Facial Expression Recognition with HOG using CNN

Tanvir Ahmed   
1410982042   
CSE465.2  
*Department of Electrical and Computer Engineering (ECE)*  
*North South University (NSU)*Bashundhara, Dhaka-1229, Bangladesh  
ahmed.tanvir@northsouth.edu

Md. Adban Akib Protik   
1610523042  
CSE465.2  
*Department of Electrical and Computer Engineering (ECE)*  
*North South University (NSU)*Bashundhara, Dhaka-1229, Bangladesh  
akib.protik@northsouth.edu

Md Shehabub Zaman Pranta   
1611251042  
CSE465.2  
*Department of Electrical and Computer Engineering (ECE)*  
*North South University (NSU)*Bashundhara, Dhaka-1229, Bangladesh  
shehabub.pranta@northsouth.edu S. M. Mahir Anowar   
1621099042   
CSE465.2  
*Department of Electrical and Computer Engineering (ECE)*  
*North South University (NSU)*Bashundhara, Dhaka-1229, Bangladesh  
mahir.anowar@northsouthedu

*Abstract*— Recognizing Facial Expression is an integral part of social interaction. It is the universal way of non-verbal communication. Therefore, deep learning for Facial Expression Recognition (FER) has been an active field for research. Some basic human expressions are Angry, Disgust, Fear, Happy, Sad, Surprise, and Neutral. There is an intra-class variation among these expressions, because of that Facial Expression Recognition is a challenging task. At the same time, it is a vast area of interest for its wide applications in human-computer interaction, identifying psychological conditions, robotics, surveillance, security, etc. In this work, we first described the FER2013 dataset used here, which contains 35,887 images of facial expressions grouped in seven classes: Angry, Disgust, Fear, Happy, Sad, Surprise, and Neutral. We also gave some background concepts for Histogram of oriented Gradients features and explored some related work for Facial Expression Recognition. Next, a model for Facial Expression Recognition using Convolutional Neural Network (CNN) has introduced. Then Histogram of oriented Gradient features is extracted from the FER2013 dataset to feed the CNN model. There is a slight drop of accuracy with the Histogram of oriented Gradients based CNN, whereas the CNN model gives better accuracy using the raw pixels of images for training. For which, we did some comparative analysis.

Keywords— Deep Learning, Facial Expression, Emotion Recognition, CNN, HOG, FER2013

# **Introduction**

The prospect of facial expression recognition is becoming more and more of an enticing topic these days. It has become something that has a huge market waiting for itself if it can be achieved. Its application can be quite varied. It can revolutionize robotics [6]. If we can integrate expression recognition into an AI, it could be the next major step for more human-like robots. It is also a desirable prospect when it comes to market research, as recognizing customer expressions could lead to understanding what customers want and need. Video game development could also achieve new heights with this addition as games could cater to the player’s expressions to create specific atmospheres and suspense [4].

Researchers have analyzed human facial expressions to consist of six different categories. They are fear, disgust, surprise, anger, sadness & happiness [3]. These expressions can be recognized by the creases on someone’s face, the orientation of their face, etc. The main challenge of it is, it can be quite varied. People may put on a big smile on their face to show happiness while someone could show the same with the smallest of smirks. The same way someone could show their fear by letting out a scream while others could by opening their eyelids wider than usual. But these variations are not the only problem for recognizing an expression, as similarities between the different categories could be considered the biggest hurdle to cross. Fear and surprise have very few distinctions between them as in both cases, humans open their eyelids wider and may have their mouths open. Disgust and anger also have these similarities, as they both prompt humans to create a crease between their eyes. These common factors between different categories of expressions make the task of expression recognition quite a hard one to achieve. It has been a highly discussed topic in computer science and physical science, as well [2]. Expressions are basic emotions being given a physical form, something quite useful in the field of psychology.

Research regarding this subject has been prevalent in the last decade, as many researchers have tried their hand on accurately recognizing expressions using neural networks. Most of these researches have good results but have a problem distinguishing the similarities between categories and the dissimilarities between expressions in the same category as stated before. Mostly faltering when recognizing the fear and disgust expressions [5].

In this paper, the intention is to create a Convolutional Neural Network (CNN) model and train it with a dataset containing properly labeled images of different facial expressions. The corresponding model should create a facial recognition system that can recognize human expressions from an image and classify it into the six expression categories deduced by researchers. Then, to extract the HOG features of the images in the dataset, putting the extracted features through the same CNN. The final goal is to compare the results between the two and find out whether extracting the features enhances efficiency.

Other papers have tried to use similar techniques [7]. This research mainly made efforts towards creating a new model and expanding towards this research. It wanted a more definitive answer to the usability of feature extraction techniques. The rest of the paper will describe the dataset, talk about the HOG feature extraction method, define the methodology of our report, display the results, and express the conclusion reached in the end and give out a few ideas about some future work done regarding this topic.

# **Dataset Description**

The dataset used for the research is one of the most popular datasets used in facial expression recognition. It was the FER2013 dataset created with the help of Google image API. The dataset was designed during a contest that asked programmers to use the dataset to create an efficient facial expression recognition system. The contest was based on the accuracy that could be achieved using the dataset. The contest was posted on many websites as well [1], attracting a lot of eager programmers. The data set contains three columns labeled emotion, pixels & usage. In this dataset, the emotion column represents the seven categorical emotions. The pixels column consists of the pixel values of the expression images and the usage column having the distinction of the data being either training, testing, or validation. The dataset has a total of 35,887 samples of images given to us in the greyscale form with their corresponding emotion values and usage criteria. With 25,709 samples in the training criteria, 3,589 in testing criteria, and 3,589 more in the validation criteria. The images present in the database are all in 48x48 dimensions. They are appropriately positioned, so the faces in the images are in the center. The emotions are labeled from 0 to 6, with them being classified in ascending form as angry, disgust, fear, happy, sad, surprise, and neutral.

The problem with this dataset is that it is not equally distributed according to its expression categories. With angry has 4953, disgust having 547, fear to have 5121, happy having 8989, sad having 6077, surprise having 4002 and neutral having 6198 samples. These are not very close numbers, especially with disgust having an absurdly low amount of 547 samples and happy having very large 8989 samples. These make the task a lot harder.

# **Related Work**

There is plenty of research works on Facial Expression Recognition (FER) and the FER2013 dataset. FER2013 isn't a big dataset [9] compared to other existing datasets consisting of images. So, to achieve good accuracy with this dataset, different researches have taken different approaches and models.

One study [10] reviews papers that have six different methods for CNN-based FER. The paper describes methodological differences in the methods and their performances. The paper identifies the bottlenecks and comes up with an improved technique to be used for human-computer interaction and data analytics. This study was able to obtain good test accuracy without needing additional training data.

Another paper [11] works with FER to form better communication between NAO robots and humans. For better communication, robots need to understand the emotions of humans. This paper works with improving accuracy, generalization, inference speed of different CNNs using different optimization methods. This study uses FER2013 as well as other databases. The study can achieve good accuracies for different CNNs.

In the next paper [9], the authors propose a method for a FER task. The method works to reduce the number of parameters based on a pre-trained deep CNN. The authors work to achieve a high level of accuracy. So, they build and train a lightweight deep CNN. The method is evaluated on two FER datasets (FER2013, AffectNet). The authors show it needs fewer parameters to achieve good accuracy.

In the following paper [12], the authors work with three different convolutional architectures. They propose modifications to improve the accuracy of recognition for the FER2013 dataset. They use the bagging method, generate variations for the dataset histogram equalization and fractalization, and boost the ResNet50 network's accuracy using transfer learning. The study achieved good prediction accuracy on the FER2013 test subset, using different techniques for modifications.

With the advancement of computer vision, a lot of work is being done on FER. So, FER is quite a popular work topic. Also, to work on it, FER2013 is a popular dataset. Researchers are using different optimization methods, modification techniques, etc. to improve test accuracy for better recognition. Some researchers are identifying bottlenecks and improving methods to get better results.

# **Methodology**

## **Dataset Processing**

The dataset has seven classes with 35887 samples, and it has class imbalance data. For class 0, 4953 data; for class 1, 547 data; for class 2, 5121 data; for class 3, 8989 data; for class 4, 6077 data; for class 5, 4002 data; and for class 6, 6198 data. Among the seven classes, class 1 has the most less amount of data. The data augmentation technique has applied to increase the number of samples for class 1 and to make the network more robust.

In this case, for each sample in class 1, 7 augmented samples have been added to class 1. After adding the augmented samples, the number of samples for class 1 is 4376. Now, the dataset has a total of 40263 samples.

The 33084 samples (with augmentation) were selected for training, 3589 samples selected for validation, and 3589 samples selected for testing, among the 40263 samples before sending data to the network.

## **The CNN**

A CNN architecture has designed to recognize facial expressions from grayscale images. The input layer of the network is compatible with 48 x48 pixel images, and the output layer classifies the seven classes of Angry (0), Disgust (1), Fear (2), Happy (3), Sad (4), Surprise (5), and Neutral (6).

This network has seven convolutional layers from C1 to C7 and four MaxPooling layers from P1 to P4. The C1 layer has 3 x 3 kernel to filter 48 x 48 input images to produce 32 matrices. This layer's weight is initialized as "He\_uniform," [18] and the activation function is selected as ReLU [19]. The C1 layer's result passed to a Batch normalization [13] layer, which is again passed to the P1 layer, which has the pool size of 2 x 2. Then, the P1 layer's output moved to the C2 layer, which has the 3 x 3 kernel to produce 64 matrices. This layer weight initialized as "He\_uniform" and the activation function used as ReLU. The C2 layer's result passed to the C3 layer comprises 3 x 3 kernels to create 64 matrices through a Batch normalization [13] layer. The C3 layer has the activation function of Sigmoid [19], and the weights initialized as "glorot\_normal" [18]. The output of C3 moved to the P2 layer, which has a pool size of 2 x 2, through a Batch normalization layer and a Dropout [14] layer. The P2 layer's output passed to the C4 layer, which comprises 3 x 3 kernels to generate 128 matrices, and the activation function is Sigmoid [15] through a Dropout layer. Then the C4 layer's output passed to the C5 layer, which has the same parameters as C4 layer except the activation function is ReLU, through a Batch normalization layer. Next, the C5 layer is passed to the C6 layer, which has 3 x 3 kernel to produce 256 matrices through a Batch normalization layer, P3 layer of pool size 2 x 2, and a Dropout layer. The C6 layer has the Sigmoid activation function, and the weight initialization method has used as "Xavier Glorot normal" [18]. After that, the C6 layers result passed to the C7 layer, which has the same parameters as C6 except the activation function is ReLU, through a Batch normalization layer. Finally, C7 layers output sent to the flatten layer through a Batch normalization layer, P4 layer of pool size 2 x 2, and a Dropout layer, which produces 2304 parameters. At last, the output layer has seven classes with a Softmax [16] activation function.

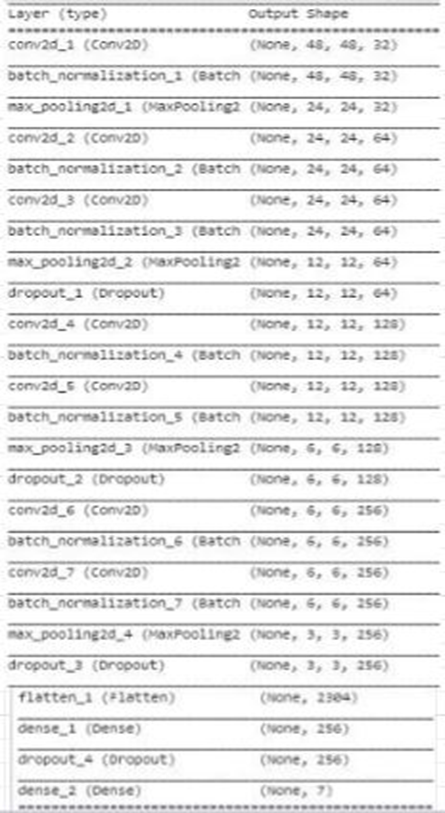


Figure 1.HOG + CNN Model architecture

## **The HOG**

There are many feature descriptors like LBP, SURF, and other methods. HOG [17] is one of them, and it is the most popular nowadays.

HOG descriptors impressively portray the depiction of the state of facial appearance discovery. A histogram of the focused picture's gradient direction is figured by separating it into little associated areas to execute these descriptors. The join of these histograms illustrates the descriptor. The procedure is separated into different stages. At first, the gradient values are computed. In the second step, particular cell-based histograms are created. Some parameters also needed to select when extracting HOG features.

Before training the network with hog image, the same number of data has divided for training, testing, and validation like before.

In this case, the number of orientation bins =9, pixels per cell= (4,4), visualization = True, and cells per block= (1,1) was selected as parameters.

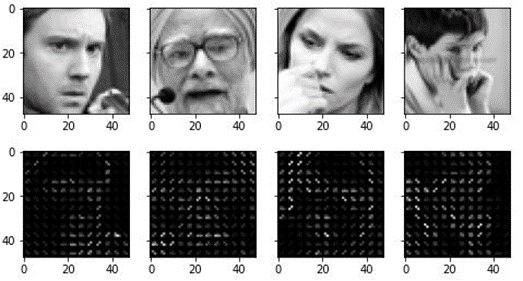


Figure 2. The Original images with their HOG features

# **Result and Analysis**

The model has tested for the FER2013 dataset. Results are assessed with various assessment matrices, to be specific, Confusion matrix, Precision, Recall, Support, Micro avg, and weighted avg, F1 Score, and training and testing accuracy graphs. The Precision shows the positive predictive value, and recall shows the true positive rate of the model. The micro-averages have used to assemble the seven categories' results to demonstrate the overall Precision and recall.

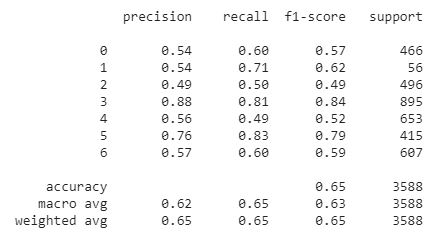


Figure 3. The precision, recall, F1-score and support of FER-CNN model

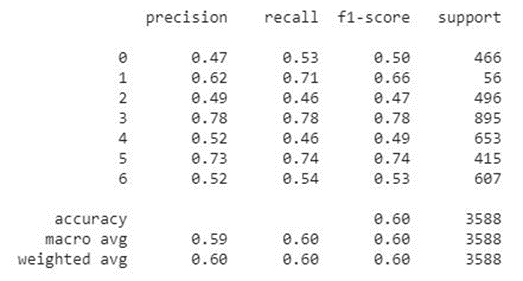


Figure 4. The precision, recall, F1-score and support of FER-HOGCNN model

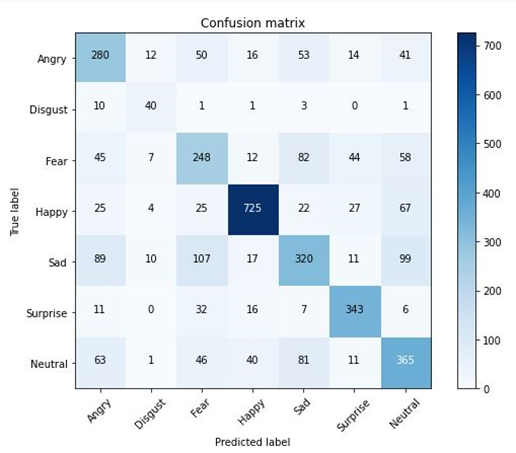


Figure 5. The FER-CNN model confusion matrix of 7-classes

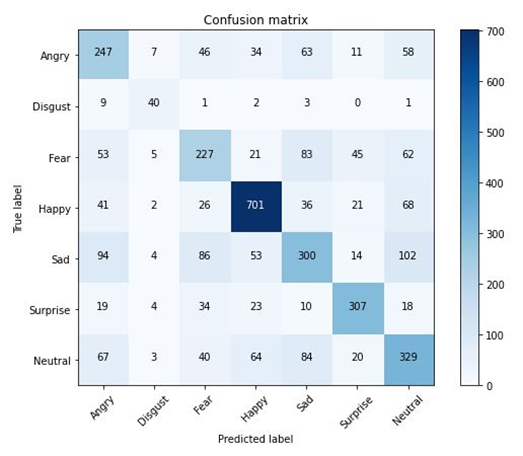


Figure 6. The FER-HOGCNN model confusion matrix of 7-classes

Figures 7 and 8 showing the training Accuracy of the FER-CNN model, which is 94%, and the testing accuracy is 66%; similarly, training Loss is 0.17, and testing loss is 1.85 on FER-CNN model. Figures 9 and 10 showing the training and testing accuracy of 89% and 61%, respectively, and losses are 0.31 on training and 1.4 on testing on the FER-HOGCNN model. The HOG feature descriptor's performances are not better than the primary model due to the small size of images and less clear image quality. The Comparative results have shown in Table 1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Accuracy | Loss | Test\_Acc | Test\_loss |
| FER-CNN | 94% | 0.17 | 66% | 1.8 |
| FER-HOGCNN | 89% | 0.31 | 61% | 1.4 |

Table 1. Accuracy for FER-CNN and FER-HOGCNN

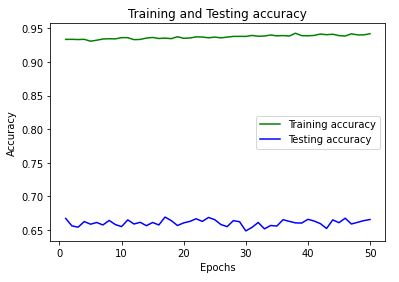


Figure 7. Training and testing accuracy of FER-CNN



Figure 8. Training and testing losses of FER-CNN

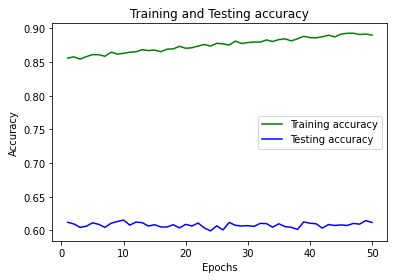


Figure 9. Training and testing accuracy of FER-HOGCNN

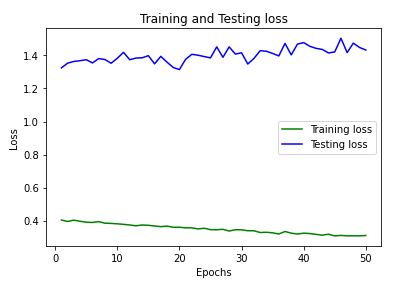


Figure 10. Training and testing losses of FER-HOGCNN

# **Conclusion**

In this research, a modified FER2013 dataset, Augmenting the images in the disgusting category was created to employ a more even distribution in the dataset. A deep neural network architecture was then introduced and applied to the augmented dataset to recognize facial expressions using CNN. Later, HOG feature data were extracted from the images and used in the same neural network architecture to test if it could extrapolate any better results. The augmented dataset contained 40,263 samples. The dataset was split into 33,085 samples of training data, 3,589 samples of training, and validation data. The CNN model with the dataset with greyscale pictures returned 94% training accuracy, 66% validation accuracy, 0.17 training loss & 1.8 validation loss. The same CNN model with the dataset of HOG features returned 89% training accuracy, 61% validation accuracy, 0.31 training loss & 1.4 validation. As can be observed, training with the HOG feature, other than validation loss, returned worse results in every other prospect. With that, it can be deduced that HOG feature extracted data is not more effective than greyscale image data when it comes to the field of facial expression recognition.

# **Future Work**

Computer vision is advancing fast. Studies are being done in all areas of computer vision. Facial Expression Recognition is vital for advancement. In recent years researchers have been able to improve accuracies significantly. New methods and models have been introduced to achieve high efficiency. So, researchers have been successful in achieving high accuracy through their work. But there is still room for improvement. No doubt, to get a better result with less computational power, new methods and techniques could come in the future. To extract important features from the images, other feature extracting methods like LBP, SURF could also work better.

# **Acknowledgment**

First and foremost, acclaims and gratitude to God, the Almighty, for His showers of blessings all through our research to effectively finish the research.

We would like to thank our honorable faculty Dr. Nabeel Mohammed, Assistant Professor, Department of Electrical & Computer Engineering, North South University, for his insight and guidance throughout the semester, which helped us to do this project. We want to thank our family: our parents, our brothers, and our sisters, to support us spiritually throughout our life.

Finally, our thanks go to all the people who have supported us to complete the research work directly or indirectly.

# **References**

1. Challenges in Representation Learning: Facial Expression Recognition Challenge. (2013, May 25). Retrieved from Kaggle: https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data
2. Chóliz, M. &.-A. (2012). RECOGNITION OF EMOTIONAL FACIAL EXPRESSIONS: THE ROLE OF FACIAL AND CONTEXTUAL INFORMATION IN THE ACCURACY OF RECOGNITION. Psychological Reports, 338-350.
3. Ekman, P. (1970). Universal Facial Expressions of Emotion. CALIFORNNIA MENTAL HEALTH RESEARCH DIGEST.
4. Facial Experssion Recognition | What is it? | How to use it? (n.d.). Retrieved from SightCorp: https://sightcorp.com/knowledge-base/facial-expression-recognition
5. N. Christou, N. K. (2019). Human facial expression recognition. In Third International Congress on Information and Communication Technology (pp. 539-545). London: Springer.
6. S. Hsu, H. H. (2017). Facial Expression Recognition for Human-Robot Interaction. 2017 First IEEE International Conference on Robotic Computing (IRC). Taichung: IEEE.
7. Sahar Zafar, I. A. (2019). Facial Expression Recognition with Histogram of. Indian Journal of Science and Technology.
8. Tawsin Uddin Ahmed, S. H. (2019). Facial Expression Recognition using Convolutional Neural Network with Data Augmentation. 8th International Conference on Informatics, Electronics & Vision. Washington: ICIEV.
9. Y. Wang, J. Wu and K. Hoashi. (Nov. 2019). Lightweight Deep Convolutional Neural Networks for Facial Expression Recognition. Available: https://www.researchgate.net/publication/336935646
10. C. Pramerdorfer. M. Kampel. (Dec. 2016). Facial Expression Recognition using Convolutional Neural Networks: State of the Art. Available: https://www.researchgate.net/publication/311573401
11. O. Melinte, L. Vladareanu. (May 2020). Facial Expressions Recognition for Human-Robot Interaction using Deep Convolutional Neural Networks with Rectified Adam Optimizer. Available: https://www.researchgate.net/publication/341355692
12. G. Porusniuc, F. Leon, R. Timofte, C. Miron. (Nov. 2019). Convolutional Neural Networks Architectures for Facial Expression Recognition. Available: https://www.researchgate.net/publication/338943405
13. “Batch normalization,” Wikipedia, 15-May-2020. [Online]. Available: https://en.wikipedia.org/wiki/Batch\_normalization. [Accessed: 01-Jun-2020].
14. “Dropout (neural networks),” Wikipedia, 18-May-2020. [Online]. Available: https://en.wikipedia.org/wiki/Dropout\_(neural\_networks). [Accessed: 01-Jun-2020].
15. “Sigmoid function,” Wikipedia, 31-May-2020. [Online]. Available: https://en.wikipedia.org/wiki/Sigmoid\_function. [Accessed: 01-Jun-2020].
16. “Softmax function,” Wikipedia, 28-May-2020. [Online]. Available: https://en.wikipedia.org/wiki/Softmax\_function. [Accessed: 01-Jun-2020].
17. “Histogram of oriented gradients,” Wikipedia, 14-May-2020. [Online]. Available: https://en.wikipedia.org/wiki/Histogram\_of\_oriented\_gradients. [Accessed: 01-Jun-2020].
18. K. Team, “Layer weight initializers,” *Keras*. [Online]. Available: https://keras.io/api/layers/initializers/. [Accessed: 02-Jun-2020].
19. K. Team, “Layer activation functions,” *Keras*. [Online]. Available: https://keras.io/api/layers/activations/. [Accessed: 02-Jun-2020].