A Facial Expression Recognition Approach Using DCNN for Autistic Children to Identify Emotions

Md Inzamam Ul Haque
Ingram School of Engineering
Texas State University
San Marcos, USA
m_h536@txstate.edu

Damian Valles
Ingram School of Engineering
Texas State University
San Marcos, USA
dvalles@txstate.edu

***Abstract*—In this paper, an initial work of a research is discussed which is to teach young autistic children recognizing human facial expression with the help of computer vision and image processing. This paper mostly discusses the initial work of facial expression recognition using a deep convolutional neural network. The Kaggle's FER2013 dataset has been used to train and experiment with a deep convolutional neural network model. Once a satisfactory result is achieved, the dataset is modified with pictures of four different lighting conditions and each of these datasets is again trained with the same model. This is necessary for the end goal of the research which is to recognize facial expression in any possible environment. Finally, the comparison between results with different datasets is discussed and future work of the project is outlined.**

Keywords—Facial Expression Recognition, Autistic Children, DCNN, Loss, Accuracy

I. Introduction

Strong and meaningful human interaction is necessary to convey feelings and communicate with another person. Along with verbal communication, conveying communicative feelings can also be carried out by a person via nonverbal communication such as body language, facial expression, attitude, movement etc. [1]. One of the non-verbal communication methods by which one can understand the mood/mental state of a person is the expression of the face [2]. Facial expressions play a significant role in interpersonal communication. In 1971, Ekman et al. [3] identified six facial expressions that are universal across all cultures - anger, disgust, fear, happiness, sadness and surprise.

As infants, nonverbal communication is learned from social-emotional communication, making the face rather than voice the dominant communication channel [4]. For autistic children, face processing is a challenging task. It has been argued that the ability of autistic children to understand facial expression is impaired and this inability may account for other problems that they demonstrate during social interaction [5]. Several studies including [6] and [7] showed the impairment of autistic children in classifying and understanding facial expressions compared to normal children of the same age. Interestingly, most of these studies used static front view images or drawings.

In this paper, a novel idea is presented of teaching young autistic children to recognize human facial expressions in a friendly and practical environment. As most autistic kids like to play with gadgets such as smartphones, tablets etc., the goal will be to teach them to recognize facial expressions using these gadget's camera. When they will point the camera towards a person, the app will automatically detect the face and classify facial expression of the person. The facial expression will be shown on the gadget's screen in the form of an emoticon. The goal of this research is to use these facial expressions as an emoticon to show autistic children how the person, to whom they are pointing the camera, is feeling and displaying emotional characteristics. To make this model robust to any environment and angle, the model will be trained with not only front-view facial images but also with images of faces from different orientation, i.e. side view, top view and bottom view. Also, our model should be able to predict facial expression in different lighting environments: darker or lighter shades of contrast. Fig. 1 shows a simple workflow diagram of the application.

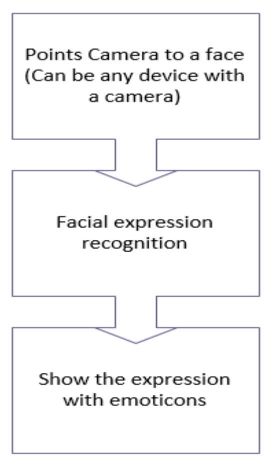

Fig. 1. *Work flow diagram for the application of research.*

The second step of the overall application, the facial expression recognition will be done with the help of computer vision and neural networks by mainly using DCNNs (Deep Convolutional Neural Networks) design approach. In recent years, deep learning has made a noticeable progress in the task of classifying facial expression. The famous Kaggle's "Challenges in Representation Learning: Facial Expression Recognition Challenge" [8] leaderboard shows that 71.16% test accuracy has been achieved in classifying seven different classes of facial expression with FER2013 dataset [9].

As an initial work of our research, a deep CNN (Convolutional Neural Network) is trained on the famous FER2013 dataset. Also, the images in this dataset have been modified such as the brightness of the images has been changed to examine the model performance in different lighting environment.

The rest of this paper is organized as follows. Section II reviews related work on facial expression recognition studies for autistic children and FER approaches in general. Section III discusses the computational resources that has been used in this initial work. Section IV describes the DCNN model architecture that has been used. Section V shows the experimental results and analysis. Future scope and conclusion are presented in Section VI.

II. RELATED WORK

A. Facial Expression Recognition studies for Autistic children

Many studies have been done about the impairments, children with Autism Spectrum Disorder (ASD) typically face while processing human facial expression and different methods have been proposed to teach them how to learn different facial expression. In paper [5], the authors assessed the influence of motion on facial expression recognition in young autistic children. They were compared on their ability to match videotaped "still," "dynamic," and "strobe" emotional and non-emotional facial expressions with photographs. Compared to previous studies showing a lower performance in autistic than in control children when presented with static faces, author's data suggest that slow dynamic presentations facilitate facial expression recognition by autistic children. Another paper, in [10], presented a detailed study of the implementation of serial and parallel implementation of PCA to identify the most feasible method for realization of a portable emotion detector for autistic children. Different experiments, surveys, and comparison have been made about autistic children's facial expression recognition problems and impairments including [11], [12].

B. CNN models for facial expression recognition

In past few years, deep learning specially DCNN has made a considerable progress in the task of recognizing human facial expression. In paper [13], the authors reviewed the state of the art in image-based facial expression recognition using CNNs and highlight algorithmic differences and their performance impact. Authors also demonstrated that overcoming one of these bottlenecks – the comparatively basic architectures of the CNNs utilized in this field – leads to a substantial performance increase. By forming an ensemble of modern deep CNNs, they did obtain a FER2013 test accuracy of 75.2%, outperforming previous works without requiring auxiliary training data or face registration.

Another project [14] applied various deep learning CNN methods to identify the key seven human emotions: anger, disgust, fear, happiness, sadness, surprise and neutrality. Using FER2013 dataset, authors leveraged ensemble and transfer learning techniques to achieve the best results. Thus, the accuracy using ensemble learning was 67.2% and with transfer learning was 78.3%, solid results are given by the winner of the Kaggle Facial Expression Recognition Challenge had an accuracy of 71.2%, and those who ranked in the top 10 of the same competition only achieved accuracies starting at around 60%. Using visual saliency and deep learning on CFEE and RaFD datasets, authors of [15] achieved test accuracies of respectively 74.79% and 95.71%. Paper [16] proposed methods of refining individual models and then combining different convolutional neural networks which achieved significant end results, with the highest test accuracy of 87.4% from their Linear SVM approach. Using a combination of algorithms for face detection, feature extraction, and classification, paper [17] achieved average expression recognition accuracy of 97.71% and 95.72% respectively for Japanese Female Facial Expression (JAFFE) and the Extended Cohn-Kanada (CK+) datasets.

III. COMPUTATIONAL RESOURCES

For training of the DCNN used in this paper, the LEAP (Learning, Exploration, Analysis, and Processing) next-generation High-Performance Computing (HPC) Cluster [18] of Texas State University has been used. The new LEAP Dell PowerEdge C6320 Cluster is configured with 120 compute nodes, each with 28 CPU-cores via two (14-core) 2.4 GHz E5-2680v4 Intel Xeon (Broadwell) processors. With 128 GBs of memory and 400 GBs of SSD storage per node, the compute nodes provides an aggregate of 15 TBs of memory and 48 TBs of local storage. Additionally, LEAP features two large memory (1.5TB) nodes with 64 CPU cores via four (16-core) 2.5 GHz E7-8867v3 Intel Xeon (Haswell) processors. Fig. 2 shows the features and components of LEAP cluster.

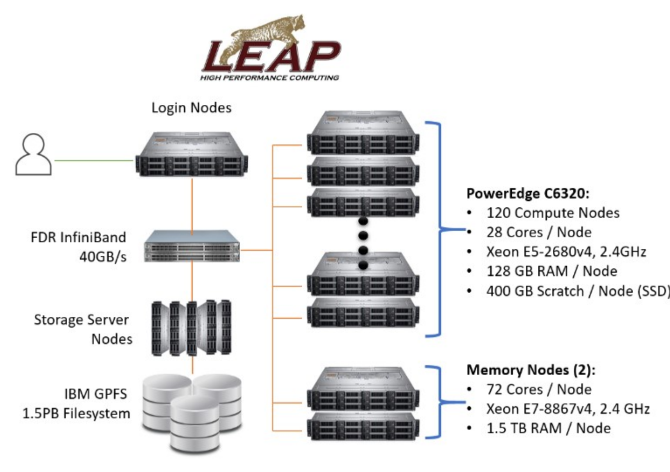

Fig. 2. *Features and components of LEAP cluster*

All scripts used in this research are written in Python 3.5.7 programming language. Conventional and popular deep learning libraries such as Keras (TensorFlow backend), Numpy, Scikit-learn, Matplotlib are used. For image processing, OpenCV for Python was utilized.

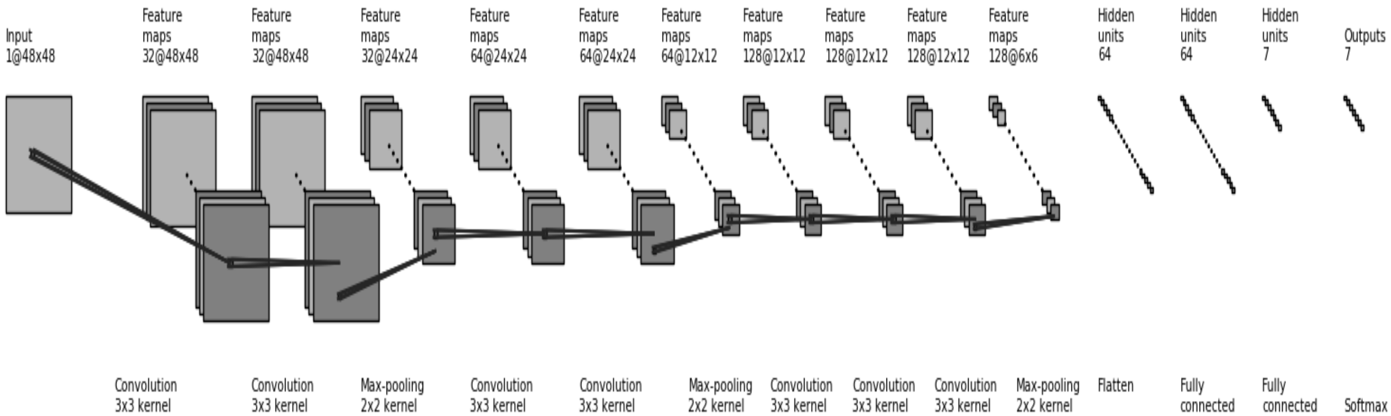

Fig. 3. *Architecture of the implemented DCNN (Figure is generated by adapting the code from https://github.com/gwding/draw_convnet)*

IV. DCNN Model Architecture

The DCNN used in this paper is inspired by the family of VGGNet networks [19], specially VGG16 architecture. The design idea is also heavily inspired by [20]. A full network architecture is shown above in fig. 3. As it can be seen, the model takes an input image of 48 × 48 pixel and after applying several Convolution, Max-Pooling and Fully Connected layer, the final output of seven classes: Anger, Disgust, Fear, Happiness, Sadness, Surprise and Neutral was achieved. The whole model architecture is also tabulated in Table I with corresponding layers starting from top to bottom.

All the convolution layers have a filter size of 3 × 3 and all the pooling layers have a pool size of 2 × 2. Neither in the diagram nor in the table, activation, batch normalization or dropout layers are included. Every convolution layer is followed by an Activation Layer. Batch Normalization is also included after every activation layer to gain higher accuracy. In the final model as discussed in the next section, every Pooling layer is followed by a dropout layer with a value of 0.25. The dropout layer is normally used after fully connected layers, but after performing run trials, it has been seen that using dropout after pooling layer helps reducing overfitting of the model. This idea was also taken from [20]. The convolution-pooling group was written as *Convolution => Activation => Batch Normalization => Pooling => Dropout*. The Fully Connected layers consists of similar architecture such as *Fully Connected => Activation => Batch Normalization => Pooling => Dropout*. The only difference is that the dropout layer has a value of 0.5. Before reaching the output layer, a Softmax Classifier is used to finally classify the image into seven distinct categories. The FER dataset has been trained using this model as well as modified FER datasets for dark and bright images.

To better understand, fig. 4 shows an actual picture of the FER dataset along with its light and dark versions. Four different datasets are made using these dark and bright pictures which are trained using the same model. This demonstration is necessary to achieve the final goal of the project which is to make an application that can evaluate human facial expressions in every possible lighting environment. The results are discussed in the next section.

TABLE I. DCNN Architecture

Type of Layer	***Output Size***	***Filter / Pool Size***
Input Layer	48 × 48 × 1	3 × 3
Convolution	48 × 48 × 32	3 × 3
Convolution	48 × 48 × 32	3 × 3
Max-Pooling	24 × 24 × 32	2 × 2
Convolution	24 × 24 × 64	3 × 3
Convolution	24 × 24 × 64	3 × 3
Max-Pooling	12 × 12 × 64	2 × 2
Convolution	12 × 12 × 128	3 × 3
Convolution	12 × 12 × 128	3 × 3
Convolution	12 × 12 × 128	3 × 3
Max-Pooling	6 × 6 × 128	2 × 2
Fully Connected	64	
Fully Connected	64	
Fully Connected	7	
Softmax	7	

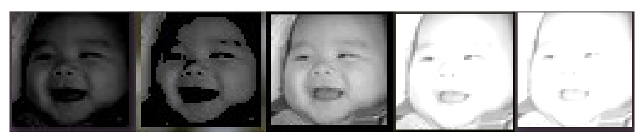

Fig. 4. *Original picture from FER dataset (middle), darker versions of the picture are on the left and brighter versions of the picture are on the right. The most left one is the darkest one and most right one is the brightest one.*

V. Results and Discussion

A lot of experiments have been done to get a satisfactory test accuracy for different version of the FER dataset as stated above. Different parameters of the model stated in previous section are changed in each experiment and the corresponding result is then compared to get the best result. At first, the standard SGD optimizer with ReLU activation function [22] was utilized. This model had a batch size of 128 and Keras default kernel initializer. Then the model was trained with different learning rates which are 0.01, 0.005, 0.001, 0.0005 and 0.0001. However, it was seen from the loss/accuracy curves that for all the above learning rates the model was overfitting. It is worth mentioning here that all these models had no dropout layer before the fully connected layers. Only after fully connected layers a dropout layer with a value of 0.5 was utilized. Here, the best validation accuracy achieved was

60.23% with a learning rate of 0.01. This batch of experiments will be denoted as the first experiment.

The second experiment was of the same configuration as the first with just one exception. Here, a dropout layer with a value of 0.25 was added after every pooling layer. This exception made a significant increase in validation accuracy. The best validation accuracy of 65.24% was achieved for a learning rate of 0.01. However, it was seen from the loss/accuracy plot that the model was again overfitting a lot. All these models were run for 70 epochs. The loss/accuracy plot for this model with a learning rate of 0.01 is given in fig. 5. Matplotlib library was used to generate these loss/accuracy plots.

To solve this overfitting problem in the next batch of experiments, identified as the third experiment, the learning rate schedulers were established. For this batch of experiment, an initial learning rate of 0.01 was selected as the best result so far was achieved with this learning rate. The default Keras learning rate schedulers was used with a decay parameter of respectively 0.01/10, 0.01/20, 0.01/30, 0.01/40, 0.01/50. The SGD optimizer was initialized with a momentum value of 0.9 and Nesterov accelerated gradient was used. Keras applies the following formula in (1) for adjusting learning rate after every batch update.

$$lr = initial\_lr \times (1/(1 + decay \times iterations)) \qquad (1)$$

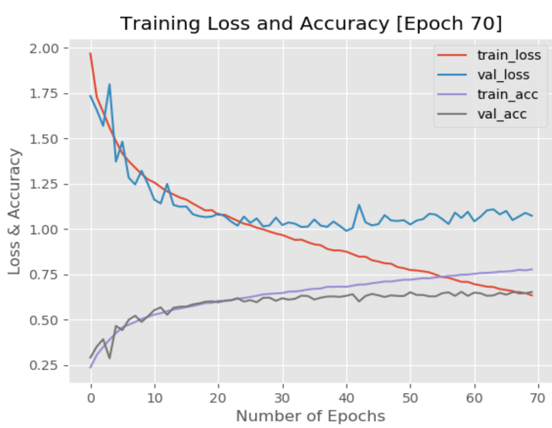

Fig. 5. *Loss/ Accuracy plot for the model of second experiment with a learning rate of 0.01.*

Here, *lr* refers to the learning rate and initial learning rate is set to 0.01. Iterations can be calculated as total number of training images in the dataset divided by the batch size which is 128 in this case.

From this batch of experiment, the overfitting problem is removed in expense of validation accuracy. The best result from experiment three was with a decay parameter of 0.01/20 and the validation accuracy was 63.77%. The loss/accuracy plot for this model is given in fig. 6. From the loss/accuracy plot it is seen that both loss and accuracy of the model do not actually stagnate after epoch 70. For the next experiment the approach was to increase the number of epochs from 70 to 100.

For the fourth experiment, the number of epochs were increased to 100 and activation function was also changed from ReLU to ELU. Details of ELU can be found in [23]. This yielded a validation accuracy of 62.50%. However, changing the batch size from 128 to 64 and the initializer to MSRA initialization [24] from Keras default yielded a validation accuracy of 63.26% and a test accuracy of 62.20%. The plot/accuracy plot for this model is given in fig. 7.

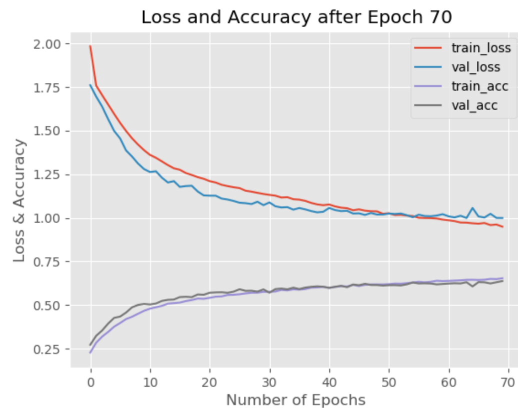

Fig. 6. *Loss/ Accuracy plot for the model of the third experiment with a starting learning rate of 0.01 and a decay parameter of 0.01/20.*

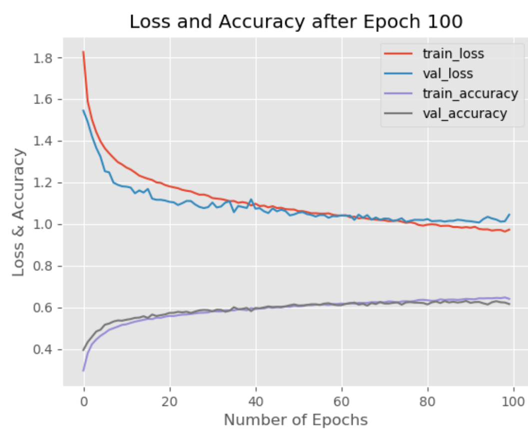

Fig. 7. *Loss/ Accuracy plot for the model of the fourth experiment with a starting learning rate of 0.01and a decay parameter of 0.01/20.*

For the fifth experiment, the Adam optimizer in [25] was initially considered over the SGD. Though this optimizer provided the best validation accuracy achieved so far which is 66.81%, the model was overfitting by a great amount. Therefore, the change to the Keras default initializer was made and a validation accuracy of 64.88% was achieved with no overfitting. The loss/accuracy plot for this model is given in fig. 8. It should be mentioned that some data augmentation techniques were applied to this model such as rotation, changes in scale and horizontal flip. These data augmentation techniques help to obtain more training data by changing geometric features of the original images. As a result, data augmentation can increase generalizability of the model as well as test accuracy. That is why the validation accuracy is slightly better than the training accuracy of 63.74%. Also, the use of dropout layers after every pooling layers while training the dataset as well as the batch size can be a reason for this. The test accuracy achieved from this model was 63.11%.

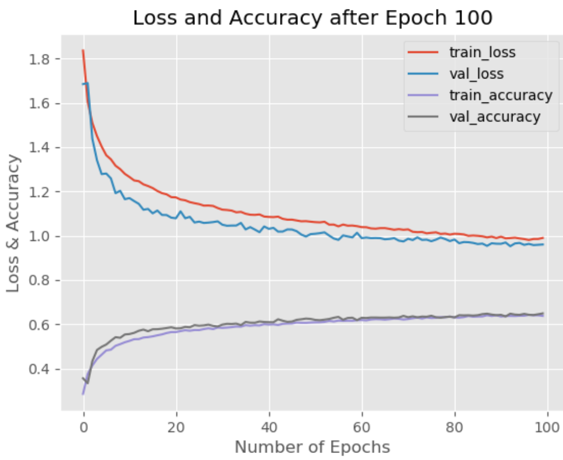

Fig. 8. *Loss/Accuracy plot for the final model of the fifth experiment.*

Next, the same final model obtained from the fifth experiment was used to train four different modified FER dataset containing images of different lighting (darkest, darker, brighter, brightest) which are discussed in the previous section. These modified datasets were produced with the help of Python and OpenCV. For the dataset with brighter and brightest images, test accuracies of respectively 59.65% and 58.37% were achieved which are close to the test accuracy of original FER dataset of 63.11% which was achieved in the fifth experiment. For the dataset with darker and darkest images, test accuracies achieved were respectively 53.10% and 46.46% which are worse compared to the brighter contrast images. Therefore, it can be concluded that the model is having some difficulties recognizing dark contrast facial images. Fig. 9 to fig. 12 show the loss/accuracy plots for these four different datasets- darkest, darker, brighter, brightest respectively using the final model from experiment five. To better understand, a comparison for test accuracies with different datasets is given in fig. 13. It is seen from this chart that this model is doing better with bright images than dark images.

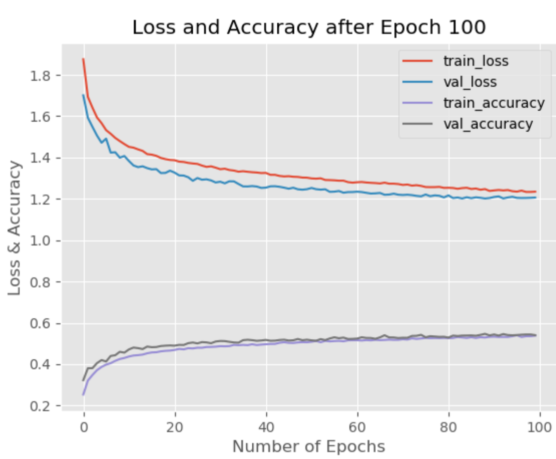

Fig. 9. *Loss/Accuracy plot for modified FER dataset with darkest images.*

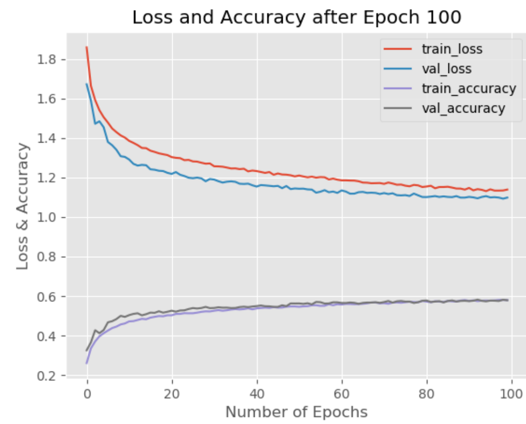

Fig. 10. *Loss/Accuracy plot for modified FER dataset with darker images.*

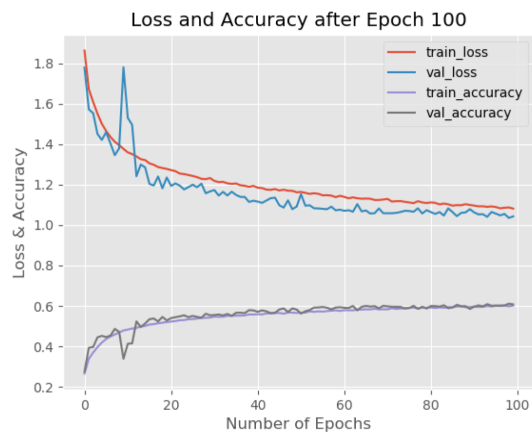

Fig. 11. *Loss/Accuracy plot for modified FER dataset with brighter images.*

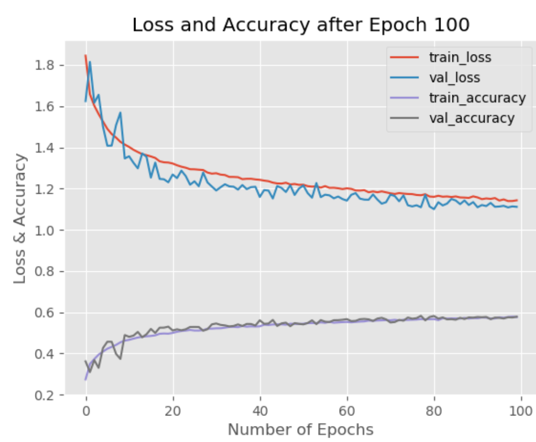

Fig. 12. *Loss/Accuracy plot for modified FER dataset with brightest images.*

VI. Conclusion and Future Scope

In this paper, the goal was to design a deep neural network for facial expression recognition that can help autistic children recognize emotions. As this is an initial work, other approaches are being considered for a better model which can recognize dark and bright images as well as can improve the test accuracy.

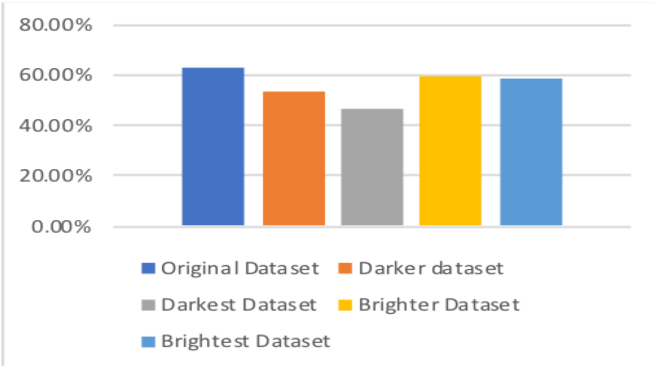

Fig. 13. *Comparison of test accuracies with different datasets*

The FER dataset was utilized as the trial image dataset and was modified to versions containing images of different lighting condition. Next idea will be to experiment with datasets containing images from different angles such as top view, bottom view and side views. It will enable this effort to have a model which can recognize human facial expression from any angle in any lighting condition in any environment.

Other two parts apart from facial expression recognition in fig. 1 are also left for future work. First task will be the preprocessing of images after capturing a face with a camera to easily classify emotions through facial expressions. Lastly, it will be showing an emoticon with the corresponding emotion recognized by the model into a working application.

From some of the results obtained, the best test accuracy of 63.11% was achieved without overfitting the model marking the result satisfactory. Training the same model, it was seen that the datasets with bright images yields to a better test accuracy than datasets with dark images. This initial work will lead to reach the end goal of this project which is to help autistic children recognize human facial expression.

REFERENCES

[1] T. A. Rashid, “Convolutional Neural Networks based Method for Improving Facial Expression Recognition,” in *Intelligent Systems Technologies and Applications 2016*, vol. 530, J. M. Corchado Rodriguez, S. Mitra, S. M. Thampi, and E.-S. El-Alfy, Eds. Cham: Springer International Publishing, 2016, pp. 73–84.

[2] J. Kumari, R. Rajesh, and K. M. Pooja, “Facial Expression Recognition: A Survey,” *Procedia Comput. Sci.*, vol. 58, pp. 486–491, 2015.

[3] P. Ekman and W. V. Friesen, “Constants across cultures in the face and emotion.,” *J. Pers. Soc. Psychol.*, vol. 17, no. 2, pp. 124–129, 1971.

[4] “Nonverbal communication,” *Wikipedia*. 09-Aug-2018.

[5] B. Gepner, C. Deruelle, and S. Grynfeltt, “Motion and Emotion: A Novel Approach to the Study of Face Processing by Young Autistic Children,” p. 11.

[6] S. J. Weeks and R. P. Hobson, “The Salience of Facial Expression for Autistic Children,” *J. Child Psychol. Psychiatry*, vol. 28, no. 1, pp. 137–152.

[7] R. P. Hobson, “The Autistic Child’s Appraisal of Expressions of Emotion: A Further Study,” *J. Child Psychol. Psychiatry*, vol. 27, no. 5, pp. 671–680.

[8] “Challenges in Representation Learning: Facial Expression Recognition Challenge.” [Online]. Available: https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge. [Accessed: 13-Aug-2018].

[9] N. Pinto, *fer2013: https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data*. 2018.

[10] K. G. Smitha and A. P. Vinod, “Facial emotion recognition system for autistic children: a feasible study based on FPGA implementation,” *Med. Biol. Eng. Comput.*, vol. 53, no. 11, pp. 1221–1229, Nov. 2015.

[11] E. Loth, L. Garrido, J. Ahmad, E. Watson, A. Duff, and B. Duchaine, “Facial expression recognition as a candidate marker for autism spectrum disorder: how frequent and severe are deficits?” *Mol. Autism*, vol. 9, no. 1, p. 7, Jan. 2018.

[12] J. Boucher and V. Lewis, “Unfamiliar Face Recognition in Relatively Able Autistic Children,” *J. Child Psychol. Psychiatry*, vol. 33, no. 5, pp. 843–859.

[13] C. Pramerdorfer and M. Kampel, “Facial Expression Recognition using Convolutional Neural Networks: State of the Art,” *ArXiv161202903 Cs*, Dec. 2016.

[14] A. Savoiu and J. Wong, “Recognizing Facial Expressions Using Deep Learning,” p. 7.

[15] V. Mavani, S. Raman, and K. P. Miyapuram, “Facial Expression Recognition Using Visual Saliency and Deep Learning,” in *2017 IEEE International Conference on Computer Vision Workshop (ICCVW)*, 2017, pp. 2783–2788.

[16] C. Huang, “Combining convolutional neural networks for emotion recognition,” in *2017 IEEE MIT Undergraduate Research Technology Conference (URTC)*, 2017, pp. 1–4.

[17] L. Nwosu, H. Wang, J. Lu, I. Unwala, X. Yang, and T. Zhang, “Deep Convolutional Neural Network for Facial Expression Recognition Using Facial Parts,” in *2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*, Orlando, FL, 2017, pp. 1318–1321.

[18] leap, “LEAP - High Performance Computing Cluster: Division of Information Technology: Texas State University,” 08-May-2018. [Online]. Available: http://doit.txstate.edu/rc/leap.html. [Accessed: 01-Aug-2018].

[19] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *ArXiv14091556 Cs*, Sep. 2014.

[20] A. Rosebrock, *Deep Learning for Computer Vision with Python*, 1.3.0. PyImageSearch.com, 2018.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[22] R. H. R. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung, “Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit,” *Nature*, vol. 405, no. 6789, pp. 947–951, Jun. 2000.

[23] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, “Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs),” *ArXiv151107289 Cs*, Nov. 2015.

[24] K. He, X. Zhang, S. Ren, and J. Sun, “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification,” *ArXiv150201852 Cs*, Feb. 2015.

[25] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *ArXiv14126980 Cs*, Dec. 2014.