Convolutional Neural Network To Classify Facial Expressions

Taher Yunus Lilywala - 2018B1A70609G

Shreyas Sarangi - 2018A7PS1023G

Aastha Bhargava - 2019A7PS0421G

Keyur Gondhalekar - 2018A7PS0118G

Convolutional Neural Networks or CNNs are a powerful tool to classify images. They work by taking the pixel activations of a given image and passing them through multiple layers perceptrons to train the weights of the model accordingly. In this Assignment, Facial Expressions are classified into 7 distinct categories using a 10 layered CNN. Data augmentation is used to create more training data and overfitting is further reduced by reduction of LR as validation accuracy plateaus. A validation accuracy of about 66% was achieved, with a validation loss of 0.9614 after 30 epochs.

Introduction

Among the various types of networks, convolutional neural networks (CNNs) are best suited for image categorization tasks. Firstly, CNNs reduce the number of parameters required, making them relatively quick at parameter tuning, which is useful given the numerous layers required for this task. They do not require as much memory, either, due to coefficients being used at different locations across space. Having been developed with image classification in mind, CNNs are particularly efficient at picking out various features in images (such as a camera lens-caused shape change, or features more central to images such as small facial expression shifts) and use them/ignore them as necessary.[1,2]

Motivation

The given task is to use a neural network to classify images of human facial expressions. There are 7 categories in the classification problem- Angry, Disgust, Fear, Happy, Sad, Surprise, and Neutral. We aim to build a model to learn differentiation of these expressions, supervised by our training data, and classify images as one of the seven expressions. For aforementioned reasons, CNNs are well suited for this task.

Solution

The data provided is divided into 7 different classes of emotions – Angry, Disgust, Fear, Happy, Sad, Surprise, and Neutral.

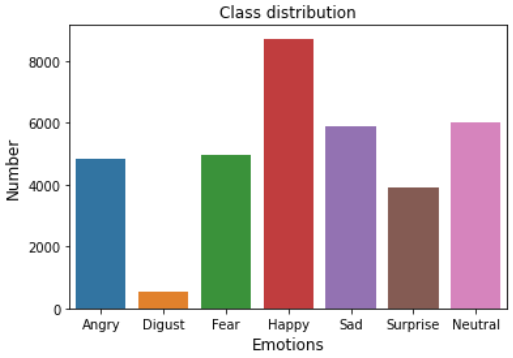

Fig 1 Frequency of Distinct Facial Expressions

The dataset was split into 80% training data and 20% validation data. The pixel sequence of each picture was reshaped to (-1,width, height,1)where width=height=48 and was also normalized to fit into the range of 0-1 by dividing each value by 255. The emotion represented by each picture was one hot encoded.

The image size was kept 48×48 and the number of epochs that were run were 30.

Layers used:

We had 3 different parts to the model, a convolution layer set, a flattening layer, and a dense layer set. The input layer was a convolutional layer 128 filters. After each layer we add a normalization layer. After 2 such convolution layers we added a Max pooling layer. Three more convolution layers were added, with the last 2 having 256. Another Max pooling layer was used, and then the model was flattened using a flattening layer. 2 dense layers with 128 and 64 filters were added. The final

output layer was a dense layer with 7 filters, one for each of the emotions. All these layers had relu activation, while the output layer had Softmax activation. Normalisation between layers encourages faster optimization of parameters and independent development of layers that is resistant to irrelevant variations in data.

Before settling on this final combination of layers, other variations we had tried had varying numbers of convolution layers and placement of batch normalization. The best results were found by applying normalization after every layer.[3]

While compiling the model, the adam optimizer was used and the loss was calculated with categorical cross entropy.

Initially the model was run for 50 epochs, but it was observed that the loss wasn't reducing over time. The epochs were then reduced to 30 without a significant loss in accuracy but loss was better.

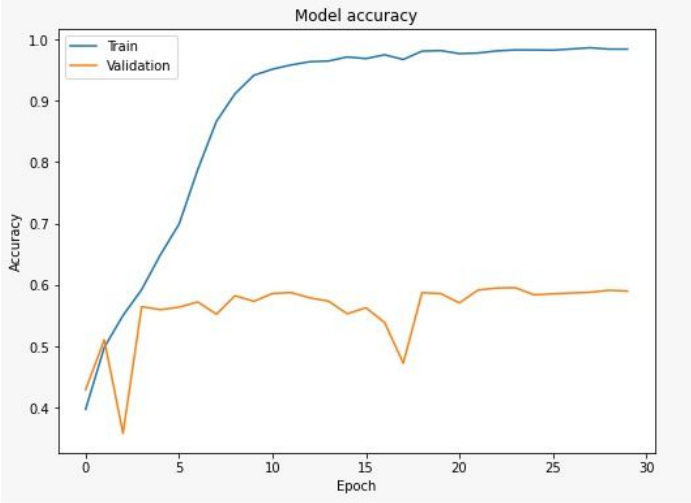

Fig 2 Initial Model Accuracy and Loss

Our model initially had significant overfitting as seen by training accuracy 0.985 and validation accuracy 0.59. In order to prevent this, we used data augmentation and reduced learning rate on plateau of validation accuracy. For data augmentation we used keras ImageDataGenerator function.[4] The images were rotated, zoomed, flipped, and had their sizes changed. For reducing LR we used the ReduceLROnPlateau function.[5] This led to significant increase in validation accuracy.

After these measures,

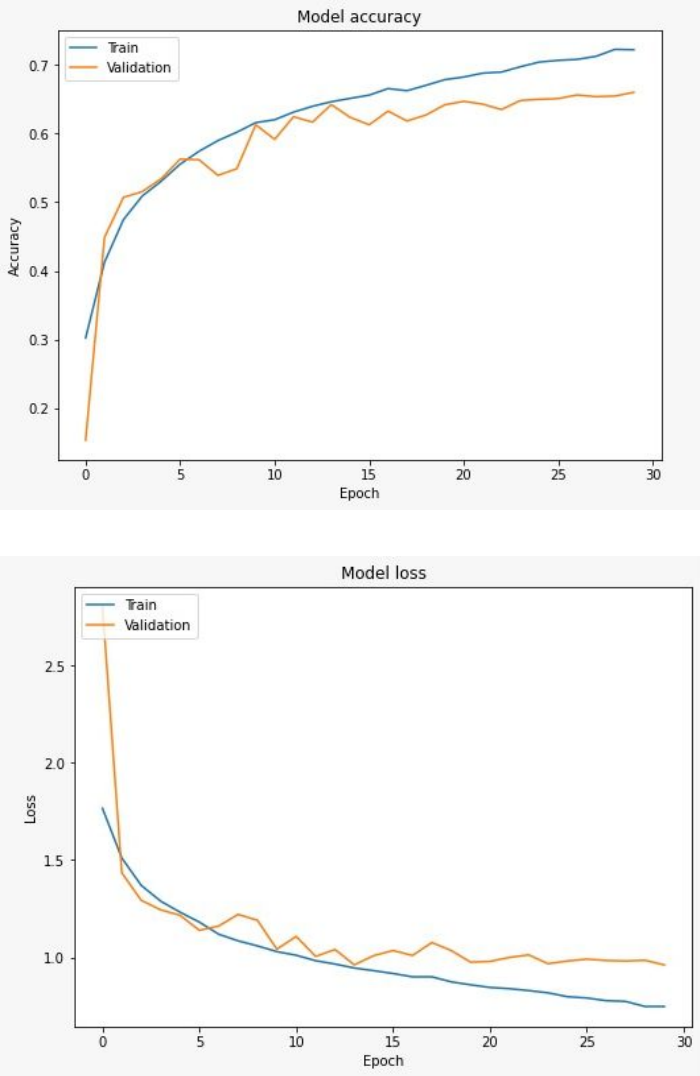

Fig 3 Model Accuracy and Loss lAfter Data Augmentation, Measures to prevent Overfitting

As can be seen, validation accuracy is much higher and more similar to train accuracy with these measures.

Result/Analysis

The final accuracy of the validation set is 0.6601(66%).

The final validation loss is 0.9614.

The confusion matrix is

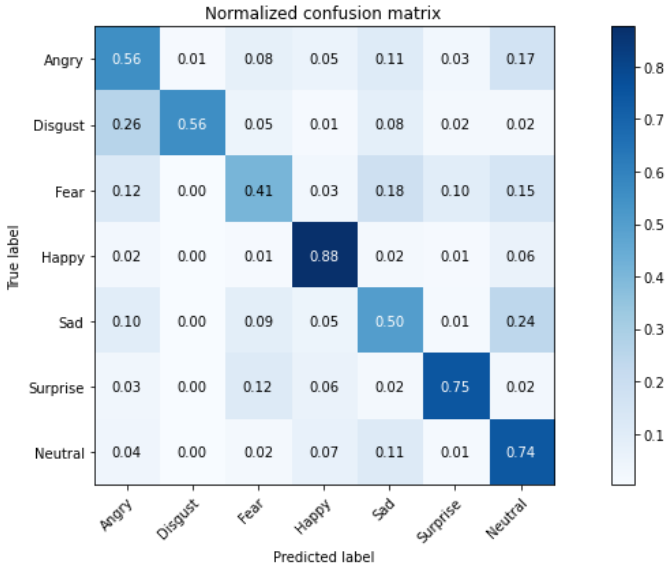

Fig 4 Confusion matrix from validation data

As can be seen, Happiness is well differentiated from other emotions. Surprise is most confused with fear, sadness with fear, disgust with anger. This is partially due to happiness having the highest number of training examples and fear the least, but it should not be surprising that emotions confused by the model are considered to be similarly expressed by humans, for example, disgust and anger.

The final weights are given in the Group-10_Assignment.h5 file.

Conclusion

Over the course of building this model, we experimented with different orders of layers for our CNN and observed the ability of CNNs in identifying relevant repeating features and their relative displacement for purposes of image classification. We learnt the importance of data augmentation and LR reduction.

References:

[1]Google Colabotary Documentation on CNNs

[2]Towards Data Science- Understanding CNNs

[3]https://www.kaggle.com/lxyuan0420/facial-expression-recognition-using-cnn

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

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