Semester VIII: Endsem Presentation

Multimodal Emotion Classification using User’s Speech & Facial Expression

Under the guidance of: **Prof. U S Tiwari**

Introduction

Emotions are one of the most basic features that distinguish human beings from the machines and robots. Humans express their emotions through various media like facial expressions, their actions and most importantly speech. To bring the artificial intelligence close to humans, it becomes necessary to be able to recognize as well as generate emotions by machines. One of a huge challenge is developing computers that can effectively simulate human interaction. A lot of research has been done in the past years in this field. This motivates us to develop a system which can efficiently determine the mental state of the user by studying the speech and facial expressions of the user.

Scope of Project Work

- **Call Center** conversation may be used to analyze behavioral study of call attendants with the customers which helps to improve quality of service of a call attendant.

- In **Aircraft** cockpits, speech recognition systems trained to recognize stressed speech are used for better performance and can be useful in detecting emergency situations.

- It is Useful for enhancing the naturalness in speech based **human machine interaction**

.

- Interactive movie, storytelling & **E-tutoring** applications would be more practical, if they can adapt themselves to listeners or students emotional states.

- **Home Automation** – Home Automation Systems capable of sensing emotion can avert risk of depression by changing ambient lighting, playing cheerful music, ordering favorite food items from nearby bakery.

Scope of Project Work

By doing this project, the followings are achieved:

a) Development of separate modules to detect the emotion of the person using Facial Expression and Speech Data

   i. **Module 1**: Emotion Classification using Facial Expressions

   ii. **Module 2**: Emotion Classification using Speech Data

b) Diffusing the results of speech and facial expression classifier to enhance the accuracy.

c) Using effective machine learning algorithms to achieve the above mentioned task.

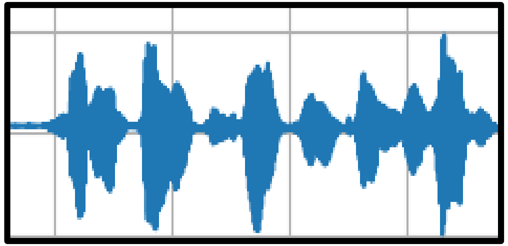
d) Comparing the accuracies obtained by different methodologies and choosing the best one

The Data-Set

Audio Data

- **Source:** Surrey Audio-Visual Expressed Emotion (SAVEE)[3]
- **480** British English utterances recorded by 4 male actors 7 emotions—happy, angry, sad, disgusted, fearful, surprised, and neutral
- **Source: Recorded IIIT-A DATA**
- **853** audio clips by **82** actors (11 female, 71 male) expressing the 7 seven emotions.

All the audio clips are captured at a frequency of 44100 Hz. The length of each audio file is around **2.0 ~ 3.0 seconds** and are stored in wave (.wav) format. The total number of audio clips are **1333,** which are further divided into **988** for training and **345** for validating the model.



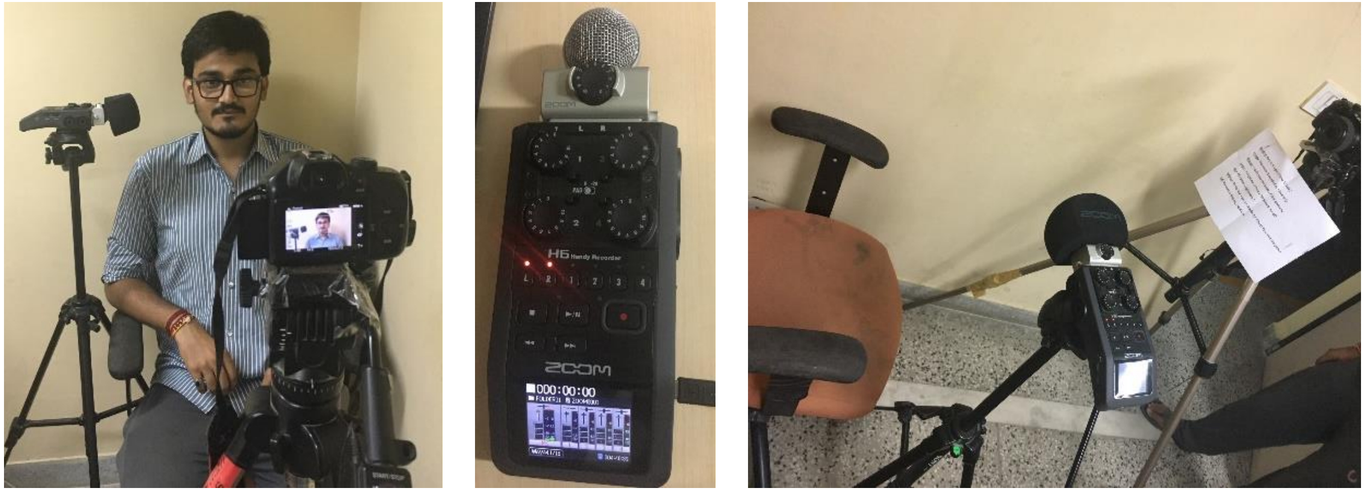
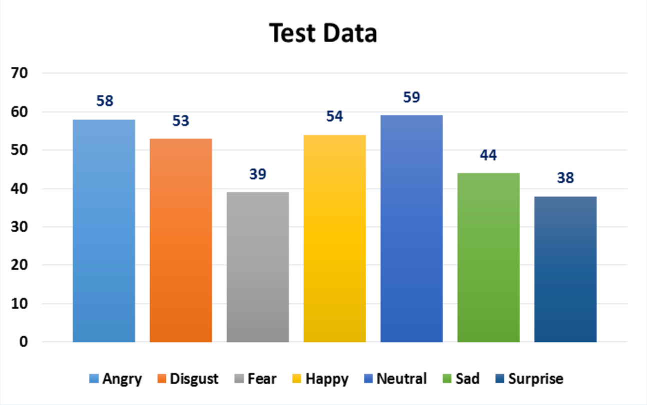


Fig. Recording for Dataset


Test Data

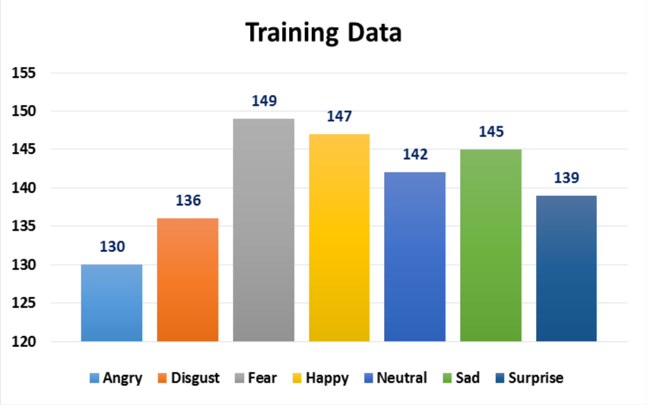
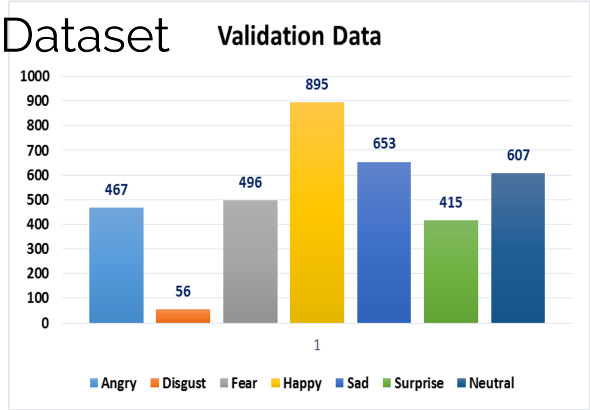
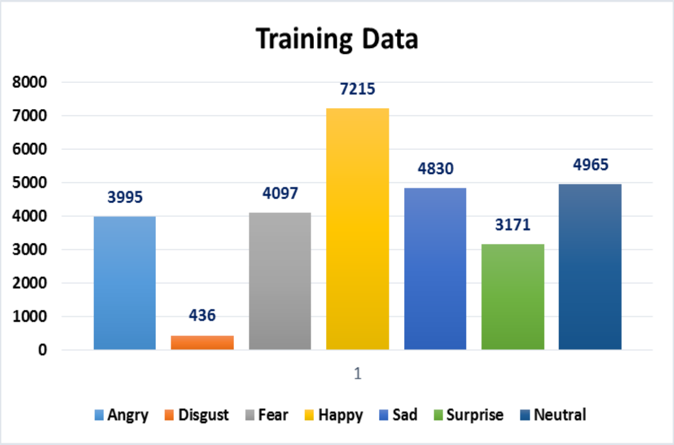

Training Data

Image Data

- 32,298 grayscale images
- Each image of 48 x 48 pixel
- 28,709 Training + 3,589 validation
- 7 emotion classes
- Source: FER2013 Dataset

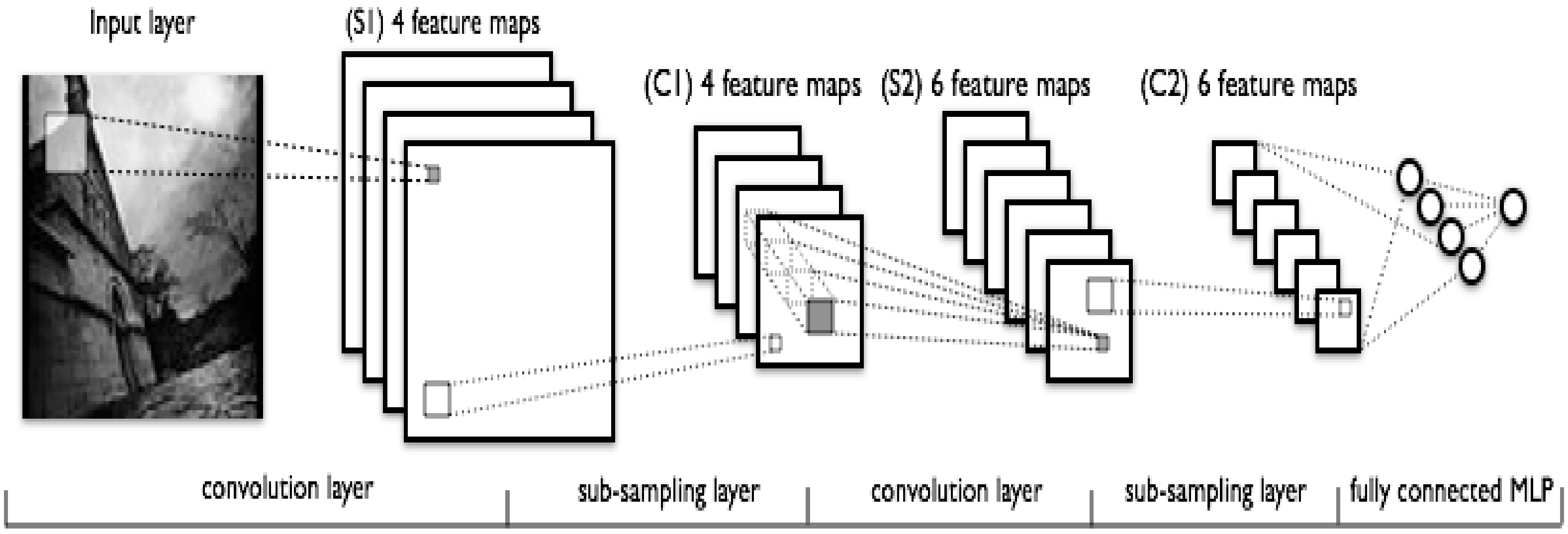
0=Angry
1=Disgust
2=Fear
3=Happy
4=Sad
5=Surprise
6=Neutral.



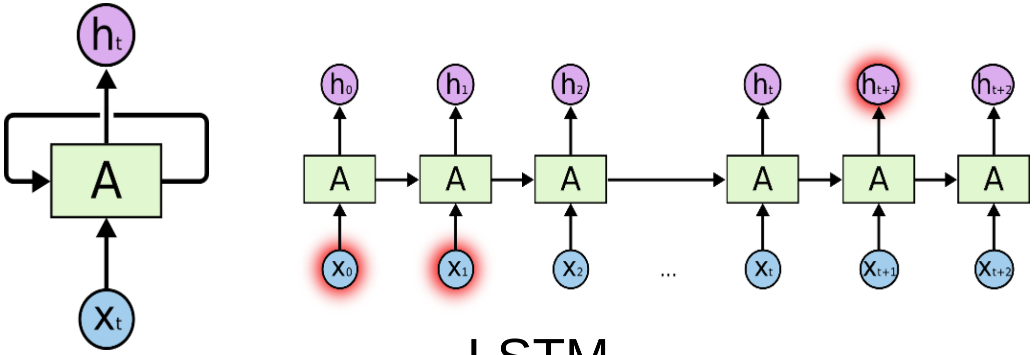


Approach: Deep Learning

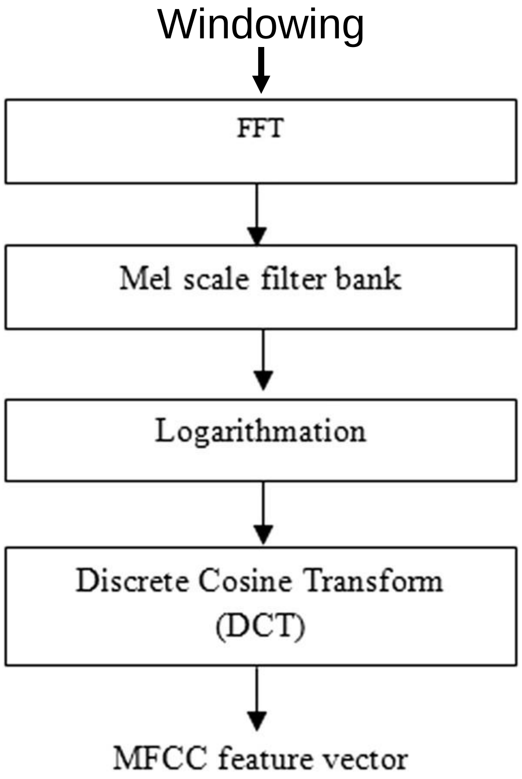
- **Convolutional Neural Network(CNN):** CNN take into consideration spatial information in an image
  - **Input Layer** will hold the raw pixel values of the image, in this case an image of width 32, height 32, and with three color channels R,G,B it would have dimensions 32 x 32 x 3.
  - **CONV Layer** will compute the output of neurons that are connected to local regions in the input, each computing a dot product between their weights and a small region they are connected to in the input volume. This may result in volume such as [32x32x12] if we decided to use 12 filters.
  - **RELU Layer** will apply an element wise activation function, such as the **max(0,x)** thresholding at zero. This leaves the size of the volume unchanged ([32x32x12]) and removes the negative intensities.
  - **POOL layer** will perform a down sampling operation along the spatial dimensions (width, height), resulting in volume such as [16x16x12] if we use a 2 x 2 pooling filter. Eg. 2 x 2 max pool filter.
  - **FC Layer**(i.e. fully-connected) layer will compute the class scores, resulting in volume of size [1x1x10], where each of the 10 numbers correspond to a class score, such as among the 10 categories.



Why CNN??- The basic need of CNN aroused for image recognitions problems as in case of images the no of parameters in input layer become large and in order to make the recognizing system efficient the number of hidden layers in the neural network are also large, due to which the effect to the weights of initial hidden layers is not much during back propagation. This increases the number of iterations needed to adjust the weights in order to obtain good accuracy from the system, thereby increasing the computation power.



LSTM



Windowing

FFT

Mel scale filter bank

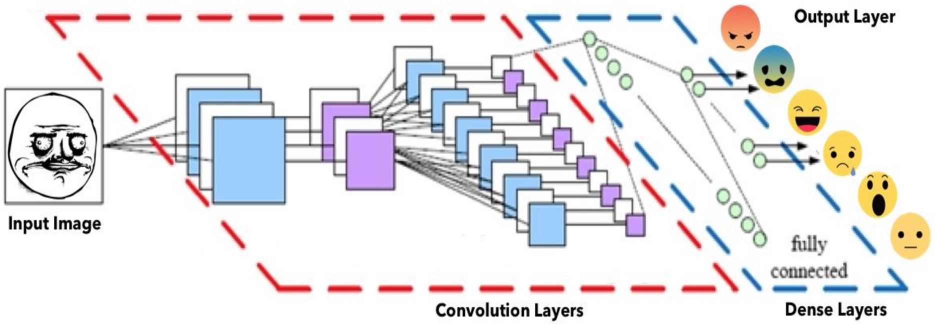
Logarithmation

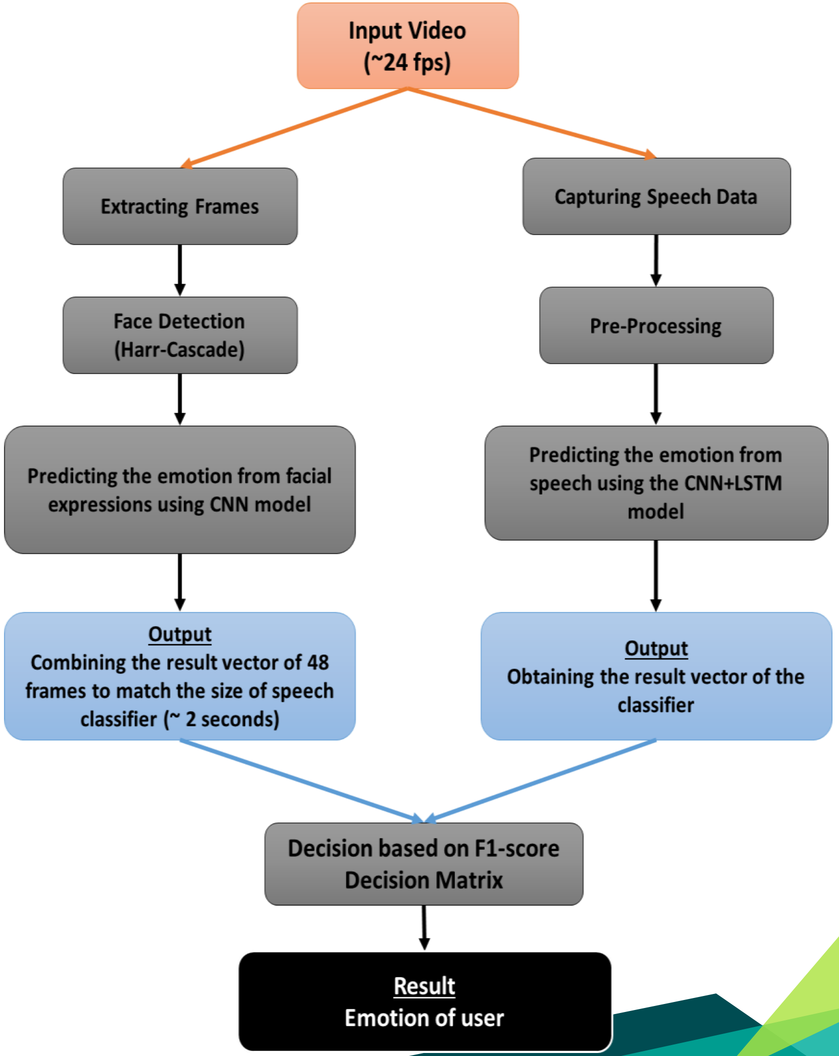
Discrete Cosine Transform (DCT)

MFCC feature vector

MFCC

Methodology





Input Video (~24 fps)

Extracting Frames

Face Detection (Harr-Cascade)

Predicting the emotion from facial expressions using CNN model

Output
Combining the result vector of 48 frames to match the size of speech classifier (~ 2 seconds)

Capturing Speech Data

Pre-Processing

Predicting the emotion from speech using the CNN+LSTM model

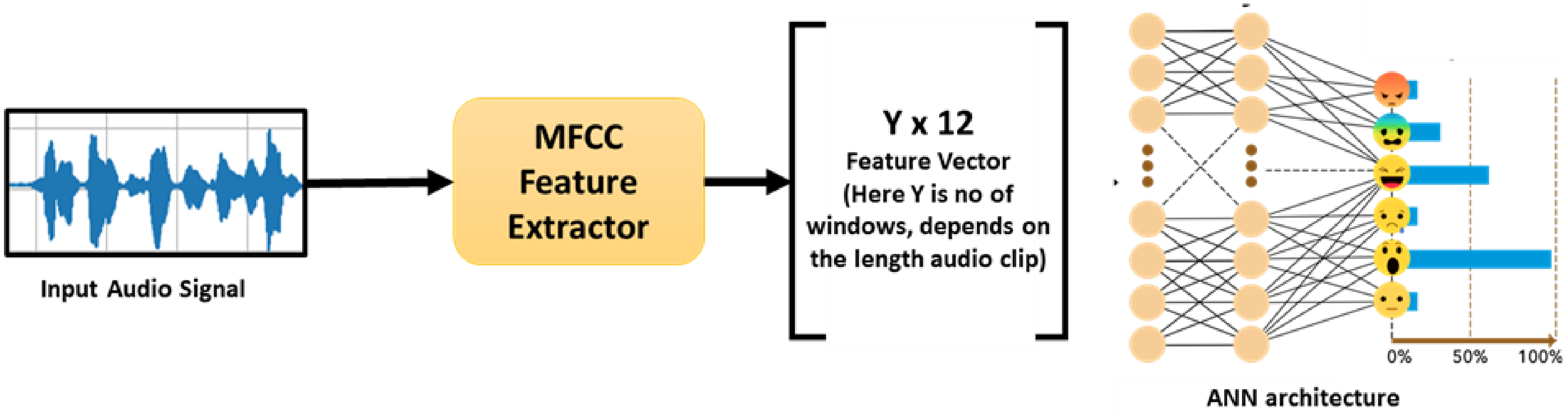
Output
Obtaining the result vector of the classifier

Decision based on F1-score Decision Matrix

Result
Emotion of user

Model 1: Using MFCC

Accuracy Obtained: **50.12%**



Input Audio Signal

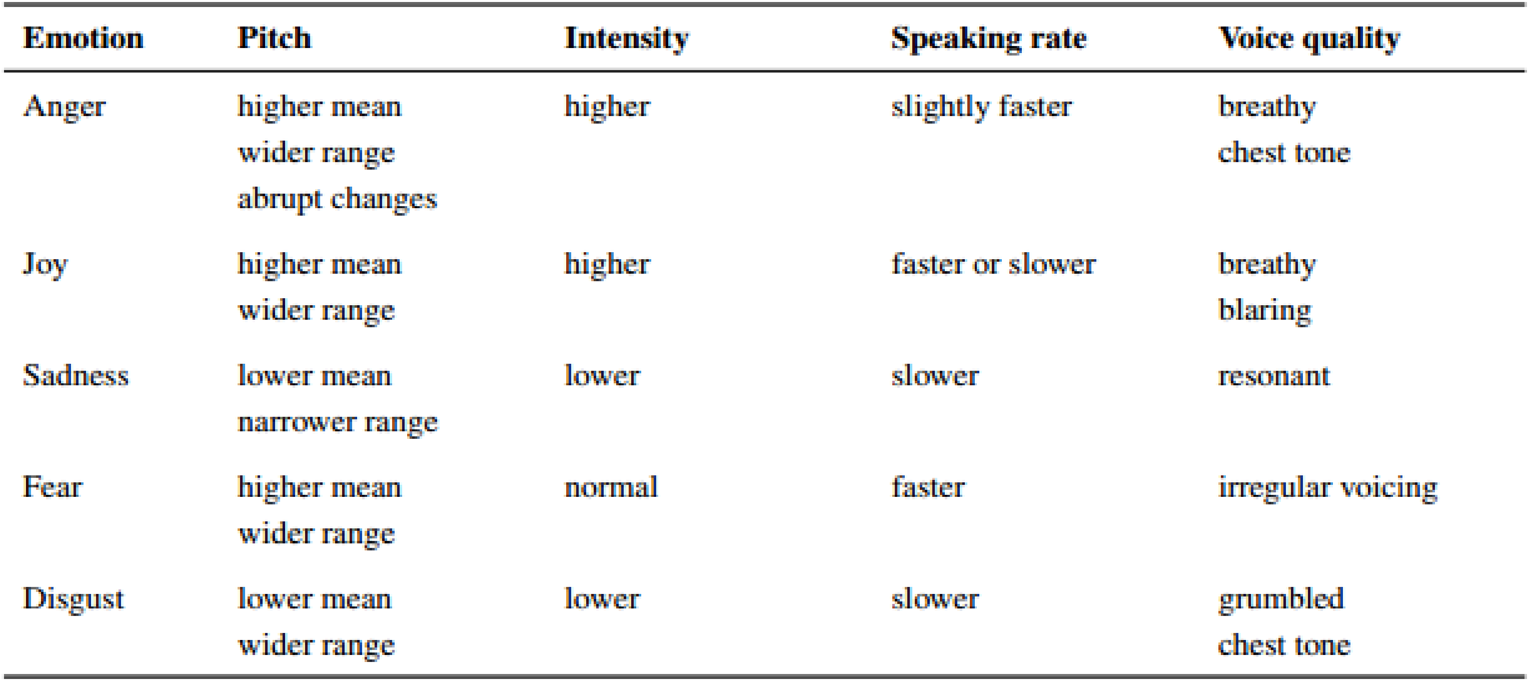
MFCC Feature Extractor

Y x 12

Feature Vector

(Here Y is no of windows, depends on the length audio clip)

ANN architecture



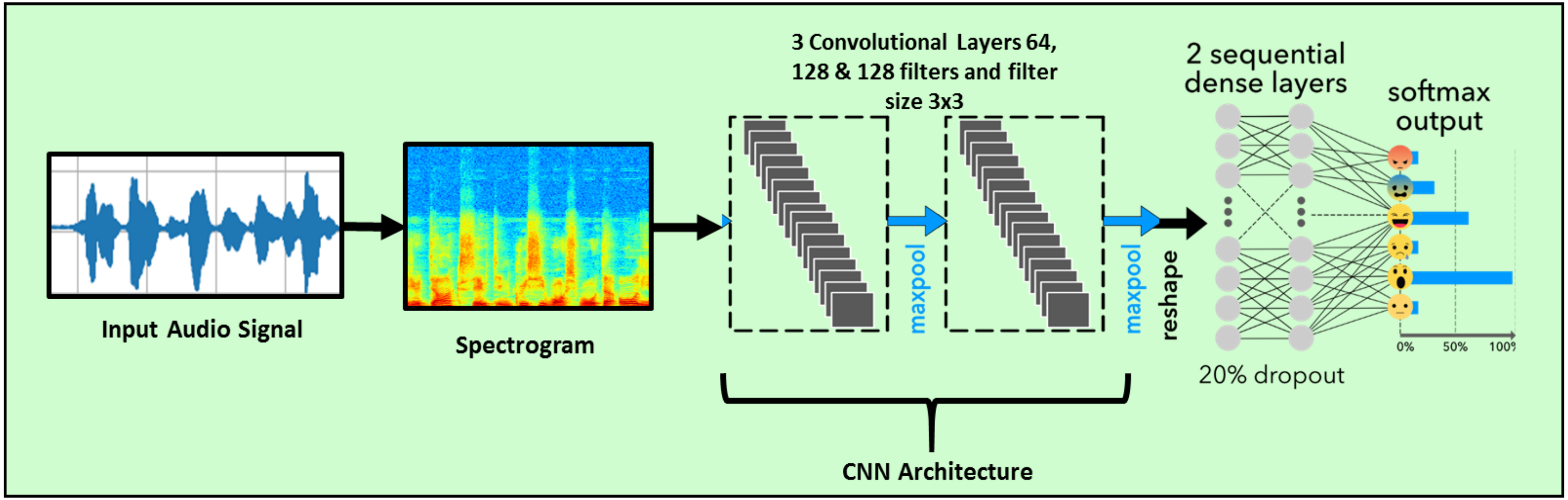
Emotion	Pitch	Intensity	Speaking rate	Voice quality
Anger	higher mean wider range abrupt changes	higher	slightly faster	breathy chest tone
Joy	higher mean wider range	higher	faster or slower	breathy blaring
Sadness	lower mean narrower range	lower	slower	resonant
Fear	higher mean wider range	normal	faster	irregular voicing
Disgust	lower mean wider range	lower	slower	grumbled chest tone

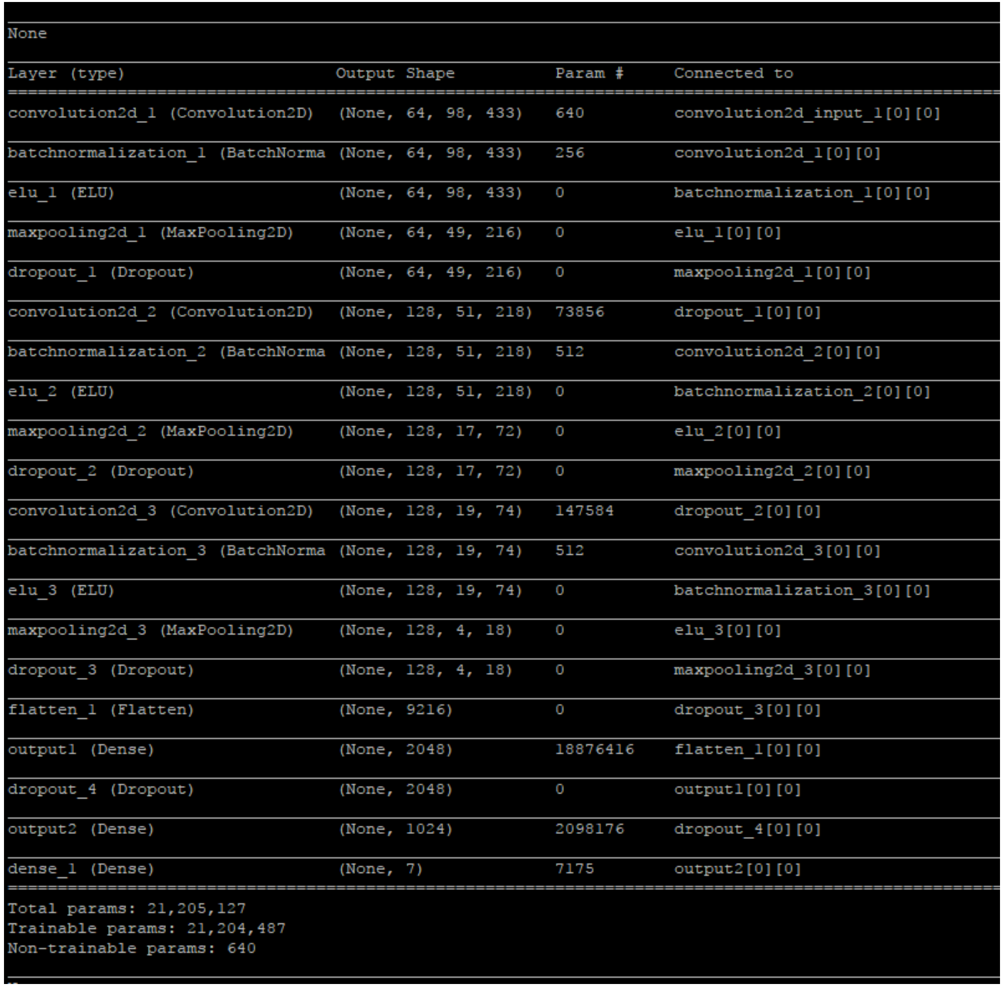
Variation of various acoustic variables

[5]R. Murray and J. L. Arnott, “Toward the simulation of emotion in synthetic speech: A review of the literature on human vocal emotion”

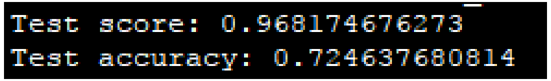
Model 2: CNN

Accuracy Obtained: **72.46%**

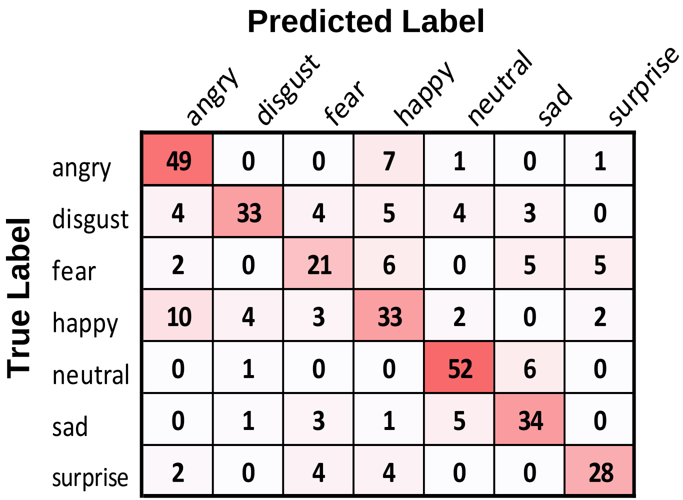




```
None
________________________________________________________________________________________________________
Layer (type)                     Output Shape          Param #     Connected to
========================================================================================================
convolution2d_1 (Convolution2D)  (None, 64, 98, 433)   640         convolution2d_input_1[0][0]
________________________________________________________________________________________________________
batchnormalization_1 (BatchNorma (None, 64, 98, 433)   256         convolution2d_1[0][0]
________________________________________________________________________________________________________
elu_1 (ELU)                      (None, 64, 98, 433)   0           batchnormalization_1[0][0]
________________________________________________________________________________________________________
maxpooling2d_1 (MaxPooling2D)    (None, 64, 49, 216)   0           elu_1[0][0]
________________________________________________________________________________________________________
dropout_1 (Dropout)              (None, 64, 49, 216)   0           maxpooling2d_1[0][0]
________________________________________________________________________________________________________
convolution2d_2 (Convolution2D)  (None, 128, 51, 218)  73856       dropout_1[0][0]
________________________________________________________________________________________________________
batchnormalization_2 (BatchNorma (None, 128, 51, 218)  512         convolution2d_2[0][0]
________________________________________________________________________________________________________
elu_2 (ELU)                      (None, 128, 51, 218)  0           batchnormalization_2[0][0]
________________________________________________________________________________________________________
maxpooling2d_2 (MaxPooling2D)    (None, 128, 17, 72)   0           elu_2[0][0]
________________________________________________________________________________________________________
dropout_2 (Dropout)              (None, 128, 17, 72)   0           maxpooling2d_2[0][0]
________________________________________________________________________________________________________
convolution2d_3 (Convolution2D)  (None, 128, 19, 74)   147584      dropout_2[0][0]
________________________________________________________________________________________________________
batchnormalization_3 (BatchNorma (None, 128, 19, 74)   512         convolution2d_3[0][0]
________________________________________________________________________________________________________
elu_3 (ELU)                      (None, 128, 19, 74)   0           batchnormalization_3[0][0]
________________________________________________________________________________________________________
maxpooling2d_3 (MaxPooling2D)    (None, 128, 4, 18)    0           elu_3[0][0]
________________________________________________________________________________________________________
dropout_3 (Dropout)              (None, 128, 4, 18)    0           maxpooling2d_3[0][0]
________________________________________________________________________________________________________
flatten_1 (Flatten)              (None, 9216)          0           dropout_3[0][0]
________________________________________________________________________________________________________
output1 (Dense)                  (None, 2048)          18876416    flatten_1[0][0]
________________________________________________________________________________________________________
dropout_4 (Dropout)              (None, 2048)          0           output1[0][0]
________________________________________________________________________________________________________
output2 (Dense)                  (None, 1024)          2098176     dropout_4[0][0]
________________________________________________________________________________________________________
dense_1 (Dense)                  (None, 7)             7175        output2[0][0]
========================================================================================================
Total params: 21,205,127
Trainable params: 21,204,487
Non-trainable params: 640
________________________________________________________________________________________________________
```



```
Test score: 0.968174676273
Test accuracy: 0.724637680814
```



Predicted Label

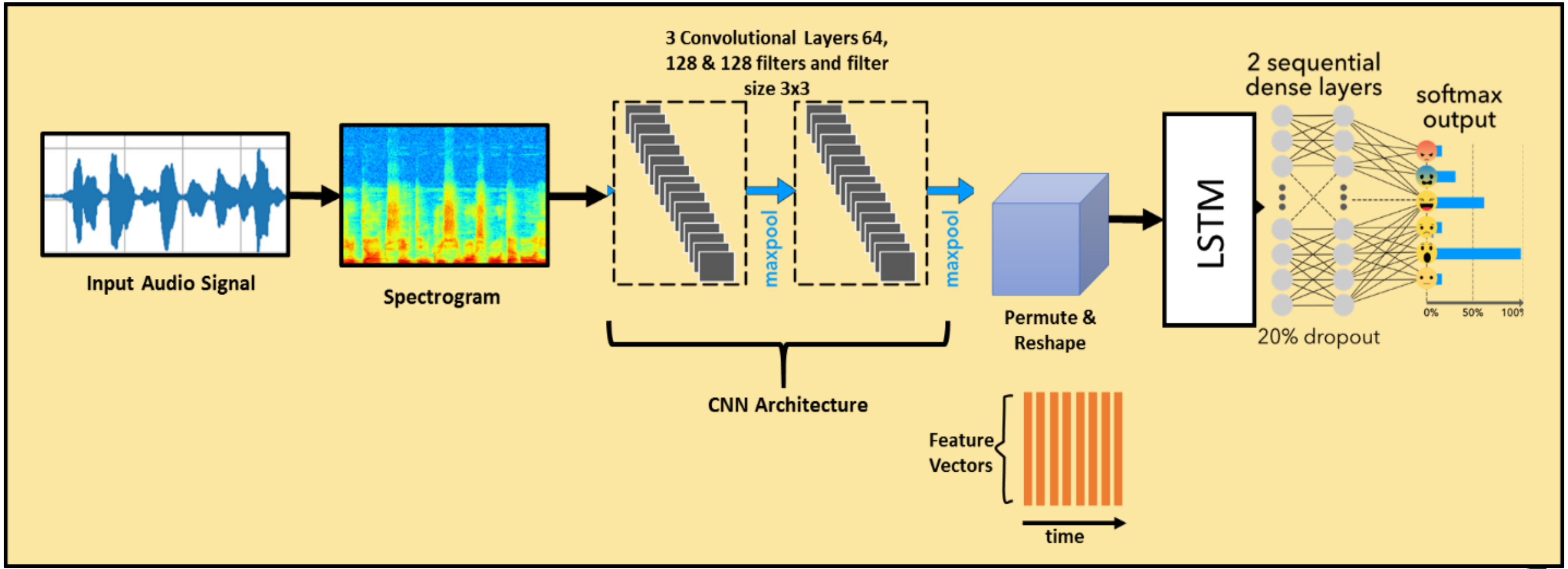
True Label

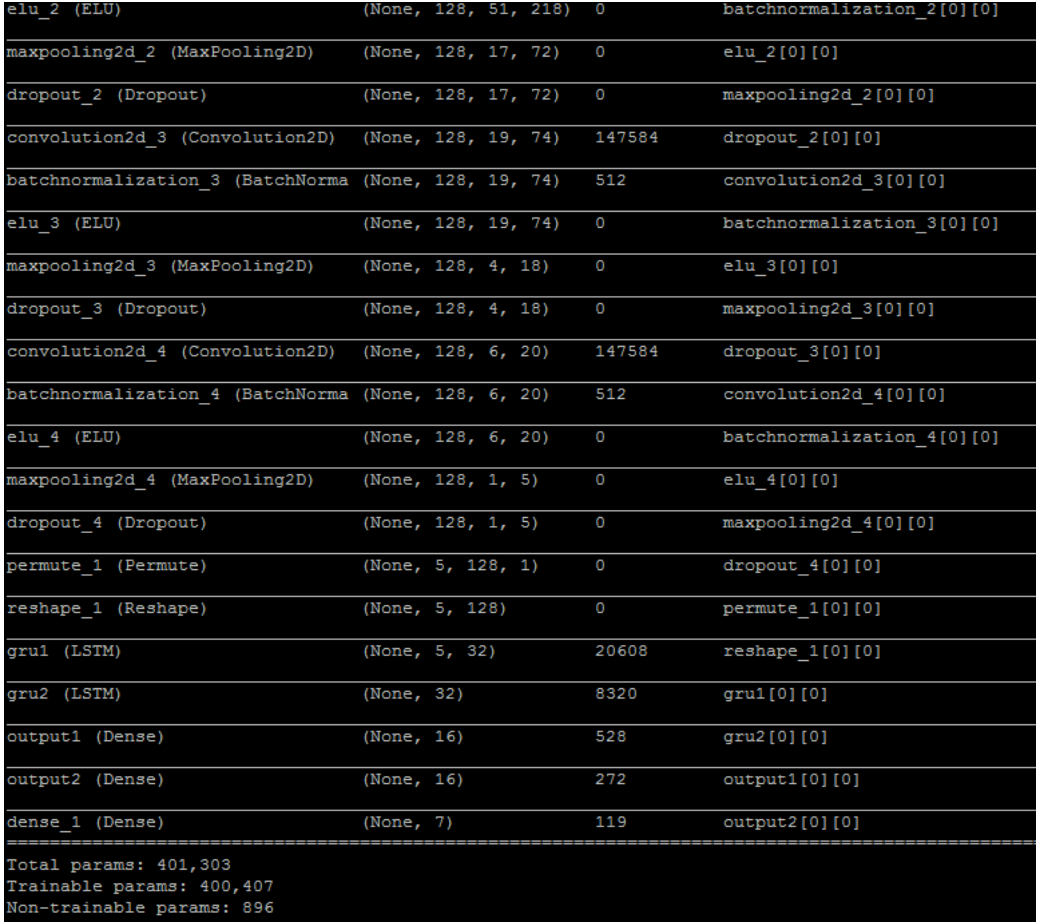
	angry	disgust	fear	happy	neutral	sad	surprise
angry	49	0	0	7	1	0	1
disgust	4	33	4	5	4	3	0
fear	2	0	21	6	0	5	5
happy	10	4	3	33	2	0	2
neutral	0	1	0	0	52	6	0
sad	0	1	3	1	5	34	0
surprise	2	0	4	4	0	0	28

Confusion Matrix

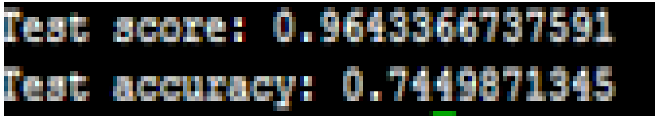
Model 3: CNN + LSTM

Accuracy Obtained: **74.49%**





```
elu_2 (ELU)                       (None, 128, 51, 218)  0           batchnormalization_2[0][0]
maxpooling2d_2 (MaxPooling2D)     (None, 128, 17, 72)   0           elu_2[0][0]
dropout_2 (Dropout)               (None, 128, 17, 72)   0           maxpooling2d_2[0][0]
convolution2d_3 (Convolution2D)   (None, 128, 19, 74)   147584      dropout_2[0][0]
batchnormalization_3 (BatchNorma  (None, 128, 19, 74)   512         convolution2d_3[0][0]
elu_3 (ELU)                       (None, 128, 19, 74)   0           batchnormalization_3[0][0]
maxpooling2d_3 (MaxPooling2D)     (None, 128, 4, 18)    0           elu_3[0][0]
dropout_3 (Dropout)               (None, 128, 4, 18)    0           maxpooling2d_3[0][0]
convolution2d_4 (Convolution2D)   (None, 128, 6, 20)    147584      dropout_3[0][0]
batchnormalization_4 (BatchNorma  (None, 128, 6, 20)    512         convolution2d_4[0][0]
elu_4 (ELU)                       (None, 128, 6, 20)    0           batchnormalization_4[0][0]
maxpooling2d_4 (MaxPooling2D)     (None, 128, 1, 5)     0           elu_4[0][0]
dropout_4 (Dropout)               (None, 128, 1, 5)     0           maxpooling2d_4[0][0]
permute_1 (Permute)               (None, 5, 128, 1)     0           dropout_4[0][0]
reshape_1 (Reshape)               (None, 5, 128)        0           permute_1[0][0]
gru1 (LSTM)                       (None, 5, 32)         20608       reshape_1[0][0]
gru2 (LSTM)                       (None, 32)            8320        gru1[0][0]
output1 (Dense)                   (None, 16)            528         gru2[0][0]
output2 (Dense)                   (None, 16)            272         output1[0][0]
dense_1 (Dense)                   (None, 7)             119         output2[0][0]
================================================================================================
Total params: 401,303
Trainable params: 400,407
Non-trainable params: 896
```



```
Test score: 0.9643366737591
Test accuracy: 0.7449871945
```

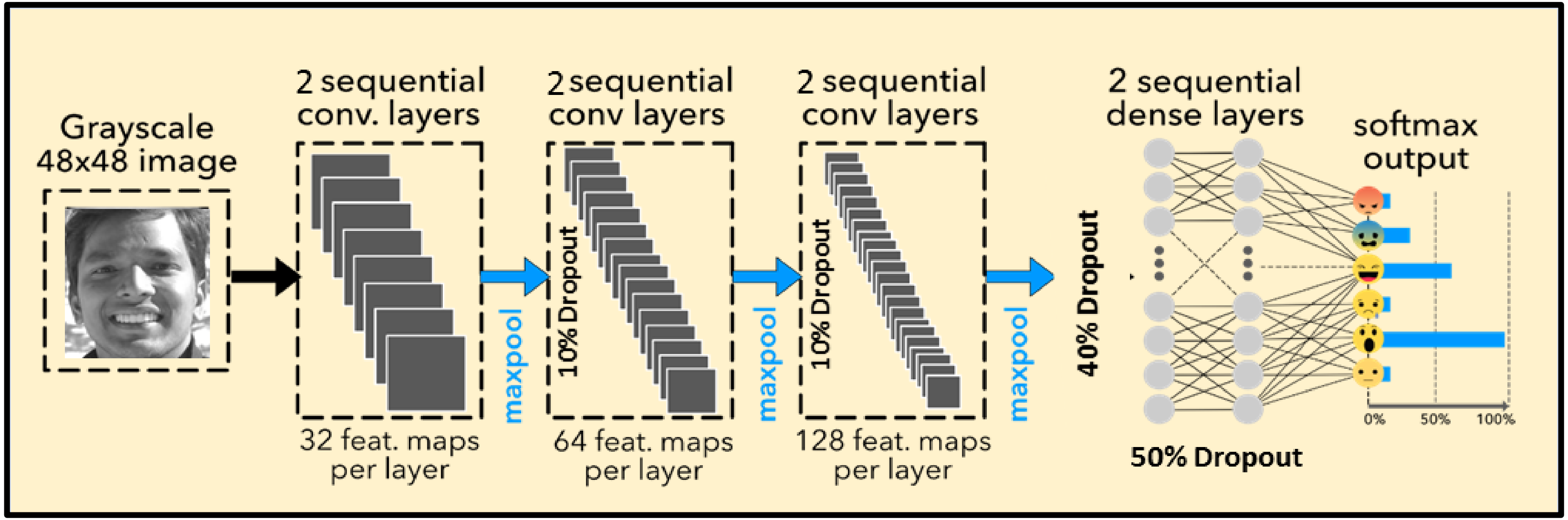
Predicted Label

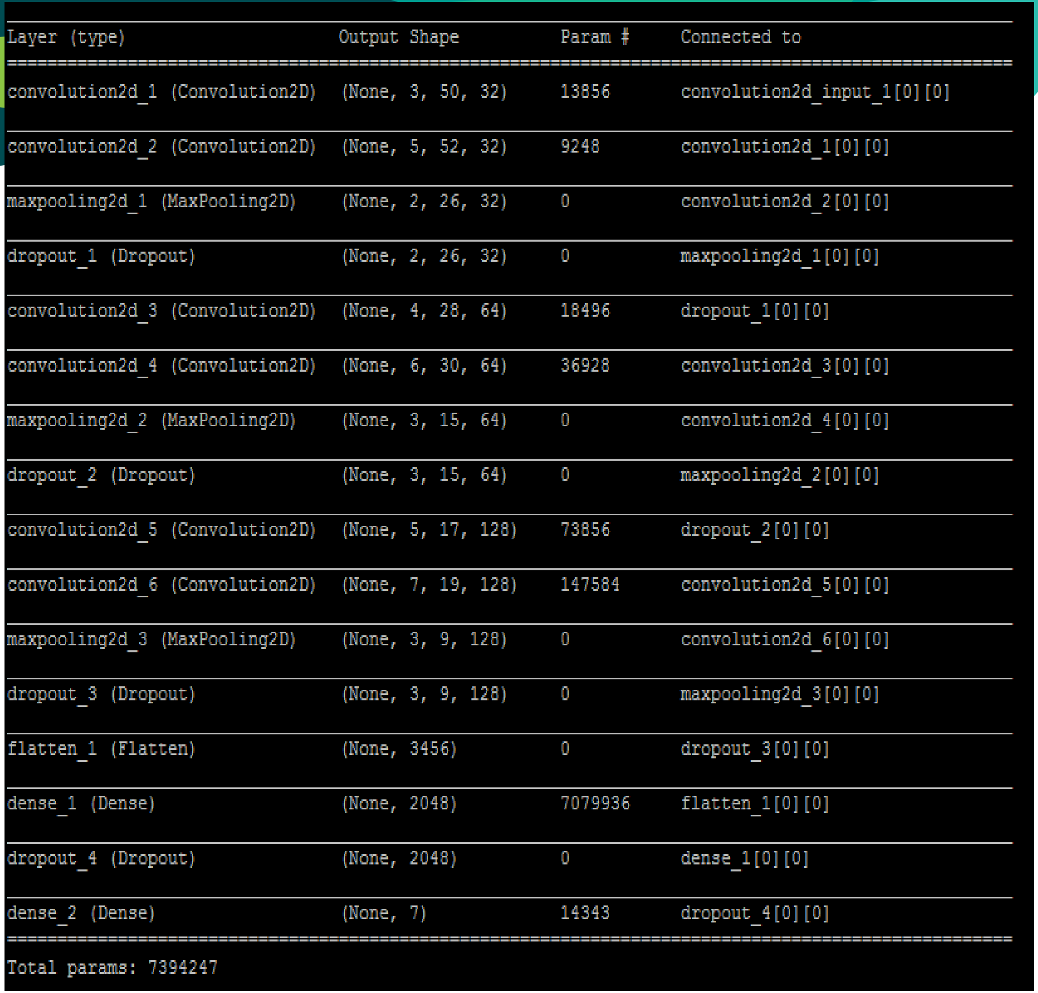
True Label	angry	disgust	fear	happy	neutral	sad	surprise
angry	49	0	0	7	1	0	1
disgust	2	35	4	4	5	3	0
fear	2	0	23	5	0	4	5
happy	8	3	3	36	2	0	2
neutral	0	1	0	0	54	4	0
sad	0	1	4	1	5	33	0
surprise	2	0	5	4	0	0	27

Confusion Matrix

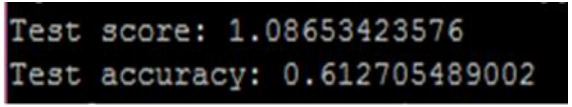
Module 2: Facial Expression

Accuracy Obtained: **61.27%**

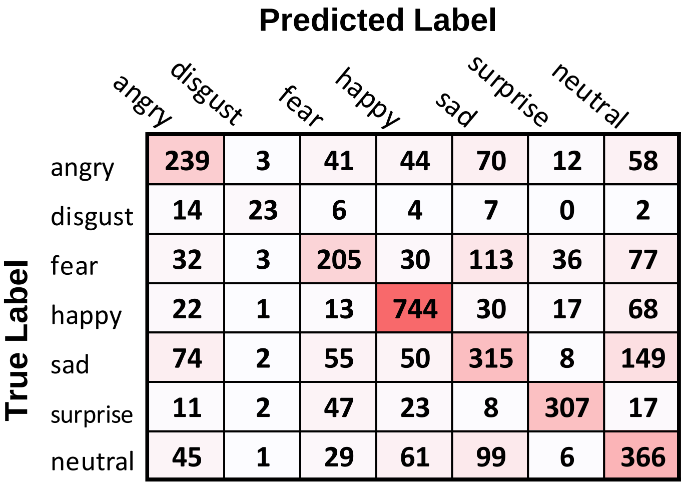




```
Layer (type)                     Output Shape          Param #     Connected to
====================================================================================================
convolution2d_1 (Convolution2D)  (None, 3, 50, 32)     13856       convolution2d_input_1[0][0]

convolution2d_2 (Convolution2D)  (None, 5, 52, 32)     9248        convolution2d_1[0][0]

maxpooling2d_1 (MaxPooling2D)    (None, 2, 26, 32)     0           convolution2d_2[0][0]

dropout_1 (Dropout)              (None, 2, 26, 32)     0           maxpooling2d_1[0][0]

convolution2d_3 (Convolution2D)  (None, 4, 28, 64)     18496       dropout_1[0][0]

convolution2d_4 (Convolution2D)  (None, 6, 30, 64)     36928       convolution2d_3[0][0]

maxpooling2d_2 (MaxPooling2D)    (None, 3, 15, 64)     0           convolution2d_4[0][0]

dropout_2 (Dropout)              (None, 3, 15, 64)     0           maxpooling2d_2[0][0]

convolution2d_5 (Convolution2D)  (None, 5, 17, 128)    73856       dropout_2[0][0]

convolution2d_6 (Convolution2D)  (None, 7, 19, 128)    147584      convolution2d_5[0][0]

maxpooling2d_3 (MaxPooling2D)    (None, 3, 9, 128)     0           convolution2d_6[0][0]

dropout_3 (Dropout)              (None, 3, 9, 128)     0           maxpooling2d_3[0][0]

flatten_1 (Flatten)              (None, 3456)          0           dropout_3[0][0]

dense_1 (Dense)                  (None, 2048)          7079936     flatten_1[0][0]

dropout_4 (Dropout)              (None, 2048)          0           dense_1[0][0]

dense_2 (Dense)                  (None, 7)             14343       dropout_4[0][0]
====================================================================================================
Total params: 7394247
```



```
Test score: 1.08653423576
Test accuracy: 0.612705489002
```



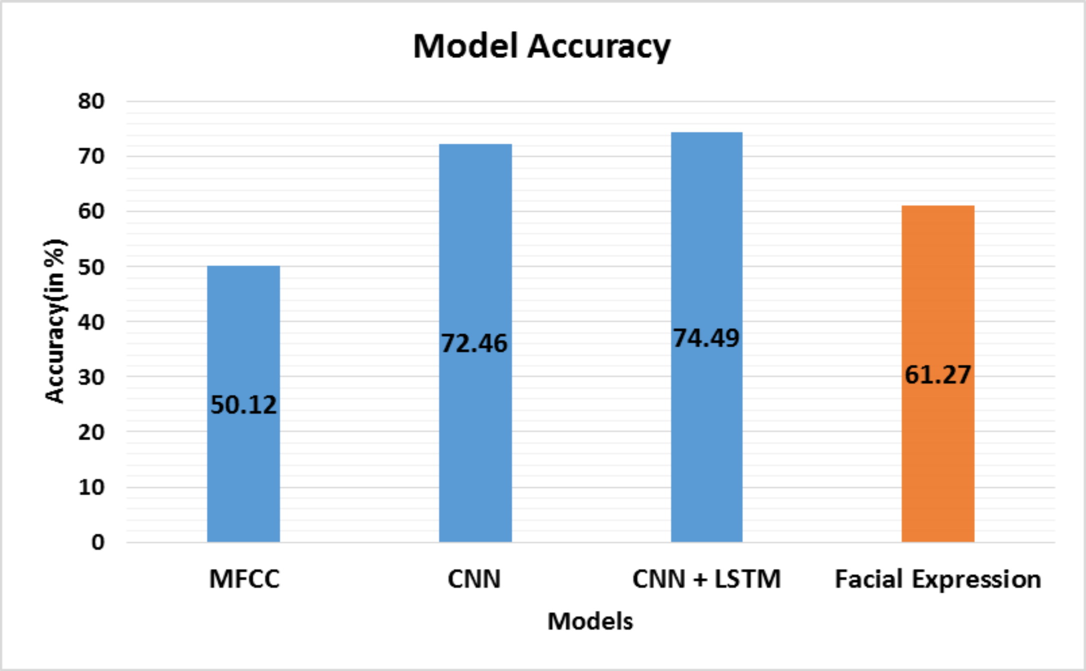
Predicted Label

True Label

	angry	disgust	fear	happy	sad	surprise	neutral
angry	239	3	41	44	70	12	58
disgust	14	23	6	4	7	0	2
fear	32	3	205	30	113	36	77
happy	22	1	13	744	30	17	68
sad	74	2	55	50	315	8	149
surprise	11	2	47	23	8	307	17
neutral	45	1	29	61	99	6	366

Confusion Matrix

Choosing the Best Model



Fusion Based on F-Scores

$$\text{Precision} = \frac{\text{TP}}{\text{TP + FP}}$$

TP=True Positive
FP=False Positive

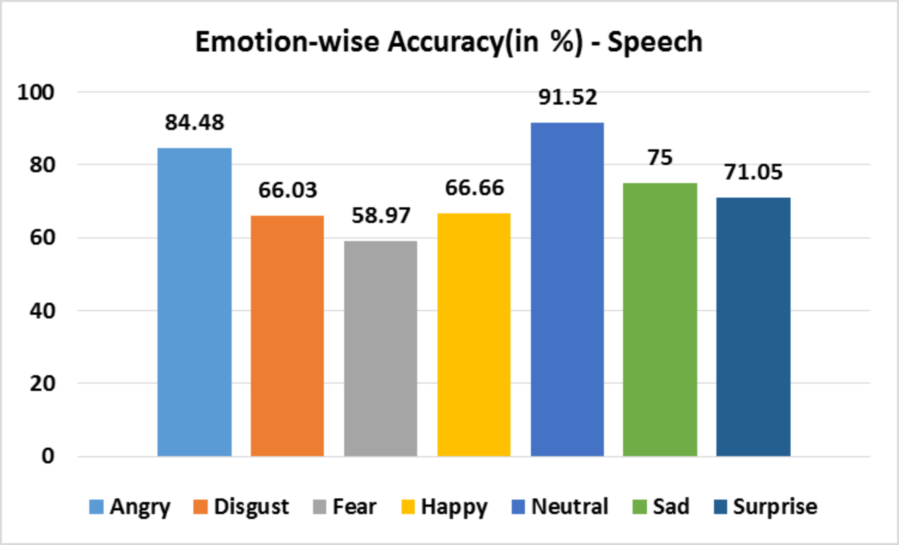
$$\text{Recall} = \frac{\text{TP}}{\text{TP + FN}}$$

TP=True Positive
FN=False Negative

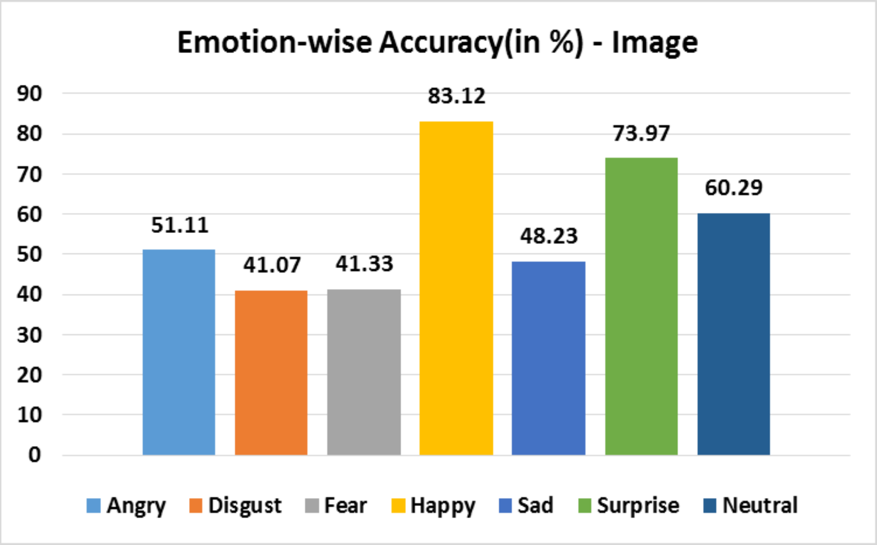
$$\text{F-Score} = \frac{2 * (\text{recall} * \text{precision})}{\text{recall} + \text{precision}}$$

Emotions	Images	Speech
Angry	0.5308	0.9034
Disgust	0.5748	0.4864
Fear	0.4581	0.5194
Happy	0.8113	0.6024
Neutral	0.5302	0.7880
Sad	0.4765	0.75
Surprise	0.7623	0.6236

Analysis

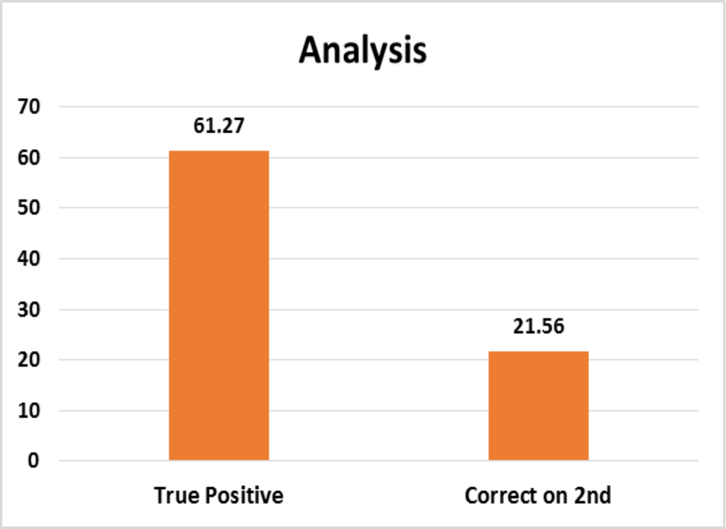


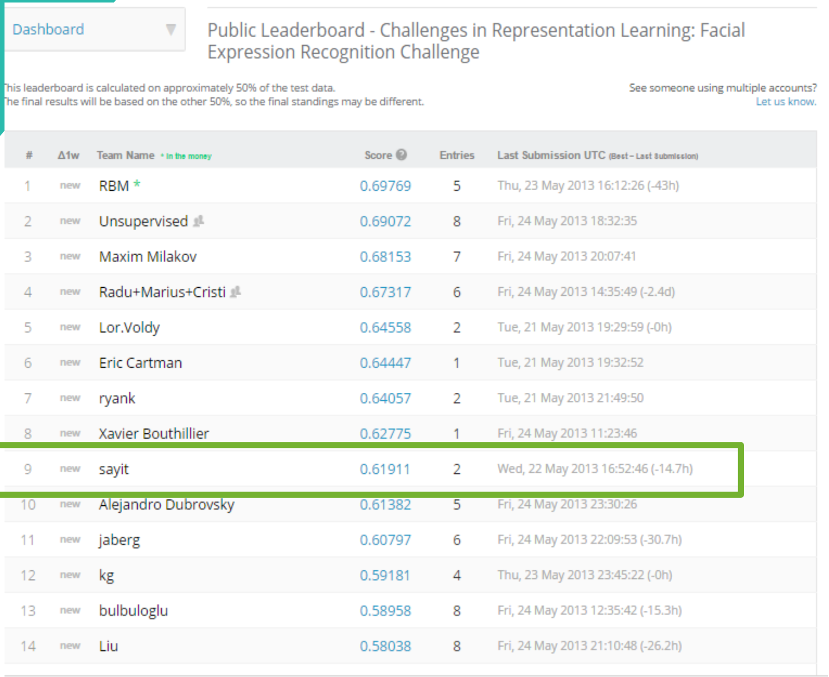
Emotion-wise Accuracy(in %) - Speech



Emotion-wise Accuracy(in %) - Image

Dataset	*Accuracy*
SAVEE	60.19%
IIIT A	80.93%
SAVEE + IIIT A	74.49%





Public Leaderboard - Challenges in Representation Learning: Facial Expression Recognition Challenge

his leaderboard is calculated on approximately 50% of the test data.
he final results will be based on the other 50%, so the final standings may be different.

See someone using multiple accounts?
Let us know.

#	Δ1w	Team Name * In the money	Score	Entries	Last Submission UTC (Best – Last Submission)
1	new	RBM *	0.69769	5	Thu, 23 May 2013 16:12:26 (-43h)
2	new	Unsupervised	0.69072	8	Fri, 24 May 2013 18:32:35
3	new	Maxim Milakov	0.68153	7	Fri, 24 May 2013 20:07:41
4	new	Radu+Marius+Cristi	0.67317	6	Fri, 24 May 2013 14:35:49 (-2.4d)
5	new	Lor.Voldy	0.64558	2	Tue, 21 May 2013 19:29:59 (-0h)
6	new	Eric Cartman	0.64447	1	Tue, 21 May 2013 19:32:52
7	new	ryank	0.64057	2	Tue, 21 May 2013 21:49:50
8	new	Xavier Bouthillier	0.62775	1	Fri, 24 May 2013 11:23:46
9	new	sayit	0.61911	2	Wed, 22 May 2013 16:52:46 (-14.7h)
10	new	Alejandro Dubrovsky	0.61382	5	Fri, 24 May 2013 23:30:26
11	new	jaberg	0.60797	6	Fri, 24 May 2013 22:09:53 (-30.7h)
12	new	kg	0.59181	4	Thu, 23 May 2013 23:45:22 (-0h)
13	new	bulbuloglu	0.58958	8	Fri, 24 May 2013 12:35:42 (-15.3h)
14	new	Liu	0.58038	8	Fri, 24 May 2013 21:10:48 (-26.2h)

Kaggle Position(9th) with 61.27% accuracy for facial data

On testing the validation data, we analysed that **21.56% misclassified images have second most likely emotion as the correct label.**

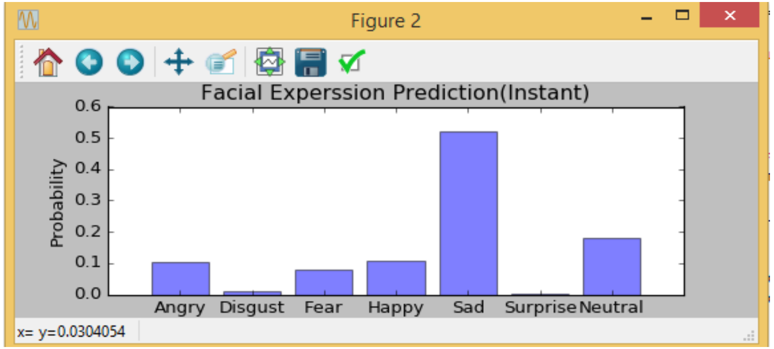
Modules Developed

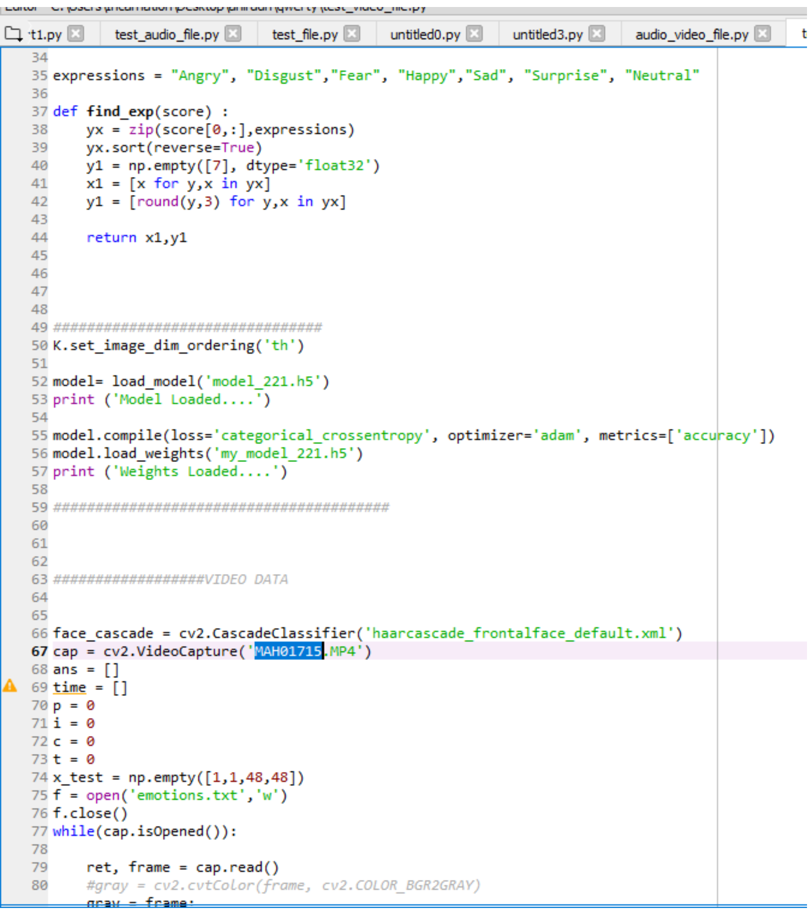
i. Emotion Classification using speech signal from audio file

ii. Emotion Classification using from microphone audio file

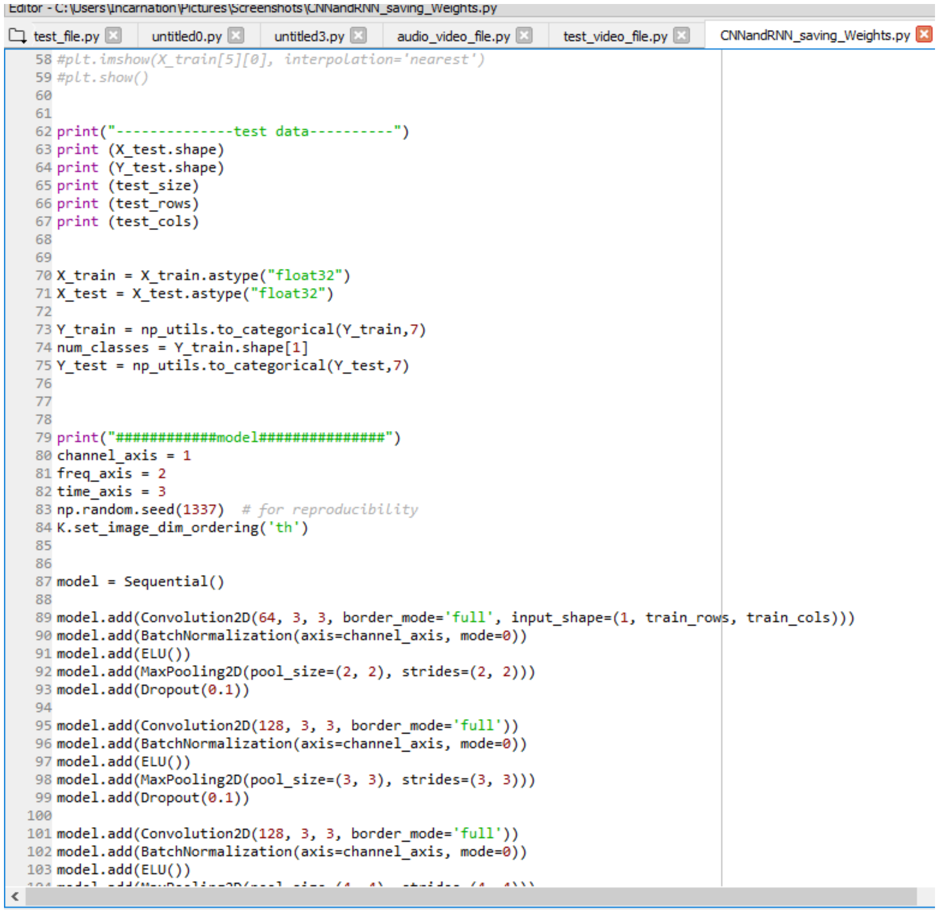
iii. Real Time Webcam Video Classification using Facial Expression

iv. Emotion Classification using fusion of speech and audio signals.

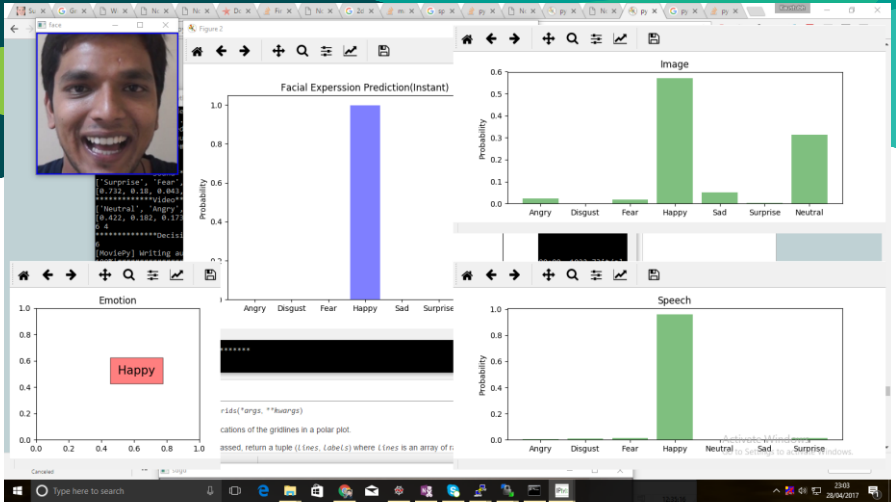




```python

expressions = "Angry", "Disgust","Fear", "Happy","Sad", "Surprise", "Neutral"

def find_exp(score) :
    yx = zip(score[0,:],expressions)
    yx.sort(reverse=True)
    y1 = np.empty([7], dtype='float32')
    x1 = [x for y,x in yx]
    y1 = [round(y,3) for y,x in yx]

    return x1,y1




#################################
K.set_image_dim_ordering('th')

model= load_model('model_221.h5')
print ('Model Loaded....')

model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
model.load_weights('my_model_221.h5')
print ('Weights Loaded....')

########################################



##################VIDEO DATA


face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
cap = cv2.VideoCapture('MAH01715.MP4')
ans = []
time = []
p = 0
i = 0
c = 0
t = 0
x_test = np.empty([1,1,48,48])
f = open('emotions.txt','w')
f.close()
while(cap.isOpened()):

    ret, frame = cap.read()
    #gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    gray = frame;
```

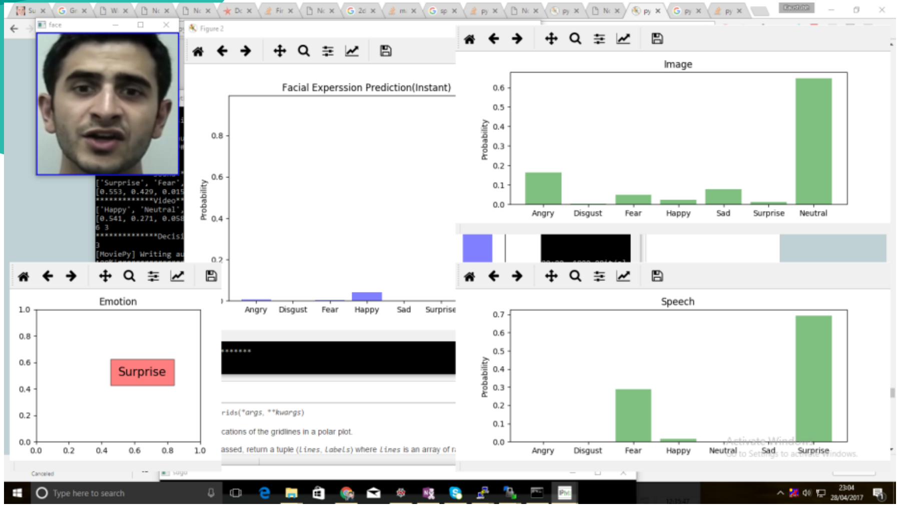

Editor - C:\Users\Incarnation\Pictures\Screenshots\CNNandRNN_saving_Weights.py
test_file.py
untitled0.py
untitled3.py
audio_video_file.py
test_video_file.py
CNNandRNN_saving_Weights.py

```python
#plt.imshow(X_train[5][0], interpolation='nearest')
#plt.show()


print("--------------test data----------")
print (X_test.shape)
print (Y_test.shape)
print (test_size)
print (test_rows)
print (test_cols)


X_train = X_train.astype("float32")
X_test = X_test.astype("float32")

Y_train = np_utils.to_categorical(Y_train,7)
num_classes = Y_train.shape[1]
Y_test = np_utils.to_categorical(Y_test,7)



print("###########model###############")
channel_axis = 1
freq_axis = 2
time_axis = 3
np.random.seed(1337)  # for reproducibility
K.set_image_dim_ordering('th')


model = Sequential()

model.add(Convolution2D(64, 3, 3, border_mode='full', input_shape=(1, train_rows, train_cols)))
model.add(BatchNormalization(axis=channel_axis, mode=0))
model.add(ELU())
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
model.add(Dropout(0.1))

model.add(Convolution2D(128, 3, 3, border_mode='full'))
model.add(BatchNormalization(axis=channel_axis, mode=0))
model.add(ELU())
model.add(MaxPooling2D(pool_size=(3, 3), strides=(3, 3)))
model.add(Dropout(0.1))

model.add(Convolution2D(128, 3, 3, border_mode='full'))
model.add(BatchNormalization(axis=channel_axis, mode=0))
model.add(ELU())
model.add(MaxPooling2D(pool_size=(4, 4), strides=(4, 4)))
```



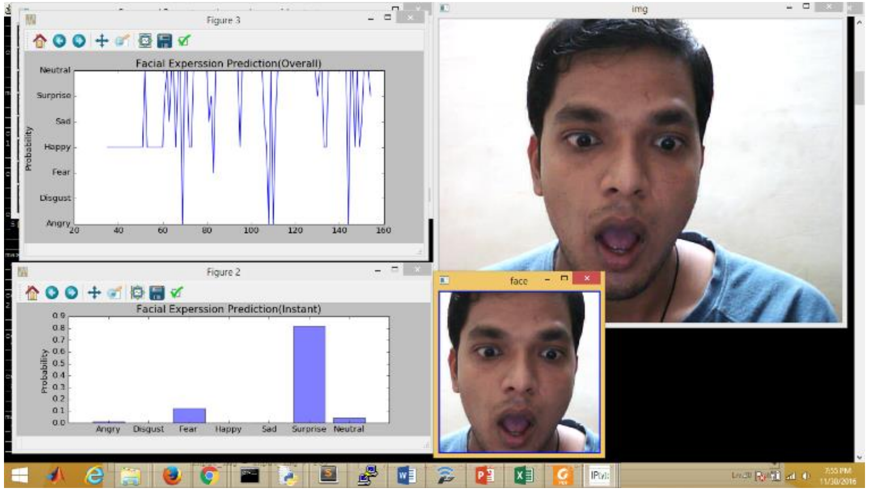
```
['Surprise', 'Fear',
[0.732, 0.18, 0.043,
**************Video**
['Neutral', 'Angry',
[0.422, 0.182, 0.173
6 4
***************Decisi
6
[MoviePy] Writing au
```

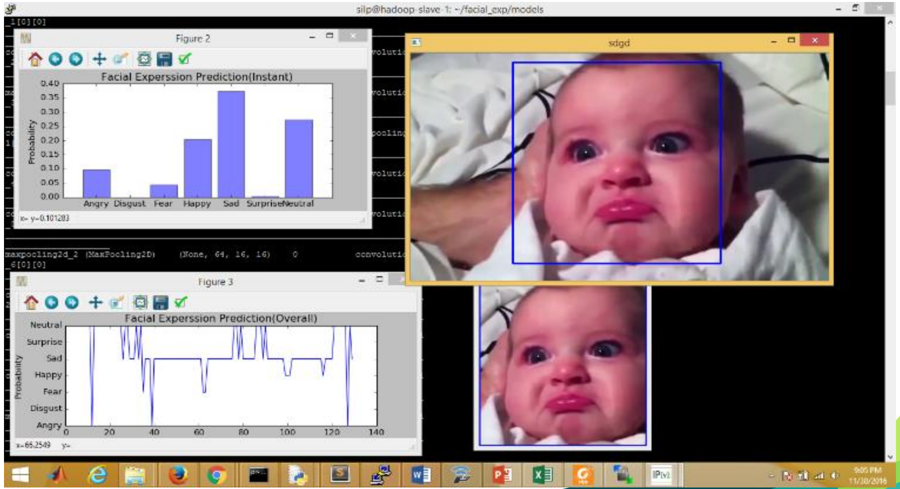
```
*******
rids(*args, **kwargs)
cations of the gridlines in a polar plot.
assed, return a tuple (lines, labels) where lines is an array of r
```



```
['Surprise', 'Fear',
[0.553, 0.429, 0.015
**************Video**
['Happy', 'Neutral',
[0.541, 0.271, 0.058
6 3
***************Decisi
3
[MoviePy] Writing au
```

```
*******
rids(*args, **kwargs)
cations of the gridlines in a polar plot.
assed, return a tuple (lines, labels) where lines is an array of r
```





```
maxpooling2d_2 (MaxPooling2D)     (None, 64, 16, 16)     0          convoluti
```

Technical Requirements

- Nvidia GPU
- CUDA
- Python
- Anaconda
- Librosa
- pyAudioAnalysis
- Theano
- Kares
- PyCharm

Thank You!

Vaibhav Srivastava (IIT2013027)
Himanshu Tuteja (IIT2013038)
Anirudh Gupta (IIT2013117)