International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:15, No:1, 2021

Facial Emotion Recognition with Convolutional Neural Network Based Architecture

Koray U. Erbas

***Abstract*—**Neural networks are appealing for many applications since they are able to learn complex non-linear relationships between input and output data. As the number of neurons and layers in a neural network increase, it is possible to represent more complex relationships with automatically extracted features. Nowadays Deep Neural Networks (DNNs) are widely used in Computer Vision problems such as; classification, object detection, segmentation image editing etc. In this work, Facial Emotion Recognition task is performed by proposed Convolutional Neural Network (CNN)-based DNN architecture using FER2013 Dataset. Moreover, the effects of different hyperparameters (activation function, kernel size, initializer, batch size and network size) are investigated and ablation study results for Pooling Layer, Dropout and Batch Normalization are presented.

***Keywords*—**Convolutional Neural Network, Deep Learning, Deep Learning Based FER, Facial Emotion Recognition

I. Introduction

AS technology in robotics and artificial intelligence (AI) advances, enhanced Human-Robot interaction seems to be an inevitable need for the future. The existing performance of this interaction could only be improved based on Human-Human interactions. During communication, according to surveys in [1] and [2], the total meaning in a message is roughly 7% verbal, 38% vocal, and 55% facial. Considering the huge impact of facial massages, recognizing facial expressions/emotions correctly (at least as good as humans can achieve) would have the primary impact on enhancing Human-Robot interaction.

With or without intention, humans are capable of making changes in the facial muscles. Studies [3], [4] show that facial changes could be grouped into three categories; Basic Emotions (BEs), Compound Emotions (CEs) and Micro Expressions (MEs). We believe that recognizing these emotions has practical benefits on building effective communication with autistic [5] and speech-impaired people. Furthermore, it is one of the important milestones for the humanization of the robots and has a huge potential for further researches such as determining mental disorder cases, surveillance and behavioral analysis, combined classification tasks (gender, ethnicity etc. classification), and sleep detection for drivers etc.

Deep Learning is a useful tool for learning complex and non-linear relationships between the input-output. With the advent of deep learning-based libraries and toolboxes (Tensorflow, Keras, Pytorch, MATLAB Deep Learning Toolbox etc.), designing complex neural network architectures and achieving higher accuracy rates is easier compared to conventional machine learning approaches. Despite its appeal; depending on the problem, finding a compatible dataset and selecting appropriate hyperparameters for the DNN structure is quite challenging. So that, comprehensive literature survey should be conducted in order to choose a generalized dataset and a number of experiments should be performed to adjust hyperparameters.

In this work, BEs (Happiness, Surprise, Anger, Sadness, Fear, Disgust, and Neutral) are classified from 48x48 gray scale images in the FER-2013 dataset using a CNN-based structure. In the rest of this section, we summarize the main researches in FER and a brief introduction about the CNNs is given. In Section II, facial emotion datasets are mentioned, and the proposed network architecture is introduced. In Section III, the experiment results are provided. Ablation study results and different hyper-parameter's effects are also presented in Section III.

II. Related Work

In conventional approaches, geometric and appearance features are extracted from detected face region. After that, commonly used multiclass machine learning algorithms are adopted for the final step of classification. For example, in [6], the authors propose to use AdaBoost for the selection of features which are Euclidean distance and angle of geometric features (52 facial landmarks' position) and Support Vector Machines (SVMs) as a classifier. Another approach [7] is extracting local features with Local Binary Patterns (LBP) and then applying Principal Component Analysis (PCA) based classification. LBP-based feature extraction method is used owing to its excellent light invariance property and low computational complexity [8]. Furthermore, in [9]-[11] Gabor Filter-based feature extraction is proposed for FER. In order to achieve higher accuracy rates, hybrid solutions are also offered such as in [12], [13].

In conventional methods, features and classifiers should be determined by the experts, because objective based specific information is needed in order to achieve useful accuracy rates in the test data. On the other hand, conventional approaches are advantageous since they enable lower computing power and memory than deep learning-based approaches [3].

Deep learning-based approaches are popular in many Computer Vision problems and provide good solutions when large data samples are fed to the network. Various architectures of DNNs and datasets are also proposed for

Koray U. Erbas is a Bachelor's Degree student in Kocaeli University in Electrical and Communication Engineering, (phone: +90 5535110389; fax, e-mail: koray24938@utexas.edu).

Facial Emotion Recognition. In [14], 57.1% accuracy rate is achieved with an FER 2013 dataset by adopting CNN and Fully Connected Layer-based model. With the same dataset and similar structure 67.12% accuracy rate is reported in [15] and comprehensive study about networks' hyperparameters is presented. After training the model with FER 2013 and testing with RAFD dataset, 71% accuracy rate is acquired in [16]. Higher test scores are also available in literature [17], [18] when working on other datasets.

Unlike conventional methods, deep-learning algorithms extract optimal features with the desired characteristic and enable end-to-end solutions. Despite of its advantages, deep learning approaches require a large amount of data and processing capacity. For FER applications using the MMI dataset [19], conventional and deep learning-based approaches' average success rates are reported as 63.20% and 72.65% in [3].

III. Material and Methodology

A. Convolutional Neural Networks

CNN-based structures are mostly composed of convolutional layers, pooling layers and fully connected layers. Convolutional layers take the input image at the beginning of the network, then a filter (the number of filters and size of the filter is defined as hyper parameter) scan of the pixels and extract feature maps. Pooling reduces the number of parameters in the feature maps by maintaining the useful ones. Pooling, which is also called subsampling, also helps avoid from overfitting and high computational workload for complex architectures. Fully connected layers flatten the output of previous convolutional layer and turns them into a single vector, then carry the features to the classification layer. Typical CNN structure is presented in Fig. 1.

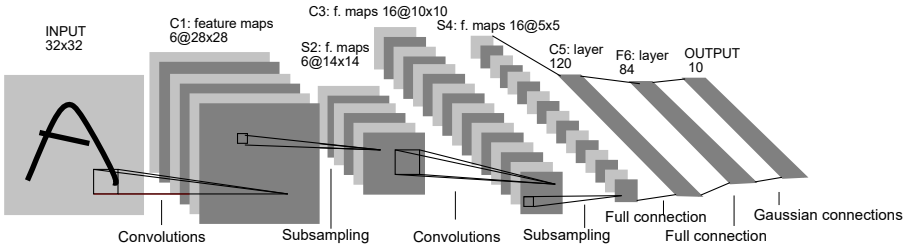


Fig. 1 Typical CNN Model (LeNet-5 [20])

Adopting convolutional layers in the Computer Vision applications has certain advantages:

- Less pre-processing is needed compared to other classification algorithms,
- Visual (Receptive) Field helps extract spatial patterns from the image more efficiently,
- As the number of layers increases, high-level features can be extracted easily (see Fig. 2 for illustration),
- Convolution operation with different filters helps extract more than one feature map in the same layer,
- Due to the weight sharing idea, the number of parameters is reduced compared to solely fully connected structures,
- Utilizing different kinds of techniques makes it easy to deal with overfitting and gradient blow-up.

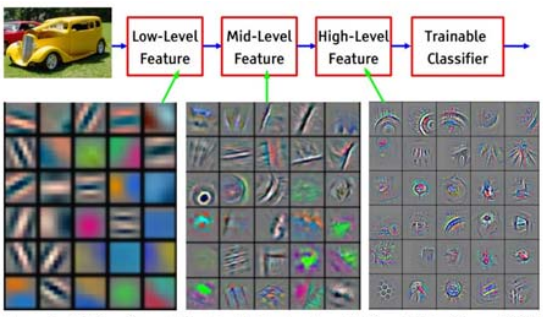


Fig. 2 Feature Visualization of CNN Cascaded Layers on Image Net [21]

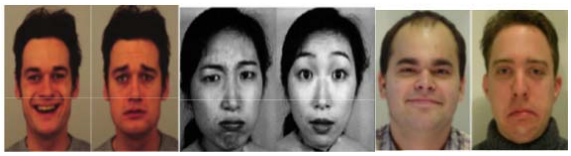
B. Dataset

In the literature various kinds of datasets are adopted for Emotion Recognition task. Each one has a different quantity and consists of samples of different quality, resolution, ethnicity, age, angle of portrait etc. Brief introduction about the commonly used datasets is provided below:

- JAFFE dataset [22] has 213 images for seven emotions and images are 256x256 pixels of Japanese female models.
- The Karolinska Directed Emotional Face (KDEF) has 4900 colorful images with resolution of 562×762. The set contains 70 individuals, each displaying seven different emotional expressions, each expression being photographed (twice) from five different angles [23].
- Radboud Faces Database (RaFD) is a set of high-quality pictures of 67 models (including Caucasian males and females, Caucasian children, both boys and girls, and Moroccan Dutch males) displaying eight emotional expressions. It provides well balanced labels [24].
- The Extended Cohn-Kanade (CK+) Dataset has 593 video sequences of 123 individuals with ages 18-30. The samples have 640x480 resolution [25].
- The MMI database consists of over 2900 videos and high-resolution still images of 75 subjects. The original size of each facial image is 720x576 pixels [26].
- FER-2013 dataset [27] contains 35887 facial images with 48x48 pixels. Each pixel is labeled one out of seven classes, where the distribution of labels is unbalanced.

The samples of images are provided in Fig. 3.

(a)CK+ (b) RaFD



(c)KDEF (d) JAFFE (e) MMI

(f) FER-2013

Fig. 3 Sample Face Images from Various Datasets

Depending on the application, researchers choose the suitable dataset for their model problem; however, it is also possible to use more than one dataset. It is obvious that deep learning needs large amounts of high-quality samples reflecting the essence of the task. But the diversity is also important in order to generalize the problem at the same time for the test phase.

Since FER-2013 contains the biggest number of samples and has a large diversity (in terms of age, gender, ethnicity), we choose FER-2013 dataset both for training (90%) and testing (10%). Training the model with the FER-2013 dataset and testing on another dataset is also suggested in order to increase the accuracy rates [16], [28], [29]. Number of emotions per each class is given in Table I.

For the implementation, we have added 28 more samples to the dataset before splitting the dataset. The corresponding labels are given in Table II. A few of the additional samples are displayed in Fig. 4. To align these samples with the original FER-2013 dataset, we first detected the face by using Haar Cascades [30] from the OpenCV library in Python, then re-sized (48x48 pixels) and converted it to a grayscale image.

TABLE I
DISTRIBUTION PER EMOTION IN FER-2013

No	Label	Number	Percentage
0	Anger	4953	% 13.8
1	Disgust	547	% 1.52
2	Fear	5121	% 14.27
3	Happiness	8989	% 25.05
4	Sadness	6077	% 16.93
5	Surprise	4002	% 11.15
6	Neutral	6198	% 17.27

TABLE II
ADDITIONAL SAMPLES TO FER-2013 DATASET

No	Label	Number
0	Anger	3
1	Disgust	4
2	Fear	2
3	Happiness	7
4	Sadness	4
5	Surprise	3
6	Neutral	5

Fig. 4 Few of the Additional Samples Used

C. Proposed Architecture

The architecture proposed for this paper is shown in Fig. 5 and the details in layers are given in Table III. After each CNN Layer, Batch Normalization has been used and for the kernels (1,1) strides have been adopted.

D. Experimental Results

For the implementation we have used Google Colab with GPU support. The training is completed in 230 epochs. Each epoch run is approximately 13.5 seconds and 51 minutes in total. The weights are initialized with He-Normal Initialization. As an optimizer, Adam is used with 0.001 learning rate, β_1= 0.9, β_2= 0.999 and $\varepsilon = 10^{-5}$. For batch size, 256 is selected and the input data are shuffled. Regularization is also adopted with l_2 norm and with $4\,x\,10^{-3}$ penalty. In order to have a stable optimization learning rate, a reducer is used with 3 epoch patience and with a factor 0.9. Finally, 70.18% accuracy rate is achieved in the test phase. Model accuracy and loss for the training and testing phase is presented in Fig. 6 and the Confusion Matrix without normalization for the test dataset is provided in Fig. 7, where the horizontal axis is "predicted label" and the vertical axis is "true label".

For comparison, the final test results of the related architectures are presented in Table IV. In order to provide a fair comparison, only the researches of an FER-2013 dataset-based CNN models are considered. The added 28 sample's effect is neglected.

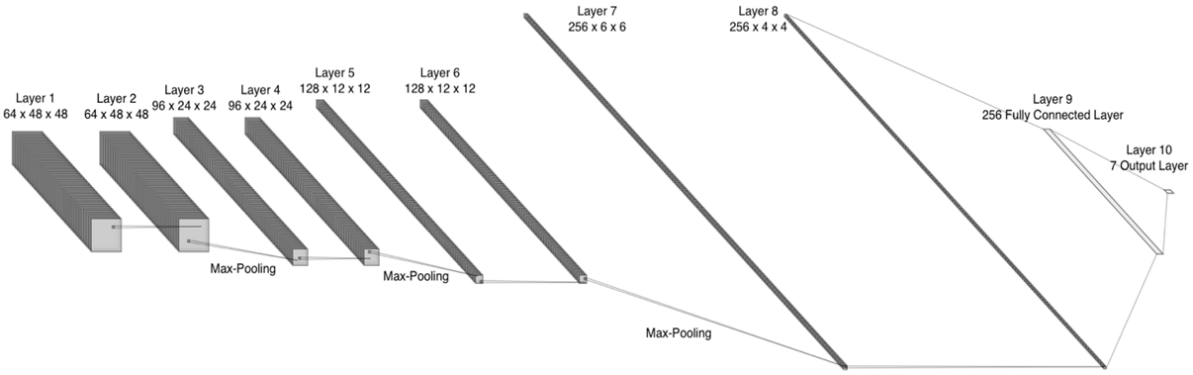


Fig. 5 The Proposed Architecture

TABLE III
ARCHITECTURE DETAILS FOR THE PROPOSED MODEL

Layer No.	Layer Type	Depth	Width	Height	Filter Size	Activation	Additional Operations
1	CNN	64	48	48	(3,3)	Leaky ReLU (0.3)	-
2	CNN	64	48	48	(3,3)	Leaky ReLU (0.3)	MaxPooling + Dropout (0.5)
3	CNN	96	24	24	(3,3)	Leaky ReLU (0.3)	-
4	CNN	96	24	24	(3,3)	Leaky ReLU (0.3)	MaxPooling + Dropout (0.5)
5	CNN	128	12	12	(3,3)	Leaky ReLU (0.3)	-
6	CNN	128	12	12	(3,3)	Leaky ReLU (0.3)	MaxPooling + Dropout (0.5)
7	CNN	256	6	6	(3,3)	Leaky ReLU (0.3)	Valid Zero Padding
8	CNN	256	4	4	(3,3)	Leaky ReLU (0.3)	MaxPooling + Dropout (0.5)
9	Dense	256	-	-	-	Swish	Dropout (0.5)
10	Dense	7	-	-	-	Softmax	-

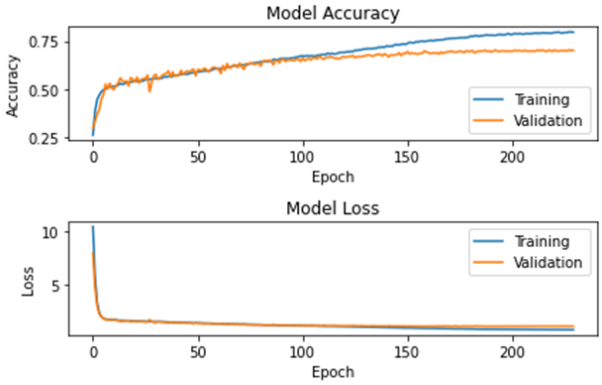


Fig. 6 Model Accuracy and Loss

TABLE IV
COMPARISON TABLE

Research	Test Accuracy
Tumen et al. [14]	57.1%
Salunke and Patil [16]	66%
Gudi and Vision [15]	67.12%
Proposed Model	70.18%

The accuracy rates in each class are presented in Table V. According to the accuracy rates and the Confusion Matrix results, we can infer that:

- The most confused classification is "Fear", where true-labeled "Fear" samples are misclassified as "Sadness",

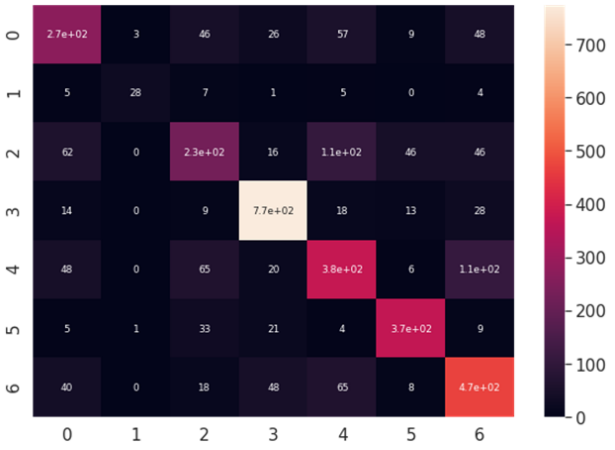


Fig. 7 Confusion Matrix

- Secondly, "Sadness" true-labeled samples are misclassified as "Neutral",
- "Happiness" and "Surprise" labels are correctly labeled with an accuracy rate 90.37% and 83.52%, respectively,
- The accuracy rates for "Fear", "Anger" and "Disgust" are low compared to other classes.

E. Ablation/Hyperparameter Study Results

During the development of the proposed model, we first define a baseline model and by tuning the hyper-parameters separately, we observe the difference between results of loss for both training and testing. We run the code for 100 epochs,

which is considered enough for validating the performance and checking for the overfitting situation. For the proposed network, we select the parameter values where the difference between test loss and training loss is minimum.

The baseline model is given in Fig. 8. In this section, certain hyperparameters' and techniques effects on the FER problem are presented.

(1) Pooling Layer

Pooling is a widely used concept in CNNs in order to down sample the feature map while preserving the features of the certain patches. In addition to helping the network to be more manageable, pooling also makes the applied CNN layer invariant to small changes in the location of feature map. There exist two common pooling techniques:

- Average Pooling: this assigns the average value of the patch on the following feature map.
- Max Pooling: this assigns the maximum value of the patch on the following feature map.

Since we do not want to lose certain characteristics of the feature map, we adopted Max Pooling rather than Average Pooling. The results of the experiment validate the intuition as seen in Table VI. Note that we have used stride (2,2).

TABLE V
ACCURACY RATES FOR EACH CLASS

Label No.	Label	Test Accuracy
0	Anger	58.82%
1	Disgust	56%
2	Fear	45.09%
3	Happiness	90.37%
4	Sadness	60.41%
5	Surprise	83.52%
6	Neutral	72.42%

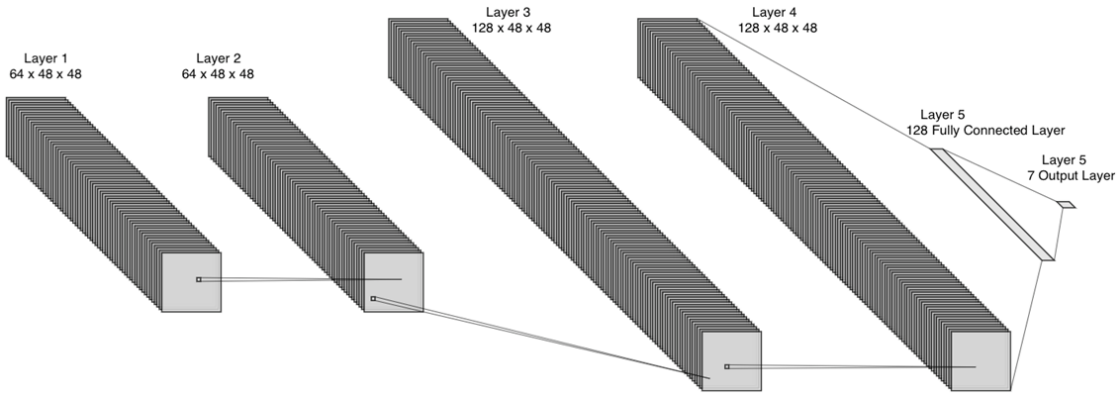


Fig. 8 Baseline Model Architecture

TABLE VI
EFFECT OF POOLING LAYER

Pooling	Training		Test	
	Acc (%)	Loss	Acc (%)	Loss
Baseline Model	99.71	0.004	49.64	4.186
Max Pooling	99.67	0.010	55.09	5.215
Average Pooling	99.66	0.022	54.34	4.771

TABLE VII
EFFECT OF DROPOUT

Dropout	Training		Test	
	Acc (%)	Loss	Acc (%)	Loss
Baseline Model	99.71	0.004	49.64	4.186
Dropout Ratio = 0.3	98.81	0.036	50.03	4.421
Dropout Ratio = 0.5	98.75	0.038	51.39	3.885

(2) Dropout

Dropout is a special type of regularization technique to prevent overfitting. With the selected parameter, randomly selected neurons are set to zero in each forward propagation during training. In the test phase, all the neurons are activated and in order to average out the randomness, the weights are multiplied by the selected parameter for the specified layer [31]. The results of Dropout implementation with two different dropout ratios are given in Table VII. As seen in the table, a 0.5 dropout ration works quite well.

(3) The Sequence of Pooling Layer and Dropout

Since we observe a positive effect of the Max Pooling Layer and Dropout with ratio 0.5, both techniques are applied together. In this sub-section, the order during implementation is investigated. As seen in Table VIII, the applying dropout after the pooling layer gives better results considering the overfitting gap.

TABLE VIII
EFFECT OF THE ORDER OF DROPOUT AND POOLING

Dropout/Pooling	Training		Test	
	Acc (%)	Loss	Acc (%)	Loss
Baseline Model	99.71	0.004	49.64	4.186
Dropout (0.5) + Maxpooling	77.28	0.62	60.18	1.1383
Maxpooling + Dropout (0.5)	62.81	0.990	60.71	1.937

(4) Batch Normalization

In order to reduce the effect of initialization and enable the network to have more stable gradient flow during the training phase, Batch Normalization is recommended first as in [32]. This technique also allows us to select higher learning rates and relatively helps the model to achieve a slightly faster

convergence. For a Batch Normalization applied layer, two more additional trainable parameters are adopted. The effect of this technique could be observed in Table IX. Note that depending on the previous section's result, the Baseline Model is upgraded.

TABLE IX
EFFECT OF BATCH NORMALIZATION

Batch Normalization	Training		Test	
	Acc (%)	Loss	Acc (%)	Loss
Baseline Model	62.81	0.990	60.71	1.937
Batch Normalization	66.75	0.888	62.31	1.042

(5) Initializer

Weight initialization is useful for avoiding exploding and vanishing gradients and enables the network to converge in early epochs. The effects of Xavier Glorot Initialization [33], Kaiming He Initialization [34] and Orthogonal Initialization [35] are given in Table X. As seen in the table, the results are very close to each other. Since we chose a large epoch number, the effect of initialization cannot be seen. Again, we revised the baseline model by adding Batch Normalization after the activation function by considering the previous section's results.

TABLE X
EFFECT OF INITIALIZER

Initializer	Training		Test	
	Acc (%)	Loss	Acc (%)	Loss
Baseline Model	66.75	0.888	62.31	1.042
Xavier Glorot Init.	66.70	0.887	63.27	1.030
He Uniform Init	67.79	0.860	63.35	1.033
Orthogonal Init.	67.86	0.860	63.61	1.013

(6) Kernel Size

For the convolution process, a filter slides over the pixels of the feature map/image and extracts features for the next layer. For the size of the filter, odd numbers are selected in order to center the kernel onto a pixel. Although selecting small kernels such as 3x3, which is a popular choice, we investigate the effect of the size as seen in Table XI. It seems that 5x5 gives the best result. However, by taking care of its overfitting gap, it seems choosing the 3x3 kernel is much more reasonable. In this section Orthogonal Initialization is adopted.

TABLE XI
EFFECT OF KERNEL SIZE

Kernel Size	Training		Test	
	Acc (%)	Loss	Acc (%)	Loss
Baseline Model (3x3)	67.86	0.860	63.61	1.013
(5x5)	74.81	0.721	65.45	1.021
(7x7)	76.37	0.632	64.83	1.039

(7) Activation Function

Activation function has a crucial effect on the Neural Networks, since it determines the output of each neuron through the net. Convergence ability and the speed of acquiring optimum parameters strongly depend on the activation function adopted. Although each has pros and cons, nowadays ReLU and its variations are commonly preferred due to their positive effect on training time. The comparison of the activation function for the FER problem is given in Table XII.

In this section, Kaiming He Initialization is adopted. Unlike from previous sections, in this study the fully connected layer is also included.

TABLE XII
EFFECT OF ACTIVATION FUNCTIONS

Activation Function	Training		Test	
	Acc (%)	Loss	Acc (%)	Loss
Baseline Model (ReLU)	67.86	0.860	63.61	1.013
LeakyReLU (0.3)	67.49	0.860	63,33	0.992
LeakyReLU (0.5)	63.54	0.971	63.11	1.01
Swish	64.61	0.945	62.58	1.011
PReLU	67.85	0.857	62.36	1.033

(8) Network Size

Intuitively in DNNs, a larger network means the better performance. However, depending on the problem and patterns in the input dataset, the extra width (number of neurons in layers) or depth (number of layers) might be redundant. In this section, we search for an answer of optimum network size for FER 2013 dataset. The comparison is given in Table XIII. As an activation function for the rest of the experiments in this section, Leaky ReLU with 0.5 is adopted.

TABLE XIII
EFFECT OF ACTIVATION NETWORK SIZE

Network Size	Training		Test	
	Acc (%)	Loss	Acc (%)	Loss
Baseline Model	65.83	0.905	63.25	0.995
Without FC Layer with 128 Neurons	62.4	1.007	62.08	1.025
Double the width of CNN layers and FCN	76.84	0.626	65.28	1.033
Quadruple the width of CNN layers and FCN	90.01	0.273	65.37	1.339
Double the depth of CNN (adding two layers for each 96 and 256 number of filters) *	69.95	0.807	65.45	0.954
Adding CNN layer with 512 number of filters to the previous row **	70.85	0.780	66.54	0.957

* Max Pooling applied at the end of CNN layers with even numbers. FCN changed to length of 256.

** Max Pooling applied at the end of CNN layers with even numbers. FCN changed to length of 512.

TABLE XIV
EFFECT OF BATCH SIZE

Batch Size	Training		Test	
	Acc (%)	Loss	Acc (%)	Loss
Baseline Model (256)	66.21	0.901	63.55	0.994
(128)	66.80	0.885	63.19	0.990
(512)	63.86	0.966	62.63	1

(9) Batch Size

Another hyper-parameter affecting the test results of the designed model is Batch Size. Using a large batch size might be advantageous in terms of speed if a GPU is used. On the

other hand, with a CPU, a smaller batch size enables faster convergence. The comparison is provided in Table XIV for different batch sizes.

IV. Conclusion and Discussion

In this paper, we introduced a CNN architecture where hyperparameters are tuned carefully to achieve higher accuracy rates. The proposed method works quite well without any pre-processing step and around 70% accuracy rate is achieved. Furthermore, we provide ablation study results for Pooling Layer, Dropout and Batch Normalization and observe different hyper-parameter's effect on the FER problem such as activation function, kernel size, initializer, batch size and network size.

We believe that Emotion Recognition is an important Computer Vision problem need to be solved for Human-Robot interaction. The main handicaps for real world applications are:

- The used datasets are labeled mostly by humans and relatively prone to label errors; for example, the human accuracy rate on FER-2013 data set is around 65±5% [36],
- Labels of the samples need to be assigned by psychological experiments, since FER 2013 dataset has ordinary people's face images that might reflect the emotions unseemly,
- There exist more than seven BEs in real life, more gestures need be considered, this would relatively increase the need for more labeled data.

For further work, real-time application of emotion recognition is to be considered. Since speech also carries the information about the emotion of humans, Speech Emotion Recognition and Face Emotion Recognition tasks could be merged and applied together.

References

[1] A. Mehrabian, A. Communication without words. Psychol. Today, 1968, pp. 53-56.
[2] A. Mehrabian, Silent Massages, 1971, pg.44.
[3] B.C. Ko, "A Brief Review of Facial Emotion Recognition Based on Visual Information," MDPI Sensor Journal, 18(2), 2018, pp. 401.
[4] S. Du, A.M. Martinez, "Compound facial expressions of emotion." Natl. Acad. Sci., 2014, pp. 1454-1462.
[5] B.H. Lee and J.G. Lee, "Therapeutic behavior of robot for treating autistic child using artificial neural network." Fuzzy Systems and Data Mining IV: Proceedings of FSDM, 2018, pp. 358-364.
[6] D. Ghimire, J. Lee, "Geometric feature-based facial expression recognition in image sequences using multi-class AdaBoost and support vector machines," Sensors 13(6), 2013, pp. 7714-7734.
[7] S.L. Happy, A. George, A. Routray, "A real time facial expression classification system using local binary patterns", In Proceedings of the 4th International Conference on Intelligent Human Computer Interaction (IHCI'12), 2012, pp. 1-5.
[8] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation Invariant texture classification with local binary patterns," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, 2012, pp. 971-987.
[9] G. K. Venayagamoorthy, S. Bashyal, "Recognition of facial expressions using Gabor wavelets and learning vector quantization," Engineering Applications of Artificial Intelligence, 2008, pp. 1056–1064.
[10] T. Ahsan, T. Jabid, and U.P. Chong," Facial expression recognition using local transitional pattern on gabor filtered facial images," IETE Technical Review, 30(1), 2013, pp. 47–52.
[11] S. Zhang, X. Zhao, B. Lei, "Robust facial expression recognition via compressive sensing," Sensors, 2012 pp. 3747–3761.
[12] D. Ghimire, S. Jeong, J. Lee, S.H. Park, "Facial expression recognition based on local region-specific features and support vector machines," Tenth International Conference on Digital Information Management (ICDIM'17) Multimed. Tools Appl., 2017, pp. 7803–7821.
[13] C.F. Benitez-Quiroz, R. Srinivasan, A.M. Martinez, "EmotioNet: An accurate, real-time algorithm for the automatic annotation of a million facial expressions in the wild," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16), 2016, pp. 5562–5570.
[14] V. Tumen O. Soylemez, B. Ergen, "Facial Emotion Recognition on a Dataset Using Convolutional Neural Network. International Artificial Intelligence and Data Processing Symposium (IDAP'17)," 2017. DOI: 10.1109/IDAP.2017.8090281
[15] A. Gudi, V. Vision, "Recognizing Semantic Features in Faces using Deep Learning," 2016 arXiv:1512.00743.
[16] V. Salunke, C.G. Patil, "A New Approach for Automatic Face Emotion Recognition," International Conference on Computing, Communication, Control and Automation (ICCUBEA'17), 2017. DOI: 10.1109/ICCUBEA.2017.8463785.
[17] A. Verma, P. Singh, J. Sahaya, R. Alex, "Modified Convolutional Neural Network Architecture Analysis for Facial Emotion Recognition," International Conference on Systems, Signals and Image Processing (IWSSIP' 19), 2019. DOI: 10.1109/IWSSIP.2019.8787215
[18] A. Mollahosseini, D. Chan, M.H. Mahoor, "Going deeper in facial expression recognition using deep neural networks," In Proceedings of the IEEE Winter Conference on Application of Computer Vision, 2019, pp. 1–10.
[19] MMI. Available online: https://mmifacedb.eu/ (accessed on 13 June 2020).
[20] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE. 86 (11), 1998, pp. 2278–2324.
[21] M.D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," ECCV 2014, Part I, LNCS 8689, 2014 pp. 818–833.
[22] M.J. Lyons, S. Akamatsu, M. Kamachi, J. Gyoba, "Coding facial expressions with Gabor wave," In Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition, 1998, pp. 200–205.
[23] KDEF Available online: http://www.emotionlab.se/resources/kdef (accessed on 13 June 2020).
[24] O. Langner, R. Dotsch, G. Bijlstra, D.H. Wigboldus, S.T. Hawk and A.van Knippenberg, "Presentation and validation of the radboud faces database," Cognition and emotion, 24(8), 2010, pp.1377-1388.
[25] P. Lucey, J.F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, I. Matthews, "The extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'10), 2010, pp. 94–101.
[26] MMI. Available online: https://mmifacedb.eu/ (accessed on 13 June 2020).
[27] Kaggle. Challenges in representation learning: Facial expression recognition challenge, 2013
[28] E. Correa, A. Jonker, M.Ozo, R. Stolk, "Emotion Recognition using Deep Convolutional Neural Networks," 2016.
[29] H.W. Ng, V.D. Nguyen, V. Vonikakis and S. Winkler, "Deep Learning for Emotion Recognition on Small Datasets Using Transfer Learning" ACM International Conference on Multimodal Interaction (ICMI'15), 2016.
[30] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," Conference on Computer Vision and Pattern Recognition (CVRP), 2001. DOI: 10.1109/CVPR.2001.990517
[31] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," Journal of Machine Learning Research 15, 2015, pp.1929-1958.
[32] S. Ioffe and C.Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015. arXiv:1502.03167.
[33] X. Glorot, Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," Conference on Artificial Intelligence and Statistics (AISTATS'10). Volume 9, 2010, pp. 249-256.
[34] K. He, X. Zhang, S. Ren, J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification"

2015. arXiv:1502.01852
[35] A. Saxe, J.L. McClelland, S. Ganguli, "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks," The International Conference on Learning Representations (ICLR14), 2014. arXiv:1312.6120
[36] I.J. Goodfellow, D. Erhan, P.L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D.H. Lee, Y. Zhou, C. Ramaiah, F. Feng, R. Li, X. Wang, D. Athanasakis, J. Shawe-Taylor, M. Milakov, J. Park, R. Ionescu, M. Popescu, C. Grozea, J. Bergstra, J. Xie, L. Romaszko, B. Xu, Z. Chuang and Y. Bengio, "Challenges in Representation Learning: A report on three machine learning contests" ICONIP 2013, Part III, LNCS 8228, 2013, pp. 117–124.