FluCold Meter (Facial expression identification)

Srikar Nallan (https://youtu.be/xFOX2QAQVhY)

INTRODUCTION

An extension to facial recognition is facial expression recognition. Most of the current papers could classify Anger, Disgust, Fear, Happiness, Sadness, Surprise, Neutral expressions, though there could be small use cases for knowing these expressions like in Psych wards, Prisons etc where Anger could turn into violence, I could not think of use cases where these could be widely applied. However, I think the idea could be applied to identify flu symptoms. Though sneezing is not an emotion, it could be synonymous to facial expression, and identifying them in public areas like schools, transits etc could give real time information about flu break which could help in taking early precautions. Though I started project keeping this idea in mind, I had to drop 'sneeze' class due to various data issues I encountered. I decided to do basic facial expression recognition and list out the challenges I faced on identifying 'Sneezing' as expression.

Data

Started with FER+, Jaffe databases along with Google downloads (sneezes), however, due to data size mismatches which were restricting my model to run, reduced data to Jaffe database which consisted of 213 images with 7 facial expression posed by 10 Japanese female models, I added 'sneeze' from google downloaded images.

Data preprocessing:
Converted images to grey scale
Applied HAAR cascade which detects the face in the image
Size of each image is set to 350*350
Applied equal percentages of data labels in Train, CV and Test

Challenges

1. As there are no datasets available with 'sneeze' emotion I relied on Google to download images, as expected they are in different shape variations and I couldn't use them into model for errors like *"Value Error: Error when checking input: expected input_45 to have 4 dimensions but got array with shape (10,1).* I performed all the methods like normalization, resizing, reshaping, however, nothing seemed working. Most of my time is spent here, will post to Piazza.
2. Python 3.7 TensorFlow challenges - I keep getting errors like *"Layer called with an input that isn't a symbolic tensor keras'* . To resolve this downgraded to Python 3.6 and reran the model
3. Sneezing class expression is close to some of other classes like yawn, also, sometimes, just through face recognition we may not get clear picture
.

IMPLEMENTATION

Process Flow

Architecture
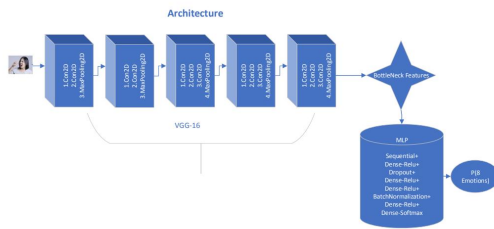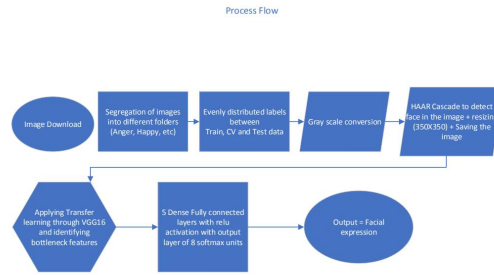
Loss and Accuracy
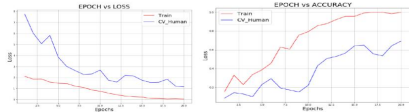
RESULTS & EVALUATION

Transfer Learning – Used VGG16 to acquire Bottleneck features of the images, this is where I faced challenge with 'sneeze' class due to image size mismatch and had to drop it to continue with the model

Model-

```
Layer (type)                  Output Shape       Param #
dense_35 (Dense)              (None, 512)        26214912
dropout_6 (Dropout)           (None, 512)        0
dense_36 (Dense)              (None, 256)        131328
dense_37 (Dense)              (None, 128)        32896
batch_normalization_7 (Batch  (None, 128)        512
dense_38 (Dense)              (None, 64)         8256
dense_39 (Dense)              (None, 7)          455
```
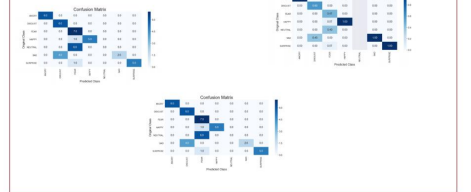
Dropout rate - Tried various dropout rates starting 0.5 without any considerable reduction in loss after 15 epochs, settled for 0.01 for better results

Loss - Used Cross entropy loss where result from Softmax is applied into loss, and below are figures for 20 epochs. I was able to reduce training loss from 2.10 to 0.04 and CV loss considerably from 7.7 to 1.1 and accuracy increased from 15% to 100% on training data and from 8% to 70% on CV data.

Result - Got **72.09%** accuracy on test data, below is confusion matrix:

REFERENCES

https://ieeexplore.ieee.org/document/8308363
https://github.com/thoughtworksarts/EmoPy
https://www.researchgate.net/publication/227031714_Facial_Expression_Recognition
https://medium.com/datadriveninvestor/real-time-facial-expression-recognition-f860dacfeb6a