

Milan Tripathi
Department of Computer Engineering, Tribhuvan University, Nepal

Abstract

Facial expressions play a significant role in social communication since they convey a lot of information about people, such as moods, emotions, and other things. Many researchers gained an optimal accuracy in most of the popular facial recognition datasets: CK+, JAFFE, IEV, but in FER2013 the best model accuracy is about 74%. This article purpose deep learning-based models to mitigate this issue. Three models based on AlexNet, VGG19, and ResNet50 are used to train with the dataset, and the very best model among them is further analyzed. The best model is trained using various optimizers and evaluated based on its training and testing accuracy, confusion matrix, ROC Curve. The finest model gained an accuracy of 91.89504% which is better than past state of art models by at least 17% accuracy.

Keywords:

Facial Expression, Confusion Matrix, Emotion, Optimizer, Receiver Operating Characteristic Curve

1. INTRODUCTION

Facial expressions play a significant role in social communication since they convey a lot of information about people, such as moods, emotions, and other things. These are created by the movement of muscles in the face that attach to the skin and fascia, creating lines and folds and causing facial features like the mouth, eye, and brows to move. Because of its numerous uses in artificial intelligence, such as human-computer cooperation, data-driven animation, and human-robot communication, detecting emotion from facial expression will become a pressing requirement. This will also have a wide range of uses, including lie detectors, robotics, and art.

The proposed research of Facial Expression Recognition through Convolution Neural Network (CNN) is to classify human faces based on emotions. With this in mind, three models based on AlexNet, VGG19, and ResNet50 are used to train with the FER2013 dataset. The very best model among the three is further optimized and evaluated. Among all the datasets being used for this task FER2013 [15] has the lowest training and testing accuracy. So, to mitigate this issue, more in-depth analysis is done using various optimizers and K-fold validation done to improve and check the model's performance. In the end, the best model performance is compared with the previous state-of-art models.

2. LITERATURE REVIEW

In recent years, facial emotion recognition has become a hot focus of research. To identify emotion from faces, most people utilize computer vision, machine learning, or deep learning technologies.

This study [1] gives a brief overview of FER research done over the last few decades. The traditional FER techniques are presented first, followed by a description of the typical FER system types and their major algorithms. The authors next describe deep-learning-based FER methods that use deep networks to enable "end-to-end" learning. This paper also looks at a new hybrid deep-learning technique that employs a convolutional neural network (CNN) for spatial characteristics of a single frame and a long short-term memory (LSTM) for temporal data of several frames. A brief overview of publicly accessible evaluation metrics is provided in the latter half of this work, as well as a comparison with benchmark findings, which constitute a standard for a quantitative comparison of FER investigations. Instead of minimizing the cross-entropy loss, learning reduces a margin-based loss.

Study of multi-level features in a convolutional neural network for facial emotion identification by Hai-Duong Nguyen [2]. They offer a model based on the data that purposely combines a hierarchy of characteristics to better the categorization job. The model was tested on the FER2013 dataset and found to be similar to existing state-of-the-art approaches in terms of performance.

Using a feedforward learning model, the authors in [3] developed an instructor's face expression recognition technique within a classroom. For successful high-level feature extraction, the face is first recognized from the obtained lecture videos and important frames are picked, removing all unnecessary frames. Then, using several convolution neural networks and parameter tweaking, deep features are retrieved and supplied to a classifier. A regularized extreme learning machine (RELM) classifier is used to classify five various expressions of the teacher within the classroom for quick learning and effective generalization of the method.

Hernández-Pérez [4] suggested a method that combined oriented FAST and rotated BRIEF (ORB) characteristics with facial expression-derived Local Binary Patterns (LBP) features. To begin, each image is subjected to a face identification algorithm to extract more useful characteristics. Second, the ORB and LBP features are extracted from the face region to boost computational speed; particularly, region division is used in a novel way in the classic ORB to prevent feature concentration. The characteristics are unaffected by changes in size, grayscale, or rotation. Finally, a Support Vector Machine is used to classify the collected characteristics (SVM). The suggested technique is put to the test on several challenging datasets, including the CK+, JAFFE, and MMI databases.

Zhang Qinhu [5] proposes a paper that first introduces the self-attention mechanism based on the residual network concept and calculates the relative importance of a location by calculating the weighted average of all location pixels, then introduces channel attention to be told completely different options on the channel domain, and generates channel attention to target the interactive options in a variety of channels. The accuracy of this study on the CK+ and FER2013 datasets, respectively, is 97.89% and 74.15

percent, demonstrating the model usefulness and superiority in extracting world choices.

Zahara [6] proposed a facial image threshing (FIT) machine that incorporates sophisticated characteristics of pre-trained facial recognition and Xception algorithm training. In addition to the data-augmentation methodology, the FIT machine required deleting extraneous facial photographs, gathering facial photos, correcting misplaced face data, and integrating original information on a vast scale. With the FER2013 dataset, the final FER results of the suggested method enhanced validation accuracy by 16.95% over the conventional approach.

3. SYSTEM OVERVIEW

Our work is to classify facial images based on emotion. The Fig.1 is the overall system architecture.

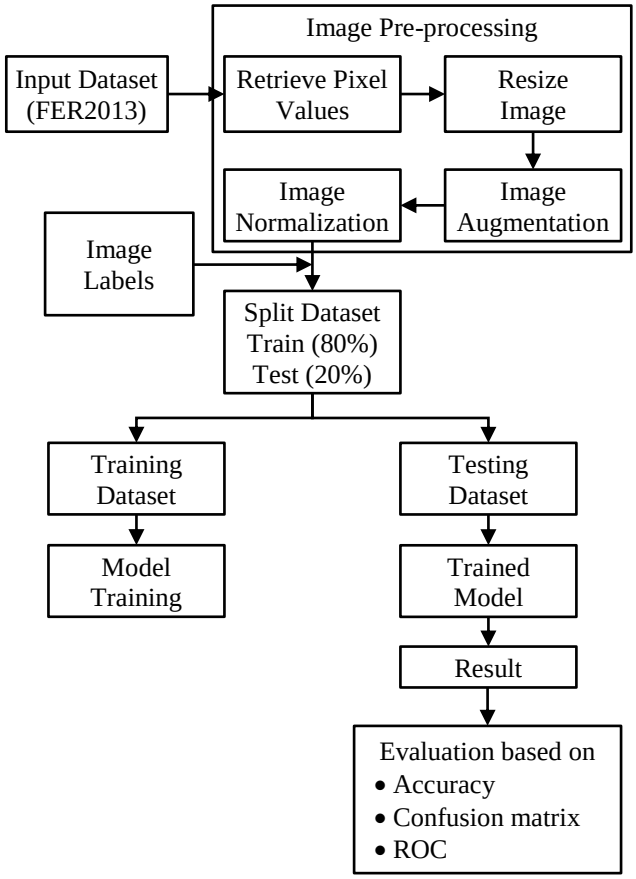


Fig.1. System Architecture

The various components of the system are described in more depth below.

3.1 INPUT DATASET

There are several datasets available for facial expression, out of those in this experiment FER2013 dataset IS used. The dataset contains images of dimensions 48×48×1. The Table.1 provides more information about the dataset. The classes and number of images per class are provided. The Fig.2 shows the visual representation of the data

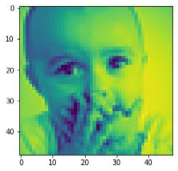
Fig.2. Visual Representation of the Sample Dataset

Table.1. List of Classes and number of images in FER2013 dataset

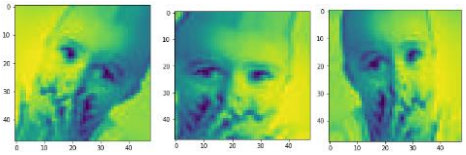
Class	Number of Images
Anger	4953
Fear	5121
Happy	8989
Sad	6077
Surprise	4002
Neutral	6198

3.2 IMAGE PRE-PROCESSING

The FER2013 dataset is a CSV file that consists of pixel values and labels of each image. Initially, the pixel value is gained and kept in an array. Simultaneously, the label of each image is kept in another array. Even Though all images are of the same size 48×48×1. To prevent any issues in the future, all the images are converted to the same size of 48×48×1. Data normalization is an important step that ensures that each input parameter has a similar data distribution. This makes convergence faster while training the network. In our case, the images are normalized by dividing each pixel value by 255.0. It changes the range of pixel value from (0,255) to (0,1). It is a technique for increasing the size of a training dataset artificially by producing modified versions of the photos in the dataset. In our experiment, the images are rotated, shifted, zoomed, sheared, and shifted. The Fig.3 shows an example of augmentation in images during training.



(a)



(b)

Fig.3. Image Augmentation (a) Sample Image (b) After Augmentation

3.3 MODEL

VGG19 based model is used in this research. Beside Alexnet [7], VGG19 [8] and Resnet50 [9] is also used, but both of them provides the very bad result. So, those model results are not taken for further evaluation. The small and simple AlexNet model is unable to capture the complex information and the RESNET50 model due to higher complexity losses most information as the

size of the model increases, there is a loss of information. The Table.2 is the architecture of the VGG19 based model used in the experiment.

Table.2. VGG19 based CNN Architecture

Layer (type)	Output Shape
Conv2D	(None, 48, 48, 1)
Conv2D	(None, 48, 48, 64)
MaxPooling	(None, 24, 24, 64)
Conv2D	(None, 24, 24, 128)
Conv2D	(None, 24, 24, 128)
MaxPooling	(None, 12, 12, 128)
Conv2D	(None, 12, 12, 256)
Conv2D	(None, 12, 12, 256)
Conv2D	(None, 12, 12, 256)
MaxPooling	(None, 6, 6, 256)
Conv2D	(None, 6, 6, 512)
Conv2D	(None, 6, 6, 512)
Conv2D	(None, 6, 6, 512)
MaxPooling	(None, 3, 3, 512)
Conv2D	(None, 3, 3, 512)
Conv2D	(None, 3, 3, 512)
Conv2D	(None, 3, 3, 512)
MaxPooling	(None, 1, 1, 512)
Flatten	(None, 512)
Dense	(None, 6)

3.3.1 Optimizer:

Optimizers are techniques or strategies for changing the characteristics of neural networks, such as weights and learning rate, to minimize losses. Five optimizers are used in this study. The optimizers are ADAM [10], AdaDelta [11], RMSProb [12], AdaGrad [13] and SGD [14].

- ***SGD:*** Gradient Descent is a variation of this game. It tries to update the parameters of the model more often. After each loss in training example has been computed, the model parameters are changed. Because model parameters are regularly changed, loss functions contain a lot of variation and fluctuation at different intensities. The Eq.(1) is the mathematical representation of the optimizer.

$$\theta = \theta - \alpha.\nabla J(\theta; x(i); y(i)) \tag{1}$$

where $\{x(i), y(i)\}$ are the training examples.

- ***AdaGrad:*** The learning rate is constant for all parameters and for each cycle, which is one of the drawbacks of all the optimizers discussed. The learning rate is altered by this optimizer. It adjusts the learning rate η for each parameter and time step t. It is a second-order optimization method of the kind. It is based on an error function derivative. At time step t, $g_{t,i}$ is the partial derivative of the objective function regarding the parameter θ_i. The mathematical representation is shown below.

$$g_{t,i} = \theta - \alpha.\nabla_{\theta} J(\theta_{t,i}) \tag{2}$$

At a given time t, the derivative of the loss function for given parameters is represented by Eq.(3).

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,i} + \varepsilon}} \cdot g_{t,i} \tag{3}$$

where η is a learning rate that is changed for a given parameter $\theta(i)$ at a particular time based on previously determined gradients for that parameter $\theta(i)$.

- ***AdaDelta:*** It is an AdaGrad extension that aims to solve the declining learning rate problem. Adadelta restricts the window of collected past gradients to some defined size w, rather than accumulating all previously squared gradients. Rather than the total of all the gradients, an exponential moving average is utilized in this case. The mathematical equation of the running average and update parameter is represented by Eq.(4) and Eq.(5) respectively.

$$E\left[g^2\right]_t = \gamma E\left[g^2\right]_{t-1} + (1-\gamma) g_t^2 \tag{4}$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E\left[g^2\right]_t + \varepsilon}} \cdot g_t \tag{5}$$

- ***RMSProb****:* The RMSprop optimizer is a momentum-based version of the gradient descent method. The RMSprop optimizer limits oscillations in the vertical plane. As a result, we may raise our learning rate, allowing our algorithm to take greater horizontal steps and converge faster. The way the gradients are calculated differs between RMSprop and gradient descent. The RMSprop is computed using the Eq.(6) and Eq.(7).

$$E\left[g^2\right]_t = 0.9 E\left[g^2\right]_{t-1} + 0.1 g_t^2 \tag{6}$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E\left[g^2\right]_t + \varepsilon}} \cdot g_t \tag{7}$$

- ***Adam:*** Adam (Adaptive Moment Estimation) works with first- and second-order momentums. Adam's intuition is that wess don't want to roll too rapidly merely to leap over the minimum; instead, we want to slow down a little to allow for a more deliberate search. Adam retains an exponentially decaying average of previous gradients M in addition to an exponentially decaying average of past squared gradients like AdaDelta. $M(t)$ and $V(t)$ are the values of the first and second moments, respectively, the Mean and the uncentered variance of the gradients.

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{8}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{9}$$

We are using the average of $M(t)$ and $V(t)$ here such that $E[m(t)] = E[g(t)]$, where $E[f(x)]$ is the anticipated value of $f(x)$. To update the parameter Eq.(10) is used.

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \varepsilon}} \cdot \hat{m}_t \tag{10}$$

3.4 MODEL EVALUATION

One of the many metrics that can be used to solve classification difficulties. Three measures are utilized to evaluate model performance in this work. These figures are described further down.

3.4.1 Accuracy:

It is a critical statistic for assessing model performance in classification tasks. Our model's correct observation rate is expressed as a percentage. The mathematical expression of accuracy is Eq.(11).

$$Accuracy = (Number\ of\ correct\ predictions)/(Total\ number\ of\ predictions) \quad (11)$$

3.4.2 Confusion Matrix:

In comparison to the accuracy confusion matrix, the results are more detailed. It provides accuracy for each class. By comparing the actual and target labels, the accuracy per class is computed. The total accuracy of the model can also be measured using the confusion matrix. The confusion matrix is depicted in Fig.5 as a general summary.

Predicted Class	**Actual Class**		
		Positive	Negative
	Positive	*TP*	*FP*
	Negative	*FN*	*TN*

Fig.5. Confusion Matrix

A Confusion Matrix is made up of four parts. These are detailed further down.

- Both the actual and predicted classes are correct when True Positive is used.
- The model predicts the actual value, but the actual value is negative.
- The term False Positive (*FP*) refers to a situation in which the model predicted a false value and the actual value was also negative.
- The symbol *FN* stands for False Negative, which means the projected value is false while the actual value is positive.

3.4.3 ROC Curve:

The true positive rate (TPR) is displayed against the FPR in the Receiver Operator Characteristics (ROC) graph (FPR). An AUC value may be calculated from the graph. Between 0.5 and 1, the range is. The model with a value of 0.5 fails to discriminate between classes, but the model with a value around 1 succeeds. It demonstrates the model's capacity to differentiate across classes.

3.4.4 K Fold Validation:

Cross-validation is a resampling technique for evaluating machine learning models on a small sample of data. The process includes only one parameter, *k*, which specifies the number of groups into which a given data sample should be divided. As a result, the process is frequently referred to as *k*-fold cross-validation. When a precise value for *k* is specified, it may be substituted for k in the model's reference, for example, *k*=10 for 10-fold cross-validation.

Cross-validation is a technique used in applied machine learning to evaluate a machine learning model's competence on unknown data. That is, to use a small sample to assess how the model will perform in general when used to generate predictions on data that was not utilized during the model's training.

In our research, we use *K*-Fold validation for VGG19 models trained in the FER2013 dataset. *K* = 5 folds is used for validation. The accuracy, losses, and models after 5 folds are saved. Based on the lowest loss, weights are chosen for further analysis.

4. RESULTS

Initially, three models are taken for analysis with the dataset. The Table.3 shows the accuracy of these models. Adam is used to testing the performance of all the models. The AlexNet based model gained the least accuracy and the VGG19 has the highest. So, the VGG19 based model is used for further analysis.

Table.3. Accuracy of Classifier

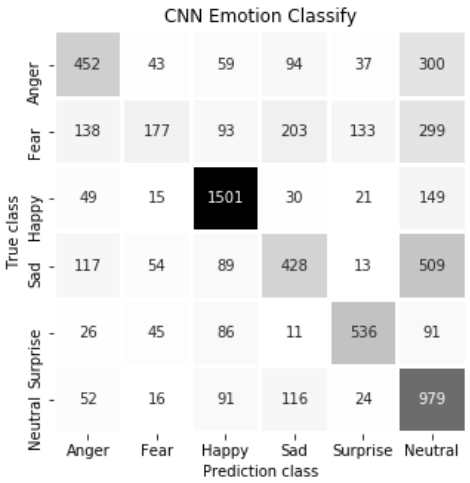
Model	**Training Accuracy**	**Testing Accuracy**	**Time (s)**
AlexNet	74.8%	75.4%	1500
VGG19	88.184%	88.1786%	4745.8
ResNet50	83.361%	85.82%	2473

The VGG19 based model is trained with FER2013 using 5 different optimizers. All the models are evaluated based on the accuracies and the final best is evaluated using confusion matrix and ROC Curve. K-fold validation is done using the best model and the models are saved in each split. The model with the least loss is further tested with testing data. The whole process and gained outputs are described below.

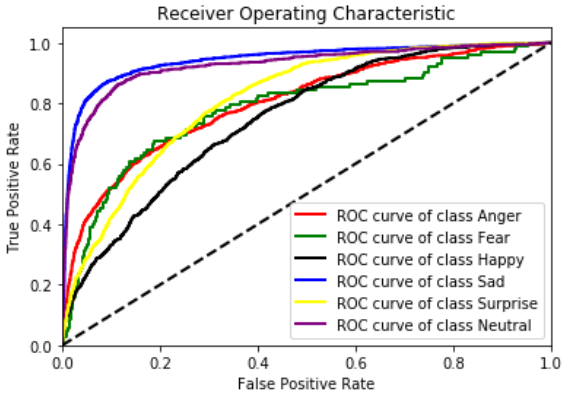
Table.4. Training and Validation Accuracy, FER2013 dataset, and VGG19 Model

Optimizers	**Training Accuracy**	**Testing Accuracy**	**Time (s)**
SGD	89.6337%	90.04717%	3904.2
AdaDelta	91.820%	91.895%	5238.2
AdaGrad	91.846%	91.9526%	4303.7
RMSProb	88.153%	88.117%	4448.6
ADAM	88.184%	88.1786%	4745.8

From Table.4, it can be seen that AdaDelta and AdaGrad model performs the best in terms of accuracy. The confusion matrix, and ROC Curve for the best models are shown in Fig.7 and Fig.8.



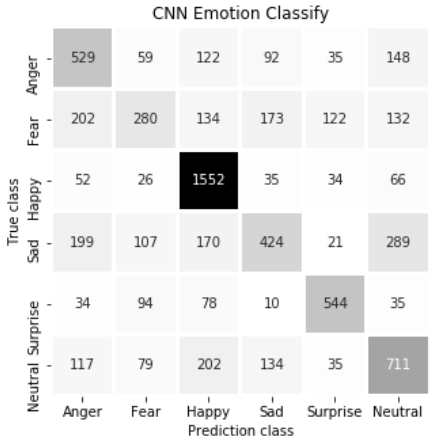
(a) Confusion Matrix



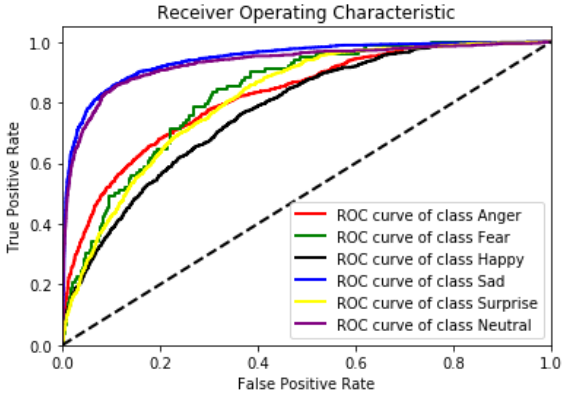
(b) ROC

Fig.7. AdaDelta Graphs

The adadelta model has optimal accuracies and the model can classify images based on different emotions which can be seen from the confusion matrix. The ROC Curve AUC value is high for both models which signifies that the models were able to distinguish among classes. The adagrad based model has a slightly better performance concerning the confusion matrix and ROC Curve, but the adadelta has better accuracies. The confusion clearly shows that the AdaGrad model can classify emotions more efficiently. The ROC Curve also clearly shows that the model can distinguish all the emotions clearly. Thus, the AdaGrad model is used for further analysis.



(a) Confusion Matrix



(b) ROC

Fig.8. AdaGrad Graphs

To evaluate the model in depth *K*-fold, validation is done for both models. A *k* value of 5 is used for validation and the values gained among the validations are shown in Table.5. The model with the lowest loss is taken.

Table.5. K-Fold Validation of Best Model

Model	Folds	Training Accuracy	Testing Accuracy	Time (s)
VGG19	1	88.0798%	91.895%	9207.6
	2	88.1705%		
	3	88.17509%		
	4	91.82077%		
	5	89.17056%		

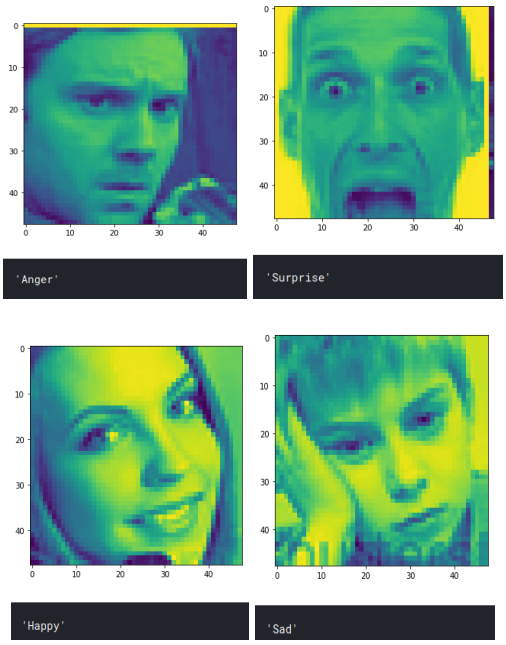


Fig.9. Test Image Sample and Model Output

Even Though the K-fold validation is only used for evaluation purposes. The model gained after each validation is saved and the model with the least loss is evaluated with the test dataset and testing accuracy is gained. Few images are passed through the model to check the model performance. The Fig.9 shows the images and model output.

The Table.5 shows the comparison of the model performance with other researchers' works for the FER2013 dataset.

Table.6. Comparison of proposed model with previous researcher model

Methods	Accuracy
Stochastic Optimization	71.10%
Online Learning	65.3%
Convolutional Neural Network	66%
Multi-Level CNN	73.03%
Ensemble of Multi-Level CNN	65.97%
Extreme Learning Machine	62.7%

Attention Mechanism	74%
Proposed Work	**91.89504%**

5. CONCLUSION

The experiment shows that among all the proposed models, the VGG19 model performs the best. VGG19 based CNN model can classify the images based on the facial emotion with state of art accuracy. The gained model is better than past models by at least 17% accuracy. The confusion matrix, ROC Curve and K-fold also show that the model performance is optimal. Furthermore, the model accuracy can be increased by using transfer learning and complex feature extraction techniques.

REFERENCES

[1] B.C. Ko, "A Brief Review of Facial Emotion Recognition based on Visual Information", *Sensors*, Vol. 18, No. 2, pp. 401-421,2018.

[2] H.D. Nguyen, S. Yeom, G.S. Lee, H.J. Yang, I.S. Na and S.H. Kim, "Facial Emotion Recognition using an Ensemble of Multi-Level Convolutional Neural Networks", *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 33, No. 11, 2019.

[3] Y.K. Bhatti, A. Jamil, N. Nida, M.H. Yousaf, S. Viriri and S.A. Velastin, "Facial Expression Recognition of Instructor using Deep Features and Extreme Learning Machine", *Computational Intelligence and Neuroscience*, Vol. 2021, No. 1-14, 2021.

[4] Ben Niu, Zhenxing Gao and Bingbing Guo, "Facial Expression Recognition with LBP and ORB Features", *Computational Intelligence and Neuroscience*, Vol. 2021, pp. 1-16, 2021.

[5] J. Daihong, Hu Yuanzheng, D. Lei and P. Jin, "Facial Expression Recognition Based on Attention Mechanism", *Scientific Programming*, Vol. 2021, pp. 1-18, 2021.

[6] L. Zahara, P. Musa, E. Prasetyo Wibowo, I. Karim and S. Bahri Musa, "The Facial Emotion Recognition (FER-2013) Dataset for Prediction System of Micro-Expressions Face using the Convolutional Neural Network (CNN) Algorithm based Raspberry Pi", *Proceedings of 5th International Conference on Informatics and Computing*,pp. 1-9, 2020.

[7] A. Krizhevsky, I. Sutskever and G.E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", *Communications of the ACM*, Vol. 25, No. 6, pp. 1097-1105, 2012.

[8] Simonyan Karen and Zisserman Andrew, "Very Deep Convolutional Networks for Large-Scale Image Recognition", *Proceedings of 3rd International Conference on Learning Representations*, pp. 1-6, 2015.

[9] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770-778, 2016.

[10] D.P. Kingma and J.L. Ba, "Adam: A Method for Stochastic Optimization", *Proceedings of 3rd International Conference on Learning Representations*, pp. 145-156, 2015.

[11] Matthew D. Zeiler, "Adadelta: An Adaptive Learning Rate Method", *Proceedings of 3rd International Conference on Learning Representations*, pp. 171-178, 2012.

[12] Yann Dauphin, "RMSProp and Equilibrated Adaptive Learning Rates for Non-Convex Optimization", *Proceedings of 3rd International Conference on Learning Representations*, pp. 340-345, 2015.

[13] J. Duchi, E. Hazan, and Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization", *Journal of Machine Learning Research*, Vol. 12, pp. 2121-2159, 2011.

[14] J. Kiefer and J. Wolfowitz, "Stochastic Estimation of the Maximum of a Regression Function", *The Annals of Mathematical Statistics*, Vol. 23, No. 3, pp. 462-466, 1952.

[15] Kaggle Dataset, Available at: https://www.kaggle.com/deadskull7/fer2013, Accessed at 2020.