Facial Expression Recognition

Mohan Sai Krishna Thota
U72136551
Boston University
Boston, MA 02215
mohant@bu.edu

Ajit Balla
U40978471
Boston University
Boston, MA 02215
ajit2001@bu.edu

Srinivas Chellaboina
U93606015
Boston University
Boston, MA 02215
srinevas@bu.edu

Abstract

Facial expression recognition plays a crucial role in understanding a person's emotions and bringing out their unique behaviors. With the help of the FER2013 dataset, we have been working on this project to create a model that has a respectable level of accuracy. Our ultimate objective is to incorporate the learning output into a home automation system so that when a person is at home, his environment can be set to the desired mood in order to control his emotions.

1. Introduction

Psychology and other behavioral sciences have done a great deal of research on emotions, which are thought to be a crucial aspect of human nature. The psychological state of a person is described by their emotions, which are typically based on internal factors like that person's mental and physical health. Facial expressions are the combination of emotions and the corresponding changes in facial muscle activity [1]. It aids in determining a person's present emotional and mood state [2], enabling them to interact with people in accordance with their mood. He classified basic emotions into six categories, which he called "standard facial expressions," including happiness, sadness, surprise, anger, fear, and disgust. Systems for emotional computing and facial expression recognition are crucial to human-computer interaction.

Recent years have seen a surge in interest in automatic facial image recognition. It is utilized in a variety of real-time applications, including animation, forensic interrogation, gaming, and psychiatry. According to studies, facial expressions account for 55% of human communication [3]. While researchers have developed numerous methods to accurately identify facial emotions, it is still difficult to distinguish between the expressions on the faces of people of different nationalities when they are captured from various perspectives. is [4]. Deep learning technology outperforms other approaches by having a large capacity for different datasets and quick processing speed. It is used to recognize and classify tasks like human emotions.

For representing human brain functions in neurons, deep learning techniques are the de facto paradigm [5]. Typically, this learning takes the form of a neural network model, in which neurons serve as inputs and are linked together to serve as outputs. Instead of the standard CNN, a DCNN is used to construct our model. We used the FER2013 dataset, which includes about 32000+ images with the following 7 emotions: happy, sad, surprised, neutral, disgust, anger, and fear. Our model performed significantly better than anticipated when we implemented Live using OpenCV. It was a little torn between disgust and rage. Later sections will provide an explanation of the results' causes and specifics.

2. Approach

To complete this project, we used the FER2013 dataset that Kaggle provided. One of the most well-known and frequently used datasets in the field of facial emotion recognition is FER2013. The dataset consists of 35887 48 x 48 pixels images of 7 different types of emotions. The emotions are denoted by the numbers 0 for anger, 1 for disgust, 2 for fear, 3 for happiness, 4 for sadness, 5 for surprise, and 6 for neutrality. There is already a division of the dataset into "training," "testing," and "private-testing." The testing and private test sets each contain 3500 images, while the training set contains 28000 labeled images. Figure 1 displays a few images from the FER2013 dataset.

But for the convenience and better results of the model, we have split the dataset into a custom range of testing and training using the train_test_split function. In FER2013, disgust has the fewest images [547], whereas happiness has the highest number of images [8989]. This unevenness can lead the model to be biased. The average of this dataset is 5126 images. We have performed data normalisation, data augmentation, and duplication of the images randomly in order to have equal representation of all classes by the average number of images and the highest number of images. These techniques have drastically improved the model's performance. The 70:30, 80:20, and 90:10 splitting ratios of the dataset were tested in this project, and the 90:10 splitting gave us the best results with equal representation of classes.

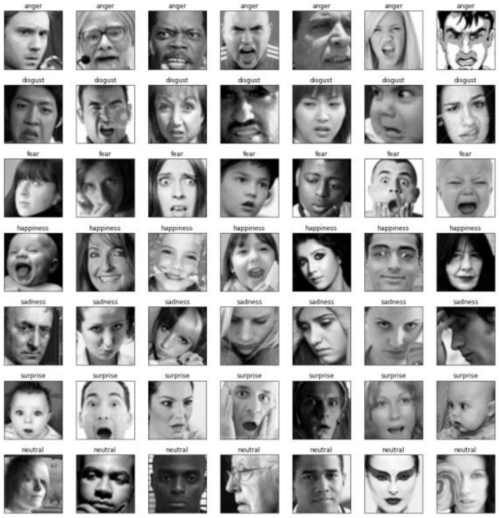

Figure 1: FER2013 data-set

A convolutional neural network (CNN) is the most popular way to classify the contents of different images. The use of which has been explosive in the area of visual computing. CNNs are supervised machine learning techniques that can extract deep knowledge from a dataset through rigorous example-based training. All the feed images will be trained by the model. The proposed model is based on the CNN framework used in classifying emotions in a person. The system design model is tested functionally at three different levels. We experimented with data centering and scaling for preprocessing. We find it generally useful to subtract the mean from the train distribution on the whole set before training or evaluation. We also implemented data augmentation: we randomly rotate, move, flip, crop, and unlink the training image. Data augmentation is a widely used technique that deals with the problem of insufficient data. This gives about 10%. Improved accuracy. Ultimately, the best performance came from a custom-designed DCNN architecture. A deep convolutional neural network (DCNN) is made up of several neural network layers. Convolutional and pooling layers are often alternated. Each filter's depth rises in the network from left to right. The last level usually comprises one or more completely linked layers.

$$f \ast g \equiv \int_{-\infty}^{\infty} f\,(\tau) g(t-\tau) d\tau$$

$$= \int_{-\infty}^{\infty} g\,(\tau) f(t-\tau) d\tau$$

Convolution is a sort of linear operation used for feature extraction in which a tiny array of numbers, known as a kernel, is applied over the input, which is an array of numbers known as a tensor. A feature map is obtained by calculating an element-wise product between each element of the kernel and the input tensor at each point of the tensor and summing it to produce the output value at the corresponding place of the output tensor. For example, if you apply a convolution to a picture, you will reduce the image size while also combining all of the information in the field into a single pixel (see Figure 4). The convolutional layer's final output is a vector. We may employ several types of convolutions depending on the sort of issue we need to solve and the features we want to learn.

One of the most typical data concerns is avoiding overfitting. Have you ever encountered a situation in which your model performs admirably on training data but fails to appropriately predict test data? The reason for this is that your model is overfitting. Regularisation is the answer to such a situation. Normalisation is a data pre-processing procedure that is used to convert numerical data to a common scale without changing the form of the data (see Figure 2). Batch normalisation is a procedure that adds extra layers to deep neural networks to make them quicker and more stable. The new layer conducts standardising and normalising actions on the input of the preceding layer. A typical neural network is trained using a pre-collected collection of input data known as a batch. Similarly, the normalising procedure in batch normalisation occurs in batches rather than as a single input.

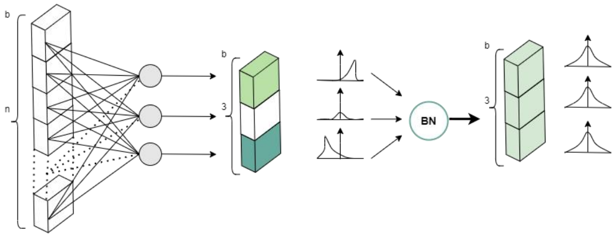

Figure 2: Batch Normalization

Analyzing errors in neural networks is very difficult. One early observation was that we fail much more at certain emotions (see the confusion matrix) and that we are failing to classify images where it is necessary to rely on fine details in the images (e.g., small facial features or curves). Due to this, we increased the number of layers and decreased filter sizes to increase the number of parameters in our network, which had a clear effect on allowing us to fit the data set better. This led to some overfitting, which we addressed by building a function to monitor stopping criteria, and parallel to this, we have designed another function that deals with adjusting the learning rate. Given this, we have successfully been able to improve the accuracy of our model and, at the same time, deal with the problem of Under-fitting/over-fitting. ReLU is the most commonly used activation function. But due to the dying ReLU problem, we shifted to the eLU function, which deals better with the limitations of the ReLU.

$$R(z) = \{z \ z > 0\}$$

$$\{\alpha \cdot (e^{x} - 1) \ z <= 0\}$$

The model consists of 6 convolutional layers, 1 dense layer, and a flattening layer between them . The Adam optimizer, as it is popularly known, is simply a combination of momentum and RMSprop. Convolution layer 1 images were batch normalised with matrix images none, 48, 48, and 64. Batch normalisation is a layer that enables each layer of the network to perform learning independently. Max pooling is now applied to the Convolution Layer 2 matrix. Max pooling is a pooling operation that selects the maximum element from the feature map region covered by a filter. The Matrix has now been converted to none, 24, 24, 64. As a result, the output of the max-pooling layer would be a feature map that contained the most prominent features of the previous feature map.

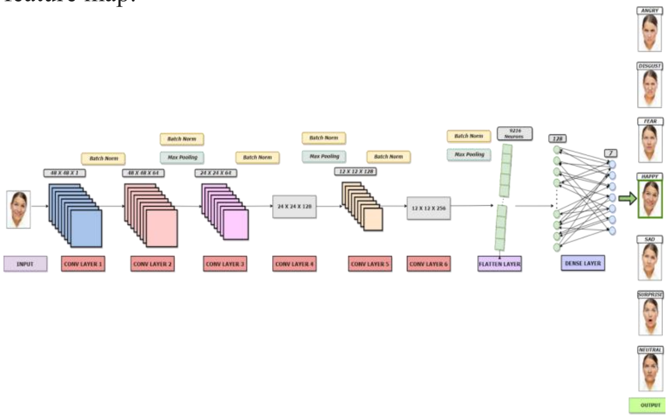

Figure 3: Model

Convolution 3 batch normalisation is used, and the matrix is converted back to none, 24, 24, 128. Max pooling is performed in convolution layer 4 with a matrix reduced to none, 12, 12, 256. Convolution layer 5 batch normalisation converts the matrix to none, 6, 6, 256. Max pooling is performed in convolution layer 6, and the matrix is flattened. Finally, a dense layer forms, and emotions are classified as happy, sad, disgusted, neutral, surprise, and fear.(see Figure 3)

3. RESULTS

In order to archive this project, we have constructed a deep convolutional neural network and even created three different datasets- Averaged , Highest , Original datsets This structure has six convolutional layers and one fully connected layer. In the first convolutional layer, we had sixty-four 5X5 kernels with a step of size 1, along with clump standardisation and dropout, but without max-pooling. In the second convolutional layer, we had the same kernels as in layer one, with the step of size 1, cluster standardisation and dropout, and furthermore max-pooling with a kernel size of 2X2. In the third and fourth convolution layers, we have increased the kernels to one hundred and twenty-eight with a size of 3X3. Following the same trend, the fifth and sixth convolution layers have two hundred and fifty-six kernels with a size of 3X3. A max pooling layer with a kernel size of 2X2 is added in the fourth and sixth layers.

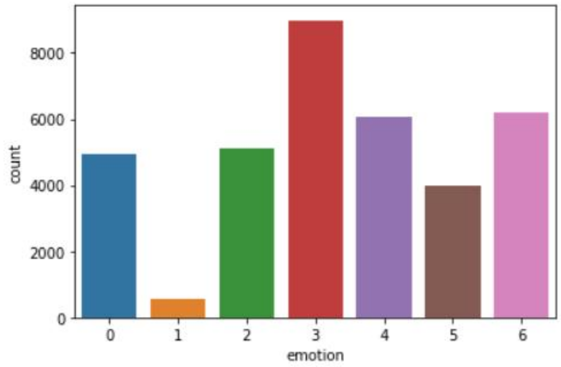

Figure 4: Distribution of class in original data-set

In the FC layer, we had a secret layer with 128 neurons and Softmax as the misfortune. Additionally, in every one of the layers, we utilized the exponential linear unit (eLU) as the initiation work. eLU is very similar to ReLU, aside from negative data inputs. They both work in a similar way. But eLU becomes smooth gradually until its result is equivalent to - $\alpha$ , where ReLU pointedly smooths. The accelerated adaptive moment estimation, or Adam optimizer, is used in this structure. The Adam algorithm is an extension to the ‘gradient descent with momentum’ algorithm and the ‘RMSP’ algorithm.

A normal kernel initializer and a learning rate of 0.001 are implemented into the structure. Initially, when we executed our model, we saw a drastic overfitting approach, which can have a negative impact. So we have utilised two functions: EarlyStopping and ReduceLROnPlateau. Early stopping monitors the overfitting/underfitting of the model by observing a few given criteria. ReduceLROnPlateau adjusts the learning rate if the model is misleading. These two functions have increased our model’s accuracy from 55% to 66%. Since our classifier has seven distinct classes, the accuracy of the model was struggling to cross 66%. Exploring the dataset (see Figure 10), we found that each class does not have an equal representation of classes, which can be one of the reasons for the fall in accuracy.

The disgust class is accompanied by 547 images, which stands to be the least represented class in the dataset, whereas the happiness class has 8989 images, which is the highest represented class(see figure 4). Along with the original dataset, which consists of uneven representation classes, we experimented with two other criteria where we managed to have the average number of images in each class as well as the highest number of images in each class. The average of the FER2013 dataset is 5126 images. We have duplicated images in a few classes and reduced the number of images in other classes to bring each class to 5126 images (see Figure 11). Having the model run with this averaged dataset, the accuracy exponentially increased to 73% from 69%.

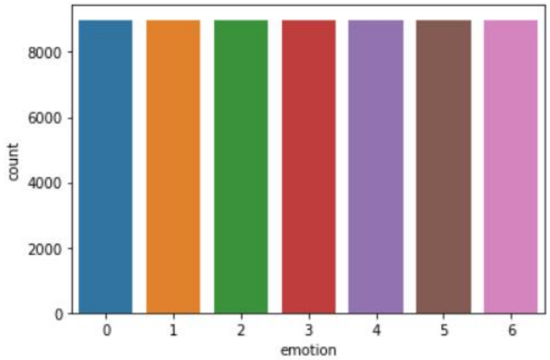

Figure 5: Distribution of class in highest data-set

As mentioned earlier, the happiness class has the highest number of images with 8989. We again adjusted and duplicated the images of the dataset in such a way that each class of the dataset has 8989 images(see figure 5). Having increased the number of images, the dataset now consists of 62923 images. Each class has the highest number of images [i.e., 8989]. The accuracy of the model stood at 80%–83%, which increased by nearly 10% from the averaged dataset and nearly 11% from the original dataset. It has the best accuracy of all the experiments. The confusion matrix (see Figure 7), accuracy, loss graph (see Figure 6).

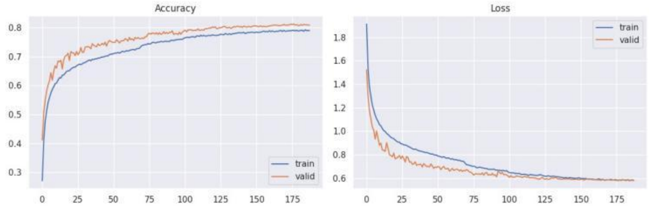

Figure 6: Accuracy and Loss

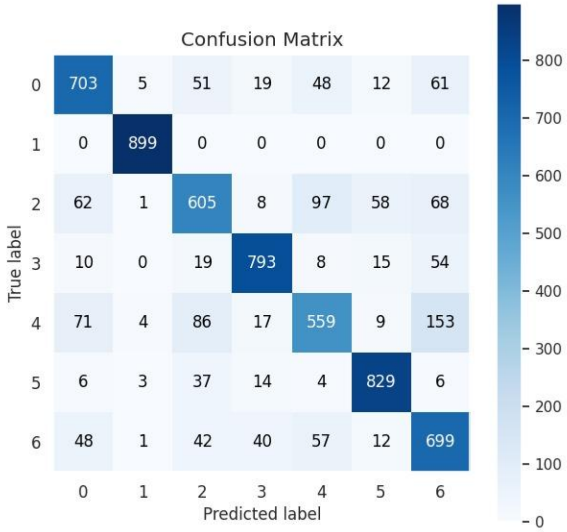

Figure 7: Confusion Matrix

As mentioned earlier we have also worked with three spitlling criterias 70:30,80:20,90:10 on each of the dataset: original , Averaged and Highest . out of all the best results were found with 90:10 on Highest dataset. After our model have attained a decent accuracy, we have used open CV to test in real time(see Figure 8) . the model have performed quite a bit better than expected but a little it struggled during anger and disgust expression. Obviouly lightning conditions , quality of webcam also played a role to get better results/

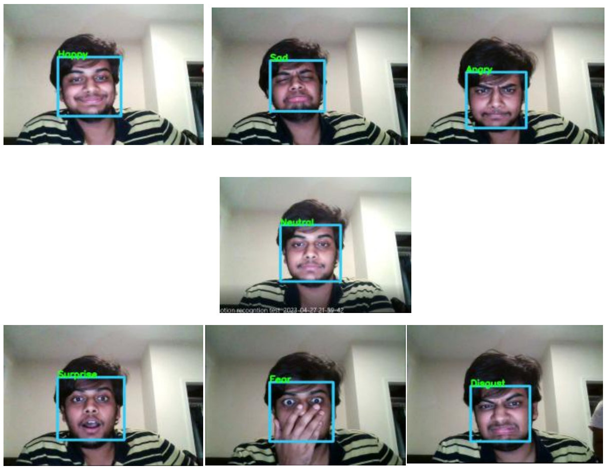

Figure 8: Live Implementation

4. Discussion and Conclusion

Face detection and expression recognition are one of the most challenging problems in the field of computer vision. The most common on is the about the dataset. initially we has difficulty to analyze the problem but later we realized that data was not equally distributed among the classes. so we had to augment the datasets into three different types .Many studies are currently underway to investigate various techniques, such as VGG16, defacements, feature extraction, etc., for effectively detecting emotions. The proposed model is based on the DCNN framework, which consists of enough information to build applications that detect spontaneous expressions in real time too. To recognize and analyse facial expressions, the FER2013 dataset is used, and the splitting ratio of 90:10 gave better results with accuracy arround 80-83% By including the seven typical facial emotions reflected by person. For better performance in the model, the activation function ELU is considered for reducing the complexity, and two functions, EarlyStopping and ReduceLROnPlateau, are used to avoid overfitting and for learning rate. For the model to perform with an ideal number of epochs, batch normalization is used. The longitudinal study results in a better performance of the proposed model, classifying all seven emotions with an accuracy of 80%-83%. though a decent accuracy is bought up the disgust class is awarded with very less images so though the accuracy of it is individually high, in real time testing that expression was hard to get captured .We further want to merge our current proposed approach with other techniques that can use EEG signals and speech combined with emotion recognition in the virtual environment.

5. WORK DISTRUBUTION

As reported, we have worked on 3 datasets of same types, splitting criterias, parameters. Each Dataset took arround 15hrs+ to test on different splitting criteria and parameters.

Mohan Sai Krishna Thota: I have worked on intial idea of this project by building the custom CNN from 3 layers to 6 layers along with Ajit Balla . I have also worked on debugging the issues in the over all code . I took up to test on Highest Dataset with Splitting criteras mentioned above, different parameters and different epoches

Ajit Balla : My contribution to this project is on the CNN along with Mohan Thota. I have also handled the live implementation of the custom built model. Averaged Datset is taken care by me for testing it with different parameters and splitting criteria

Srinivas Chellaboina: My part of the project is to work on the data preprocessing and generate different datasets to test on. Testing the models was taken care of by me. I also handled the original dataset with different spitting criteria and parameters.

6. References

[1] Ekman, Paul, and Harriet Oster. "Facial expressions of emotion." Annual review of psychology 30.1 (1979): 527-554.

[2] P Forgas, Joseph., and H Gordon. Bower. "Mood effects on person-perception judgments." Journal of personality and social psychology 53.1 (1987): 53.

[3] A. Mehrabian, "Communication without words," Psychol. Today,vol. 2, no. 4, pp. 53–56, 1968.

[4] Lv Y, Feng Z, Xu C (2014) Facial expression recognition via deep learning. In: Smart Computing (SMARTCOMP), 2014 International Conference on. IEEE[5] Liu, F L C Y, 2015. Improving Steganalysis by Fusing SVM Classifiers for JPEG Images. IEEE, pp. 185-190.

[5] Mehendale, N. (2020, February 18). *Facial emotion* recognition using convolutional neural networks (FERC). SN Applied Sciences.

[6] Z. Yu and C. Zhang. Image based static facial expression recognition with multiple deep network learning. ICMI Proceedings.

[7] Kanade, T., Cohn, J. F., & Tian, Y. (2000). Comprehensive database for facial expression analysis. Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition (FG'00), Grenoble, France, 46-53.

[8] Lei Xu, Minrui Fei, Wenju Zhou, Aolei Yang (2019) *Face Expression Recognition Based on Convolutional Neural Network.*