Recognizing Emotions by Analyzing Facial Expressions (April 2021)

Kaustubh Katkar, *University of Florida*

***Abstract*— This paper reports the performance of 4 phases of a Convolutional Neural Network for Facial expression detection. Facial expressions can be used to determine the emotions a person is feeling at a given moment. Using fer2013 dataset the model is trained and tested against unseen data with 2 different configurations. Each configuration is trained once with the original data and once with an additional data augmentation stage. Using Softmax activation we can observe the probability distribution of unseen data to identify overlapping features between emotions.**

***Index Terms*—Convolutional Neural Networks, Data Augmentation, Emotion recognition, Multi-label classification.**

I. Introduction

FACIAL expressions play an important role in conveying a person's emotion. An accurate identification of emotions can assist many domains significantly.

Mental health issues including PTSD and depression may be treated better by understanding the patient's response to a topic. As patients may not always want to share the personal events, it would serve the specialist to understand the patients thought process without having to investigate deeper.

It can be an effective indicator for physical exams as pain threshold may vary between subjects, but this can provide an additional metric for its evaluation.

In terms of organizations, emotions can be a valuable feedback. As emotions are a candid representation of a person's opinion, organizations can leverage a technology to recognize emotions and identify the practices and/or products that appeal to their customer base the most. Such responses can influence decisions and be a factor for assuring successful business operations.

Its application in human computer interaction is widely researched with attempts to develop humanoids or systems have a natural flow of conversation by understanding a person's response.

The CNN architecture presented in this report involves convolution layers with max pooling layer and added regularization of dropout layer. The training performance is evaluated with increase in data set using data augmentation.

.

II. Description

A. Dataset

This paper evaluates the performance of convolutional neural networks on the fer2013 [1] to learn the distribution the samples based on the facial expression. This dataset is divided into 3 attributes – emotion, pixels and Usage.

- emotion – These are the labels of the emotion for the given image. The label mappings are:

Table 1

Label to emotion mapping

label	emotion
0	Anger
1	Disgust
2	Fear
3	Happy
4	Sad
5	Surprised
6	Neutral

- pixel – This attribute contains the pixel values of each sample. Every record here is a string of spaced integers. Therefore, requires preprocessing.
- Usage – The dataset provides tags for Training, Validation and Testing

B. Model

A Convolutional Neural Network is built to extract, identify and learn the features from our input set. Convolutional Neural Networks (CNN) are fully connected feed forward networks. A major challenge with other neural network models is handling the increasing complexity resulting from the increase in the parameters at each layer making learning computationally expensive and slow. This is addressed in CNN by reducing the parameters passed between layers and identifying the principal features. The input is processed through filters at every layer which retain the information of the image. These filters are mathematical functions that convolve the input to smaller dimensions. Thus, the image representation is fully maintained between the layers.

In the project, the input to the model is an array of 48 x 48 dimensions for each sample. Besides the convolving operations Dense, MaxPool and Dropout layers complete the model. Dense layers are fully connected layer which performs

liner operations on the input. MaxPooling and DropOut are generalization layers which assist in avoidance of over-fitting [3].

This project uses the tensorflow.keras.layers module [2].

C. Activations

For the model, I have used ReLu activation function for the convolution layers and Softmax activation for the output layer.

i. ReLU – The Rectified Linear activation function returns a value if it is positive otherwise returns 0. This provides faster training for the neural network mand overcomes the Vanishing Gradient problem encountered by Sigmoid and TanH activation functions.

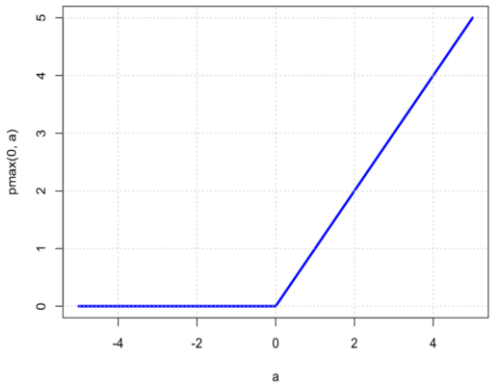

Fig 1. ReLU activation function

ii. Softmax – The decision of whether a sample belongs to one in a list of labels in our project should not retrieve an absolute value. This is because features between expressions overlap and the model should be able to identify and learn these overlapping features. For such multi-label classification Softmax activation provides the probability distribution between all possible labels.

$$Softmax(\vec{x_i}) = \frac{e^{x_i}}{\sum_k e^{x_k}}$$

D. Model parameters

The training model created can be used to configure losses and metrics. The Facial Expression Analyzer (FEA) model made in this project uses Adam optimizer to train itself with 0,001 learning rate. The choice of Adam optimizer for learning can be attributed to it adaptive learning rates to different parameters by using the first and average of second moments of gradients. Advantages of using Adam optimizer can be found here [4].

The loss used for our classifier is categorical cross entropy and the accuracy metric will determine its training performance.

Alternatively, KL-divergence loss also provides good performance but is computationally slow and unnecessary for a multi-label classification problem. It is commonly used in auto-encoders or GANs where the model attempts to regenerate an image. It is also computationally expensive.

E. Data Augmentation

We can improve the quantity and diversity of data by a significant extent for image classification models by using Data Augmentation techniques. Using the cropping, horizontal flipping, sheer range and such augmentation techniques, new data can be created for existing labels. Passing these inputs to the deep learning model also enables the network to identify image characteristics at angles and orientations it may not have encountered in training.

F. Evaluation Metrics

Accuracy score is used to determine the performance and using confusion matrix we can visually measure the difference between true labels and predictions.

III. Pre-processing data

The data set contains Integer labels, string of pixels and string of usage tags for training, validation and test splits. Through exploratory data analysis we can find that the dataset contains a total of 35,887 samples. These samples are tagged in the ratio 80:10:10 for train, validation and test sets respectively.

The string of pixels is reshaped into an array to be compatible to train the model. Using this array we can plot the image to get an idea of the data.

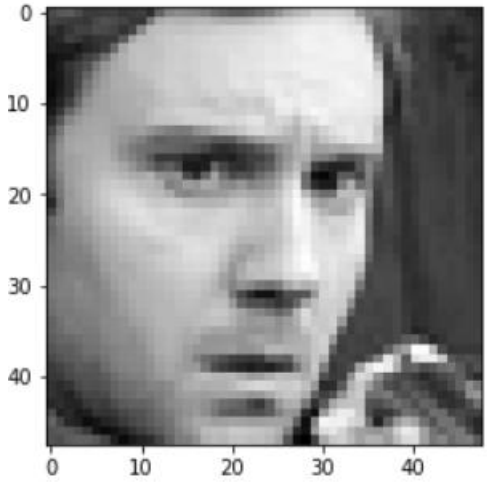

Fig 2. Angry labelled Image

Insights:

The low resolution of data indicates that the final trained FEA may not perform highly on test. However, performance comparisons between models can show improvements and therefore this dataset is good for evaluating competing models.

The data within the pixel array can be normalized to rescale the distribution between 0 and 1. By visualizing the

distribution of the data per label, more information and insights can be gained.

From the distribution it can be observed that the data for "Disgust" emotion is sparse in both training and validation sets. This will cause the model to misclassify "Disgust" labels as the data does not seem to be sufficient to train the model.

Therefore, the samples associated with "Disgust" emotion are dropped before training the model.

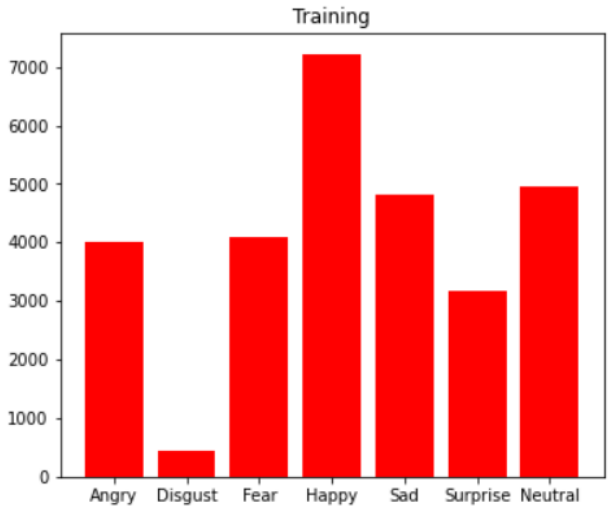

Fig 3. Label wise distribution of training samples

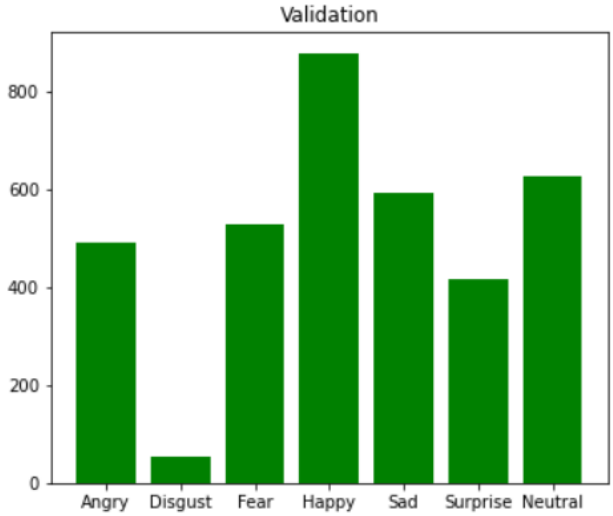

Fig 4. Label wise distribution of Validation samples

Besides dropping the "Disgust" class, distribution of the remaining classes is also imbalanced. To address this a list of class weights is calculated and passed as a parameter to the model's fit method.

IV. Data Augmentation

After training the model, we can create augmented data from the previous dataset and double the sample size of training and validation. From the preprocessing module in Keras the ImageDataGenerator class is utilized to achieve augmentation.

The augmentations performed on the fer2013 dataset:

Table 2
Augmentations

parameter	value
Shear_range	0.2
Rotation_range	30
Horizontal_flip	True

- Shear_range parameter changes the angle of the image causing a perspective shift.
- Horizontal_flip flips the image horizontally.
- rotation_range randomizes rotation for sample.

There are many configuration options provided in the ImageDataGenerator library which can be found here [6]. Applying drastic augmentations on this data will distort the images.

Fig 5. Images from augmented data

The facial region for the augmented images is not distorted ensuring that augmented data is useable for training the model further.

V. Building the Model

The neural network built is sequential with alternating convolutional and MaxPooling layers. Two configurations of models were built so that we can evaluate their performances.

Furthermore, the models were saved before augmentation and after augmentation. These models were independently tested over unseen data.

The model parameters were kept similar to maintain a controlled environment.
Parameter used:
Epochs = 100 (Later configured for early stopping)
Batch size = 64

Model for 1st Run:

Model: "sequential_1"

Layer (type)	Output Shape	Param #
conv2d_3 (Conv2D)	(None, 46, 46, 32)	320
max_pooling2d_2 (MaxPooling2	(None, 23, 23, 32)	0
conv2d_4 (Conv2D)	(None, 21, 21, 64)	18496
max_pooling2d_3 (MaxPooling2	(None, 10, 10, 64)	0
conv2d_5 (Conv2D)	(None, 8, 8, 64)	36928
flatten_1 (Flatten)	(None, 4096)	0
dense_2 (Dense)	(None, 64)	262208
dense_3 (Dense)	(None, 7)	455

Total params: 318,407
Trainable params: 318,407
Non-trainable params: 0

Fig 6. Model for 1st run of augmented and original data.

Model for 2nd Run:

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 46, 46, 32)	320
max_pooling2d (MaxPooling2D)	(None, 23, 23, 32)	0
dropout (Dropout)	(None, 23, 23, 32)	0
conv2d_1 (Conv2D)	(None, 21, 21, 64)	18496
max_pooling2d_1 (MaxPooling2	(None, 10, 10, 64)	0
dropout_1 (Dropout)	(None, 10, 10, 64)	0
conv2d_2 (Conv2D)	(None, 8, 8, 64)	36928
flatten (Flatten)	(None, 4096)	0
dense (Dense)	(None, 64)	262208
dense_1 (Dense)	(None, 7)	455

Total params: 318,407
Trainable params: 318,407
Non-trainable params: 0

Fig 7. Model for 2nd run of augmented and original data.

The input to these models will have the shape (28273, 48, 48, 1) where 28273 are the number of input samples, 48x48 are the image dimensions and 1 is for grayscale.

VI. Evaluation

Model performance with 100 Epochs:

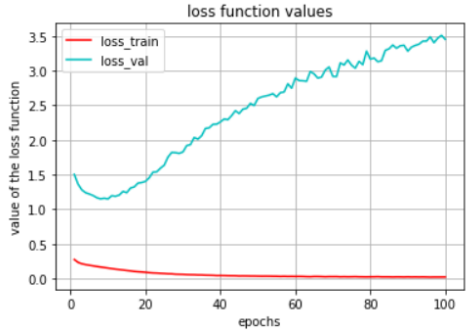

Fig 8. Training vs Validation Loss

We can see the model overfitting to the data as validation loss increases. Implementing Early Stopping at epoch 20 for the models to avoid overfitting.

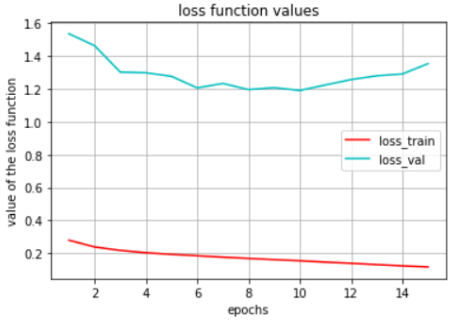

Fig 9. Training vs Validation Loss (Early Stopping)

1st Model (original data):

With the original data the first model had a training accuracy of 97% but had and accuracy of only 51% on unseen test data. This disparity in the performance is indicative of the model overfitting the dataset.

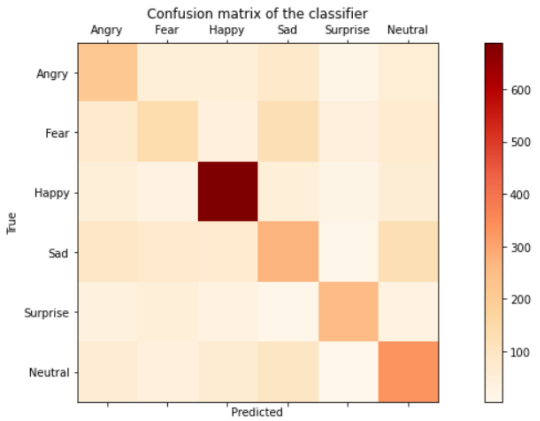

Fig 10. Confusion matrix of model1-original data

1st Model (original data + augmented data):

On training with augmented images the model's training accuracy went down to 65%. Its accuracy with the test data was 58.8% which is an improvement over the original data.

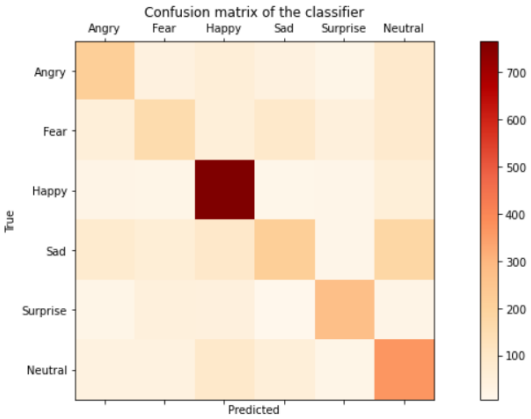

Fig 11. Confusion matrix for model1- generated data

The confusion matrices in Fig 8 and Fig 9 show that the emotions "Surprise" and "Happy" are most accurately predicted. There is ambiguity in predicting "Fear" as it gets classified equally as "Sad". This can be attributed to the overlapping features of these two emotions.

To address the overfitting which was evident in the 1st model I added Dropout layers. The Dropout layer randomly sets the input units to 0.

2nd Model (original data):

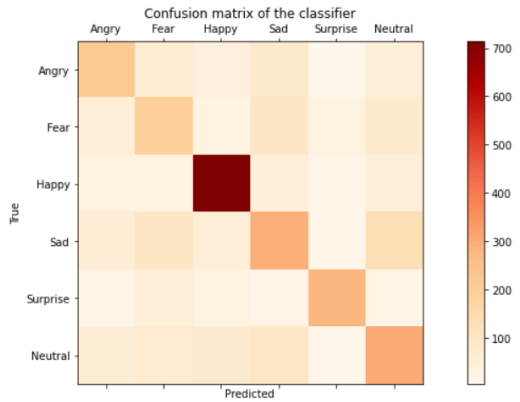

Fig 12. Confusion matrix for model2- original data

Model_2 continues the trend of predicting "Happy" most accurately. We can observe an improvement in recognizing "Fear" in comparison to Model_1. Sad is less frequently misclassified as "Fear". An improved accuracy of 56.80% is achieved on test data, which in an improvement on both the previous model and is due to the additional regularization provided by Dropout layer.

2nd Model (original data + augmented data):

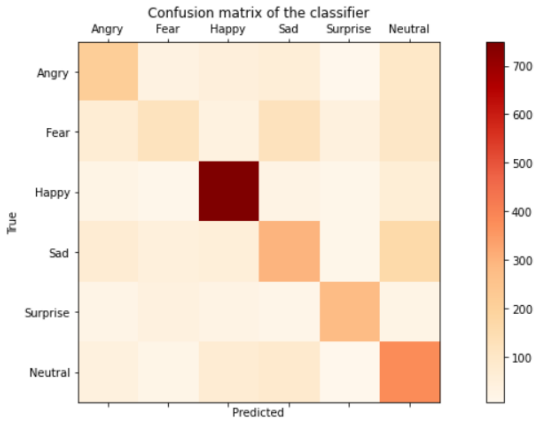

Fig 13. Confusion matrix for model2- generated data

Again, the trends are similar to the three previous models but this model again increases misclassification between Fear and Sad emotions. It performs well on other emotions. An accuracy score of 57.91% is observed which is and improvement on all the previous models.

Table 3

Evaluation metrics

Model	Train accuracy	Test accuracy
Model1 – original data	73.15%	54.06%
Model1 – generated data	60.95%	56.58%
Model2 – original data	72.85%	56.80%
Model2 – generated data	60.09%	57.91%

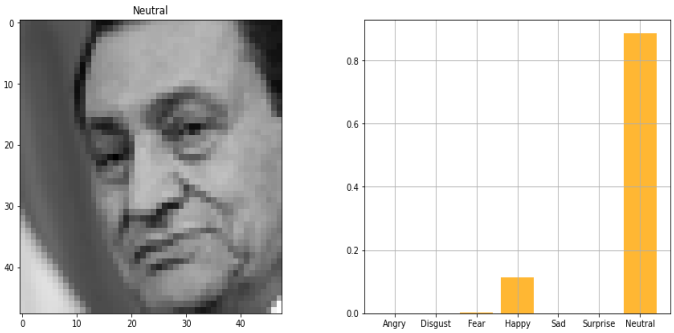

Fig 14. Prediction of a neutral emotion

VII. Related Work

The paper "Robust real-time emotion detection system using CNN architecture" [8] focuses on implementing a real time emotion detection network for human-robot interaction. The model uses varying filter size, convolution layers and down sampling to extract the best features in the deep layers. The model also manages to be computationally less expensive to its counterparts based on the datasets used for evaluation.

Burkert el al. [7] built an architecture with parallel feature extraction consisting of convolution, pooling and ReLU activation layers. Utilizing cross-validation the models

developed were highly accurate over the MMI database. The MMI database consists of videos of people showing emotions. Frames from each video was sampled and used to train and test the model. Additionally, majority of the misclassifications in their model was in the earlier frames where most of the images were not yet conveyed.

A profound study dealing in human emotion [9] combines the facial expression and speech of an interaction to predict human emotion. It focuses on the literature behind the interpretation of information through two channels, namely speech and visual. The study also discusses the inherent pitfalls of multi-label classification as every may have a different portrayal of emotion. It dives deeper into the attitudes, moods and multiple contributing facets.

VIII. Conclusion

In summary gradual improvements in the FEA models were observed throughout the project. In the process, overfitting was identified and addressed using Early Stopping and regularization layers. Also, a practical application of data augmentation was presented and improved results were attained.

The project successfully developed a generalized model for prediction over fer2013 dataset. Fer2013 is a challenging dataset and the model performed well in analyzing the probabilities for each emotion. This model with additional cross-validation can improve in performance but will be computationally expensive to train.

References

[1] https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data

[2] https://www.tensorflow.org/api_docs/python/tf/keras

[3] https://elitedatascience.com/overfitting-in-machine-learning

[4] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." *arXiv preprint arXiv:1412.6980* (2014).

[5] https://bair.berkeley.edu/blog/2019/06/07/data_aug/#:~:text=Data%20augmentation%20is%20a%20strategy,to%20train%20large%20neural%20networks.

[6] https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator

[7] Burkert, P., Trier, F., Afzal, M.Z., Dengel, A. and Liwicki, M., 2015. Dexpression: Deep convolutional neural network for expression recognition. *arXiv preprint arXiv:1509.05371*.

[8] Jaiswal, S., Nandi, G.C. Robust real-time emotion detection system using CNN architecture. *Neural Comput & Applic* **32,** 11253–11262 (2020). https://doi.org/10.1007/s00521-019-04564-4

[9] Cowie, R., Douglas-Cowie, E., Tsapatsoulis, N., Votsis, G., Kollias, S., Fellenz, W. and Taylor, J.G., 2001. Emotion recognition in human-computer interaction. *IEEE Signal processing magazine*, *18*(1), pp.32-80.

[10] https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/