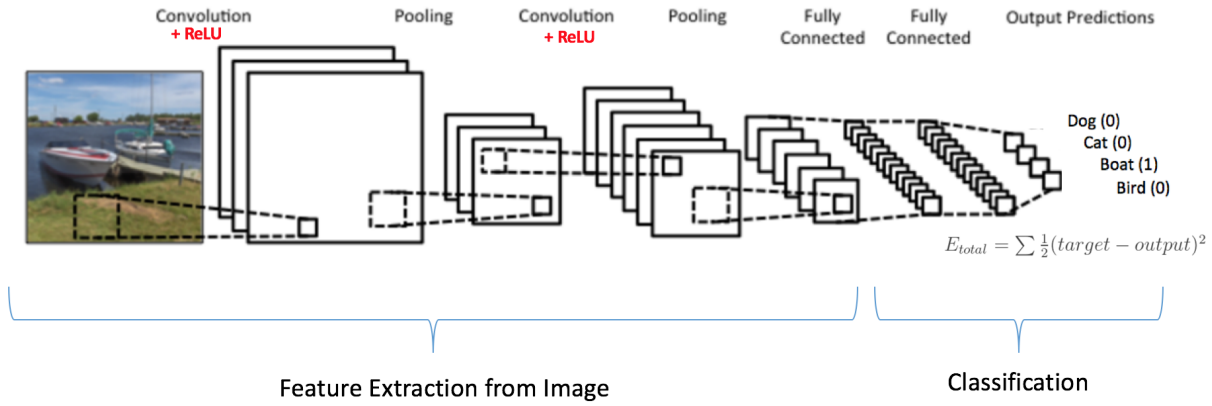


Emotion Detection With Keras and Convolutional Networks

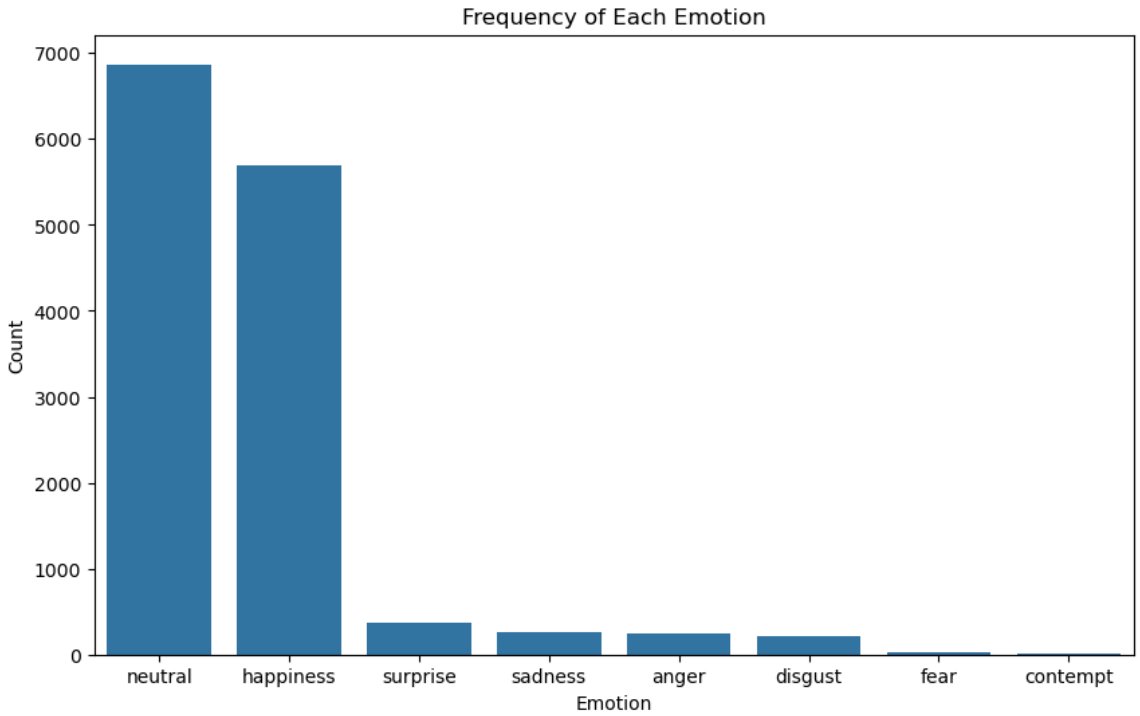
By Chris Cho, Lucas Belvedere, and Tanis Sarbatananda

Introduction

Our team has chosen to work with the emotion detection dataset, which is a dataset that contains approximately thirteen thousand images of individuals' faces in grayscale. Because we were working with the emotion detection, we have chosen to apply a convolutional neural network, as they are neural networks that are devoted to work with an image as an input.

Visualization

You will find some images below that will assist us interpret the data.

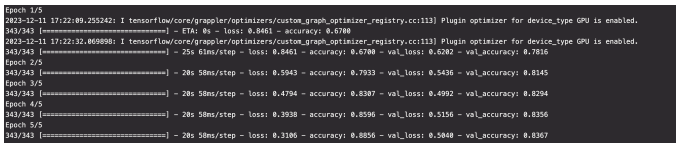


The number of emotion counts. So there are 8 different emotions. However, the dataset has bias of neutral and happiness.

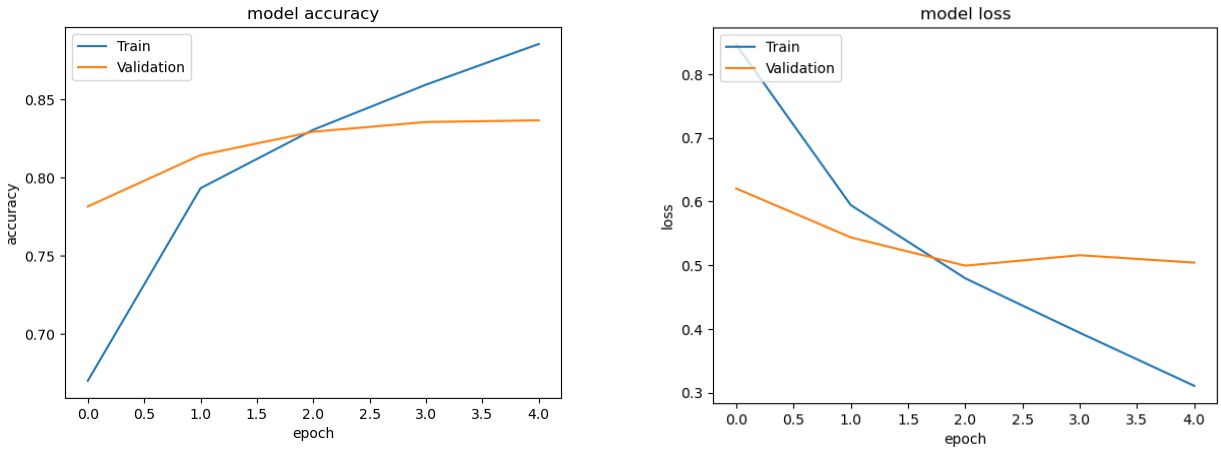
```
 Layer (type)                 Output Shape              Param #
=================================================================
 conv2d_6 (Conv2D)            (None, 176, 176, 32)      832

 max_pooling2d_6 (MaxPooling  (None, 88, 88, 32)       0
 2D)

 conv2d_7 (Conv2D)            (None, 85, 85, 64)        32832

 max_pooling2d_7 (MaxPooling  (None, 42, 42, 64)       0
 2D)

 conv2d_8 (Conv2D)            (None, 40, 40, 128)       73856

 max_pooling2d_8 (MaxPooling  (None, 20, 20, 128)      0
 2D)

 flatten_2 (Flatten)          (None, 51200)             0

 dense_4 (Dense)              (None, 128)               6553728

 dropout_2 (Dropout)          (None, 128)               0

...
Total params: 6,662,280
Trainable params: 6,662,280
Non-trainable params: 0
```

To classify the images respect to the emotion, output layer would have 8 nodes with softmax activation function.

Random imaged, resized by 180x180.



With 5 epochs it shows accuracy and loss.



Details

We utilized two CSV files as part of the project, namely 500_picts_satz.csv and legend.csv. The first CSV file contains five hundred data points with three columns including user id, image, and emotion. Emotion data is one of the most important information in the project. The list of emotions is a set of eight emotions, including neutral, happiness, sadness, disgust, fear, anger, surprise, and contempt. One detail we would like to mention is the fact that these emotions might be ambiguous in some cases, as our group members were looking at some images and realized that someone might have a neutral emotion, but the data states the emotion is disgust, for example. In addition, one of the emotions, contempt, has only nine images with that label.

At first, our group thought about changing the images to vectors that would have a value between 0 and 255 (grayscale) and, instead of using a convolutional neural network, utilizing a normal neural network that would take n inputs (number of pixels in each image) and work with it to predict the emotion of each face. However, we ended up using convolutional neural networks from class. Moreover, because of the grayscale, we normalize the data by dividing it by 255.

We ended up choosing for TensorFlow and Keras, due to its simplicity in programming a convolutional neural network. With about 15 to 40 lines of code, you could write down the neural network with this library. Because it was our first time coding a neural network, even utilizing Keras was quite challenging.

Biased Model At First

For our model, in most cases, the neural network would predict either neutral or happiness. As we can see from the graph above, these are the emotions that occur the most in the dataset. That means our model were biased and underfitting. We can observe that the model is not paying attention to enough details that it should from the images to predict the right emotions. But, as mentioned before, because of ambiguity, the model would chose an emotion that would make even more sense sometimes.

The emotion presented in the dataset for this person in the right is disgust. But as our group members discussed about it, this individual seems to be more neutral than disgusted. So, having a biased model in this case actually worked well due to ambiguity. However, we still worked to fix the bias and try to approach the right labels.

The emotion presented in the dataset for this person is disgust, but our model predicted neutral.

For images with neutral or happiness labels, the model did a precise work.

Changing Amount of Channels

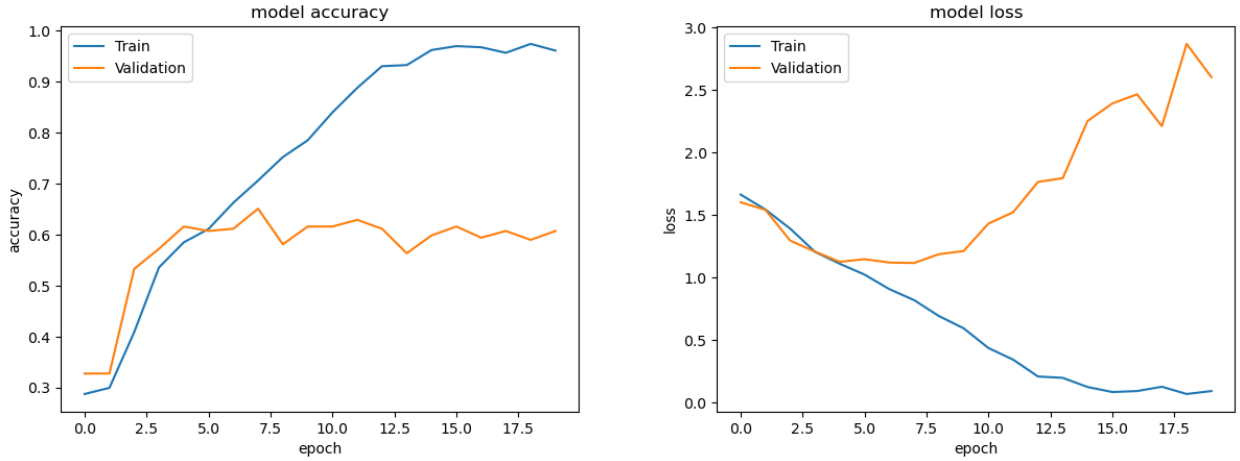
Another details we struggled with is how many channels (dimensions in the input data tensor) we have. Although the images were in grayscale, as mentioned before, we have no idea that changing it to grayscale would be better to avoid our model to have too much useless noise. As our neural network's goal is to predict emotions of black and white images, it was unnecessary to have three channels (RGB). Because of the lack of

complexity in our model, even with three channels, our model was still biased and predicting, for most images, either happiness or neutral. So this problem was not a major concern; yet, we fixed it to avoid overfitting in the future.

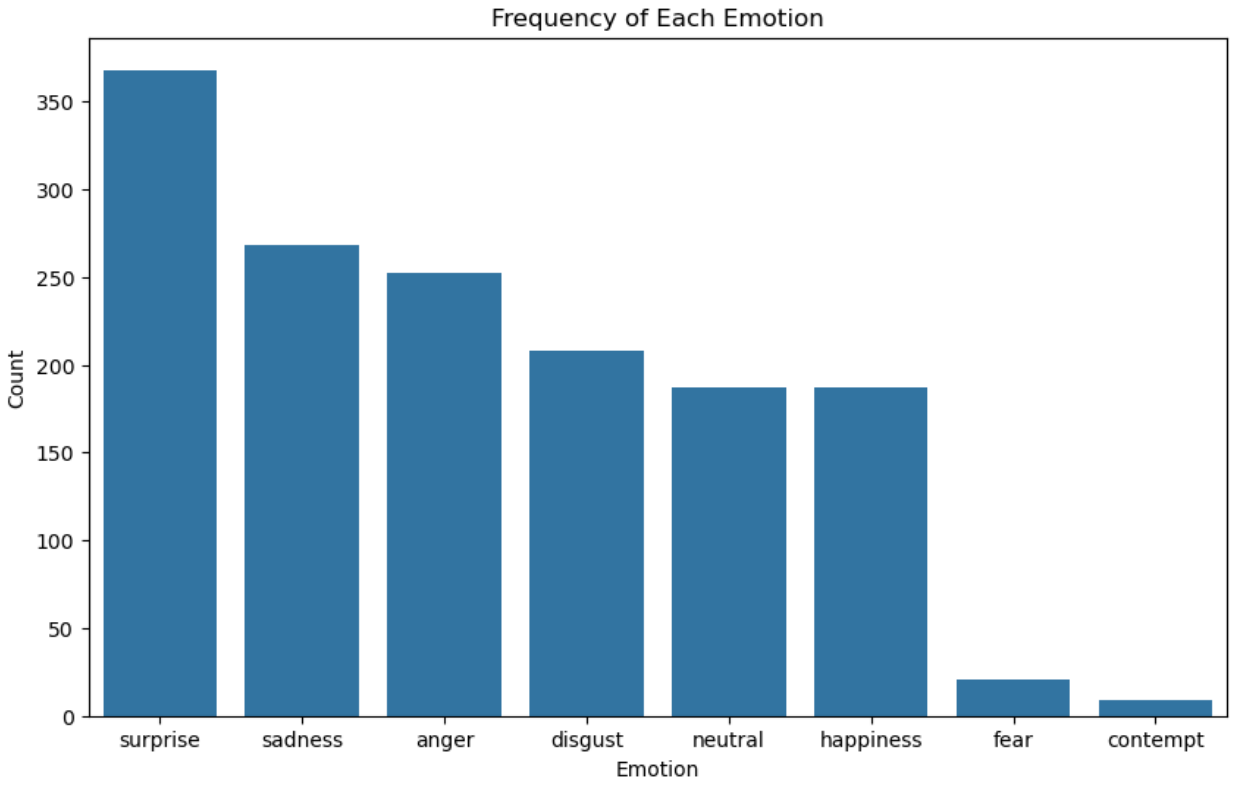
Optimizer ‘Adam’

Adam is a optimizer that works well for most models, including neural networks. Different than SGD (stochastic gradient descent,) Adam does not have a constant learning rate. Instead, it uses backpropagation to dynamically update the learning rates. We chose to stick with Adam because of the precision with the learning rates, and also because the majority of models in Keras are built with this optimizer.

After Down_Sampling



Since the data set that was given has high bias. The data set has too many datas for happiness and neutral.
We applied downsampling to change total number of data set from 13690 to 1500. With similar number of datas of each label.



The emotion of fear and contempt, they have only a few datas to train. Even though down scaling, the graph is skewed to the left.

Conclusion:

For this data set, initially, we had insanely large data for neutral and happiness. On the other hand, there were only a few data for fear and contempt.
Through the downsampling, we could adjust the data size of neutral and happiness. However, in order to fix fear and contempt, we need more images related to them. During this convolutional neural network, we needed to change some hyper-parameter to avoid overfitting. Also, we needed to apply some techniques to adjust overfitting such as dropout method and regularization.

We enjoyed working with convolutional neural networks, as it was fun looking at the predicted emotions and compare it to the labels of the csv data. In addition, we used opencv to predict our own emotions with pictures of our group members.