Flu_Cold_Meter – Srikar Nallan

https://youtu.be/xFOX2QAQVhY

Abstract

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.

1. Introduction

Identifying flu symptom people in public areas based on facial expressions(shape) like sneezing could help in analyzing flu breakout in real time when used with already available tools, for example if 30%(could be different threshold) of people in the area sneezed based on cctv footages along with weather being low for past 3 days then chances of flu breakout in that area is high to take further precautions.

2. Process Flow:

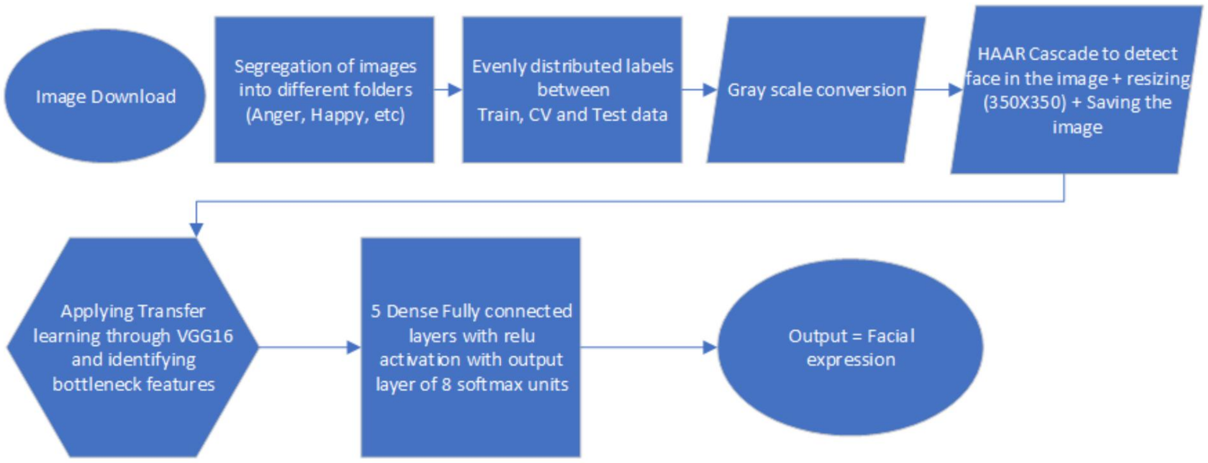
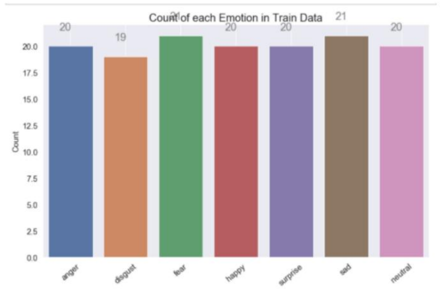


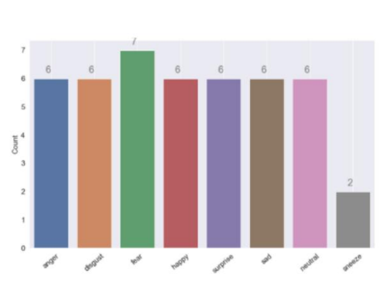
Image Download
Segregation of images into different folders (Anger, Happy, etc)
Evenly distributed labels between Train, CV and Test data
Gray scale conversion
HAAR Cascade to detect face in the image + resizing (350X350) + Saving the image
Applying Transfer learning through VGG16 and identifying bottleneck features
5 Dense Fully connected layers with relu activation with output layer of 8 softmax units
Output = Facial expression

3. Data

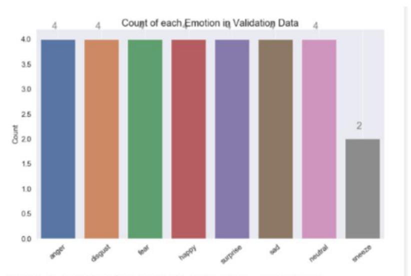
Started with FER+, Jaffe databases along with Google downloads (sneezes), however, due to data size mismatches which were restricting my model to run, reduced data to Jaffe database which consisted of 213 images with 7 facial expression posed by 10 Japanese female models, I added ‘sneeze’ from google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



```
Number of training data points in class fear = 21(14.8936%)
Number of training data points in class sad = 21(14.8936%)
Number of training data points in class anger = 20(14.1844%)
Number of training data points in class happy = 20(14.1844%)
Number of training data points in class neutral = 20(14.1844%)
Number of training data points in class surprise = 20(14.1844%)
Number of training data points in class disgust = 19(13.4752%)
```



```
Number of training data points in class fear = 7(15.5556%)
Number of training data points in class anger = 6(13.3333%)
Number of training data points in class disgust = 6(13.3333%)
Number of training data points in class happy = 6(13.3333%)
Number of training data points in class neutral = 6(13.3333%)
Number of training data points in class sad = 6(13.3333%)
Number of training data points in class surprise = 6(13.3333%)
Number of training data points in class sneeze = 2(4.4444%)
```

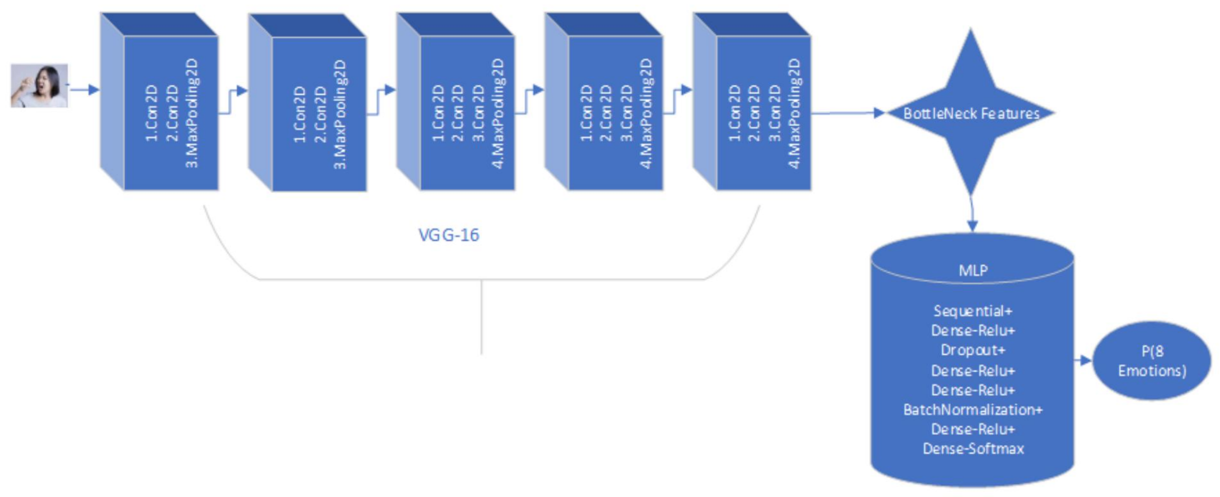


```
Number of training data points in class anger = 4(13.3333%)
Number of training data points in class disgust = 4(13.3333%)
Number of training data points in class fear = 4(13.3333%)
Number of training data points in class happy = 4(13.3333%)
Number of training data points in class neutral = 4(13.3333%)
Number of training data points in class sad = 4(13.3333%)
Number of training data points in class surprise = 4(13.3333%)
Number of training data points in class sneeze = 2(6.6667%)
----------------------------------------------------------------------------
---------
```

3.1. Data preprocessing:

a. Converted images to grey scale
b. Applied HAAR cascade which detects the face in the image
c. Size of each image is set to 350*350
d. Applied equal percentages of data labels in Train, CV and Test

4. Architecture –

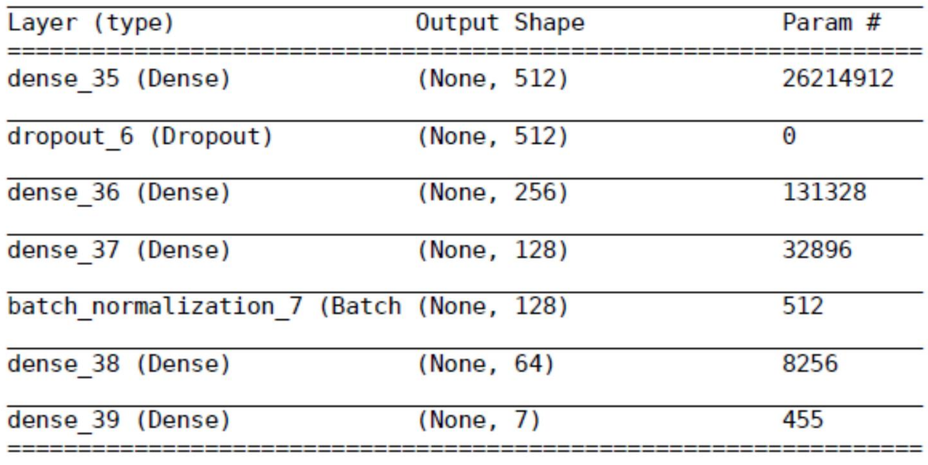

1.Con2D 2.Con2D 3.MaxPooling2D
1.Con2D 2.Con2D 3.MaxPooling2D
1.Con2D 2.Con2D 3.Con2D 4.MaxPooling2D
1.Con2D 2.Con2D 3.Con2D 4.MaxPooling2D
1.Con2D 2.Con2D 3.Con2D 4.MaxPooling2D
VGG-16
BottleNeck Features
MLP
Sequential+
Dense-Relu+
Dropout+
Dense-Relu+
Dense-Relu+
BatchNormalization+
Dense-Relu+
Dense-Softmax
P(8 Emotions)

5. 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

6. Model

I added MLP model with 5 Dense layers out of which 4 had Relu activation and last one with Softmax activation:



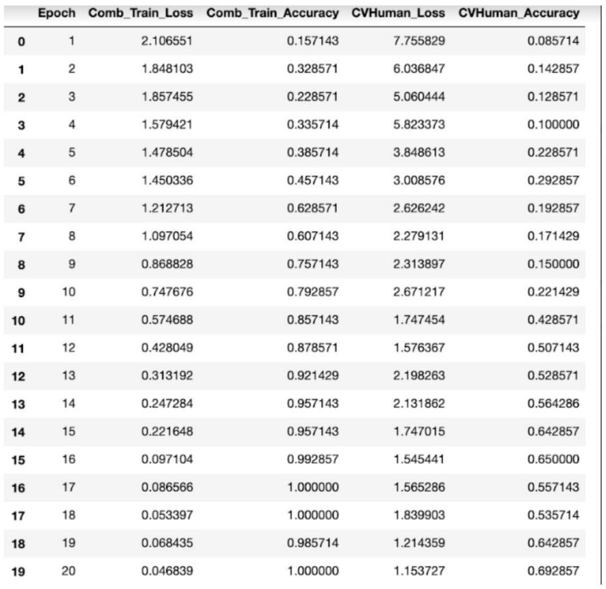
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_35 (Dense)             (None, 512)               26214912
_________________________________________________________________
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
=================================================================
```

6.2 Dropout rate-

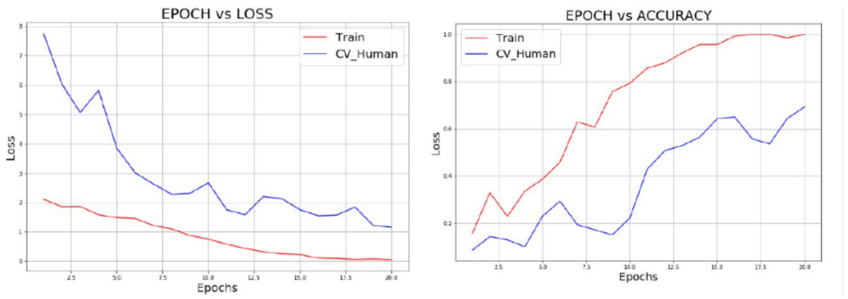
Tried various dropout rates starting 0.5 without any considerable reduction in loss after 15 epochs, settled for 0.01 for better results

6.1 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.

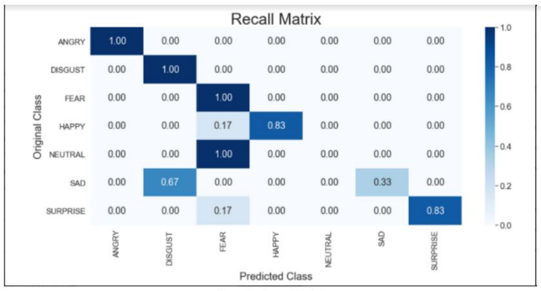


	Epoch	Comb_Train_Loss	Comb_Train_Accuracy	CVHuman_Loss	CVHuman_Accuracy
0	1	2.106551	0.157143	7.755829	0.085714
1	2	1.848103	0.328571	6.036847	0.142857
2	3	1.857455	0.228571	5.060444	0.128571
3	4	1.579421	0.335714	5.823373	0.100000
4	5	1.478504	0.385714	3.848613	0.228571
5	6	1.450336	0.457143	3.008576	0.292857
6	7	1.212713	0.628571	2.626242	0.192857
7	8	1.097054	0.607143	2.279131	0.171429
8	9	0.868828	0.757143	2.313897	0.150000
9	10	0.747676	0.792857	2.671217	0.221429
10	11	0.574688	0.857143	1.747454	0.428571
11	12	0.428049	0.878571	1.576367	0.507143
12	13	0.313192	0.921429	2.198263	0.528571
13	14	0.247284	0.957143	2.131862	0.564286
14	15	0.221648	0.957143	1.747015	0.642857
15	16	0.097104	0.992857	1.545441	0.650000
16	17	0.086566	1.000000	1.565286	0.557143
17	18	0.053397	1.000000	1.839903	0.535714
18	19	0.068435	0.985714	1.214359	0.642857
19	20	0.046839	1.000000	1.153727	0.692857

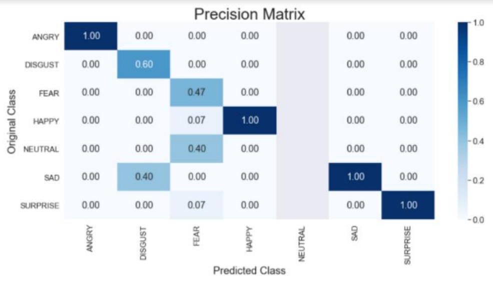


7. Results –

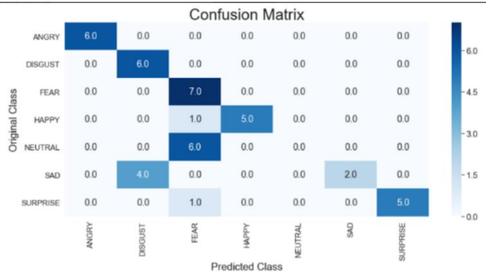
Got **72.09%** accuracy on test data, below is confusion matrix:



	ANGRY	DISGUST	FEAR	HAPPY	NEUTRAL	SAD	SURPRISE
ANGRY	1.00	0.00	0.00	0.00	0.00	0.00	0.00
DISGUST	0.00	1.00	0.00	0.00	0.00	0.00	0.00
FEAR	0.00	0.00	1.00	0.00	0.00	0.00	0.00
HAPPY	0.00	0.00	0.17	0.83	0.00	0.00	0.00
NEUTRAL	0.00	0.00	1.00	0.00	0.00	0.00	0.00
SAD	0.00	0.67	0.00	0.00	0.00	0.33	0.00
SURPRISE	0.00	0.00	0.17	0.00	0.00	0.00	0.83



	ANGRY	DISGUST	FEAR	HAPPY	NEUTRAL	SAD	SURPRISE
ANGRY	1.00	0.00	0.00	0.00		0.00	0.00
DISGUST	0.00	0.60	0.00	0.00		0.00	0.00
FEAR	0.00	0.00	0.47	0.00		0.00	0.00
HAPPY	0.00	0.00	0.07	1.00		0.00	0.00
NEUTRAL	0.00	0.00	0.40	0.00		0.00	0.00
SAD	0.00	0.40	0.00	0.00		1.00	0.00
SURPRISE	0.00	0.00	0.07	0.00		0.00	1.00



	ANGRY	DISGUST	FEAR	HAPPY	NEUTRAL	SAD	SURPRISE
ANGRY	6.0	0.0	0.0	0.0	0.0	0.0	0.0
DISGUST	0.0	6.0	0.0	0.0	0.0	0.0	0.0
FEAR	0.0	0.0	7.0	0.0	0.0	0.0	0.0
HAPPY	0.0	0.0	1.0	5.0	0.0	0.0	0.0
NEUTRAL	0.0	0.0	6.0	0.0	0.0	0.0	0.0
SAD	0.0	4.0	0.0	0.0	0.0	2.0	0.0
SURPRISE	0.0	0.0	1.0	0.0	0.0	0.0	5.0

8. 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

9. Future steps:

    1. As I started the build with focus on identifying sneezing class and most of the build is present, once I figure out the data reshape issue, will continue with the task
    2. Want to increase the database to thousands and run for better results

10. 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