**Facial Recognition Capstone Project Report**

# **Introduction**

COVID 19 saw increasing use of telemedicine for various service providers. Many insurance companies began to support such options. Today, telemedicine is expanding to provide cheaper alternatives to healthcare, but with the use of such platforms, voice, image and other data can be used to improve patient outcomes.

Combining this kind of data with patient medical records can help in predicting mental disorders, like major depressive disorder.

Many studies have used image data from social media posts and using metrics like facial expression, lighting of the picture, number of people in the picture, etc. as indicators of depression.

# **Problem Identification**

There are gaps in methods to predict depression using text, image, etc. Training sets are also hard to come by that link image data to clinically diagnosed outcomes due to privacy restrictions. Identifying emotional trends is a first step using the [fer2013](https://www.kaggle.com/competitions/challenges-in-representation-learning-facial-expression-recognition-challenge/data) **dataset**:

**Can a model be built from scratch to be trained to effectively recognize emotion in facial expressions, while correctly inferring emotions from a new dataset, as a first step to detecting red flags for depression?**

The deep neural network of this project categorizes images into 6 categories (0=Angry, 1=Fear, 2=Happy, 3=Sad, 4=Surprise, 5=Neutral), the Disgust category was dropped due to it having few examples compared to the other categories for model training.

# **Data Exploration**

The icml\_face\_data contains two columns, **emotion**, **Usage** and **pixels**. The **emotion** column contains a numeric code ranging from 0-6 (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral), for the emotion that is present in the image.

The **Usage** column determines is made up of **Training**, **PrivateTest**, and **PublicTest.** The **Training** label will be for training data, the **PrivateTest** as validation data and as test data.

The pixels column contains a string surrounded in quotes representing an image that is currently flattened. The contents of this string a space-separated pixel value (0-255).

|  |
| --- |
|  |
| Figure 1: Histogram of Emotion Column |

From Figure 1 it can be seen that the dataset is unbalanced. The highest emotional category is 3(Happy) and the lowest 1(Disgust), which makes it a good category to be dropped.

The pictures in the dataset has a wide variety of genders, ages, and races in different postures and brightness. There is a total of 35,887 rows in total and 3 columns, the validation and testing datasets are 3,589 each out of 35,887.

# **Data Wrangling**

1. The 1(Disgust) emotion category was dropped due to it’s low count based on the histogram
2. Remap the emotion categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral) to (0=Angry, 1=Fear, 2=Happy, 3=Sad, 4=Surprise, 5=Neutral)
3. Assigned the Usage == “Training” to train\_df & dropped the Usage column
4. Assigned the Usage == “PrivateTest” to valid\_df & dropped the Usage column
5. Assigned the Usage == “PublicTest” to valid\_df & dropped the Usage column
6. Set the emotion columns equal to the y values for the train, valid, & test datasets
7. Convert each pixel row into a numpy array and set them as the x values for train, valid, and test datasets
8. Hot encode the array of y values for train, valid and test datasets
9. Normalize the x arrays by dividing by 255
10. Reshape the x arrays to 48 x 48 x 1
11. Create ImageGenerators to center and normalize the validation and test data fitted on the training data statistics with a batch size of 32
12. Create an ImageGenerators for the training data to rotate, zoom, etc. to help the data generalize better with batch size 32

# **Modeling**

The approach to modeling was to start with a simple base model with about two layers with 32 units each, 1 unit strip and 3 by 3 filters, then flattened into dense layers. This caused the model to underfit to about 43% accuracy(which was the original metric before changing it to recall) according to the training and validation data.

Then the complexity of the model was increased until the model overfitted then Dropout layers were added until the model neither overfitted or underfitted.

The initial activation function was **relu**, but **elu** and initializing weights to **he\_normal** with a **learning rate of .01** worked the best for this model.

The number of epochs was 200 and the steps\_per\_epoch= len(x\_train)/batch\_size

|  |
| --- |
|  |
| Figure 3: Recall Metric Graph |

|  |
| --- |
|  |
| Figure 4: Model Loss |

As seem from the above figure starts overfitting —­­ the blue line is the validation data and the orange lines are the training data. This occurs around 60% recall and goes up to about 65% recall.

|  |
| --- |
|  |
| Figure 5: Test Data Recall |

This is inline with the test data, that is separate from the validation data) which gets a recall score of 65%

|  |
| --- |
|  |
| Figure 6: Confusion Matrix |

From Figure 6, the Confusion Matrix shows that the false negatives are being reduced. It’s not the best performing model. There are many false negatives (around 40%) from Sad and Surprise labels, which is not for detecting depression since predicting someone is happy when they are sad could be a missed opportunity for treatment. The model also does not seem to be that good at detecting negative emotions in general. Part of this may be to labeling errors (pictured labeled happy but are really sad),the dataset needs to be further cleaned or a clustering analysis performed before modeling.

# **Conclusion & Recommendations**

Overall, the model did about 65% recall on test data, which was an improvement from about 40% originally from data augmentation and model tuning. The application this model could be applied to is a starting point for recognizing depression. The goal of this project was to build a deep learning model from scratch while classifying emotions from facial expressions. In practice, transfer learning would be used given the amount of time it takes to train neural networks.

## **Recommendations**

* Make a model that handles classifying negative emotions and one that handles positive emotions to prevent misclassification.
* Use a dataset that is better vetted (these datasets need special access due to privacy reasons)
* Use Unsupervised learning to build a better dataset
* Perform more model tuning
* Use transfer learning
* Adopt early stopping to prevent overfitting

# **Future Research**

There is a need to identify those who are depressed and need help overtime. I logical next step is to build a RNN to help predict emotions overtime from pictures/video from a few volunteers or open datasets.

# **Conclusion**

Based on the results of the random forest model, and model scenarios, there is enough support to consider raising the price. So, data such as the number of customers in a given year as maintenance costs, would make this model more robust, since skiing is also seasonal with peak periods and low periods.