Affective Computing | Assignment 3

Divyanshu Kumar Singh | 2017048

Part A.) Implementing Baseline

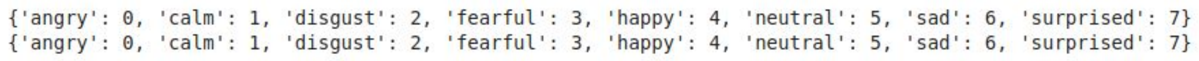
For the baseline implementation, we were given the original paper [1], so instead of considering their baseline to its high computation nature. I have used [2] as my baseline model, also since we were allowed to go for a unimodal approach for the baseline I have only used Audio-based features as mentioned in the [2].

The **feature** considered were :

MFCC's (13)	**ZCR**
Chroma-Based Feature	**energy**
spectral_flux	**entropy_energy**
spectral_entropy	

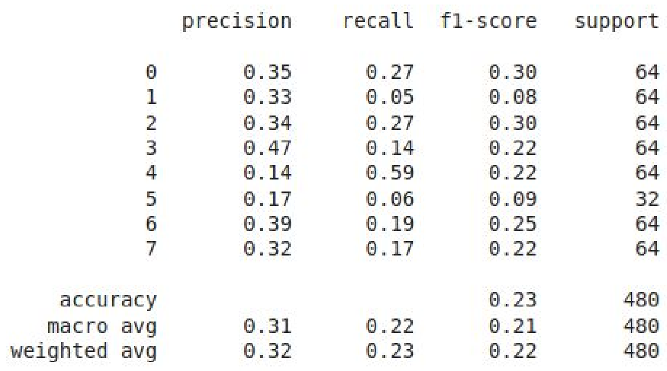
Based on my feature extraction and given the fact that data was completely different from the paper provided. Here are my baseline results -

These are **encoded labels** from the encoder-



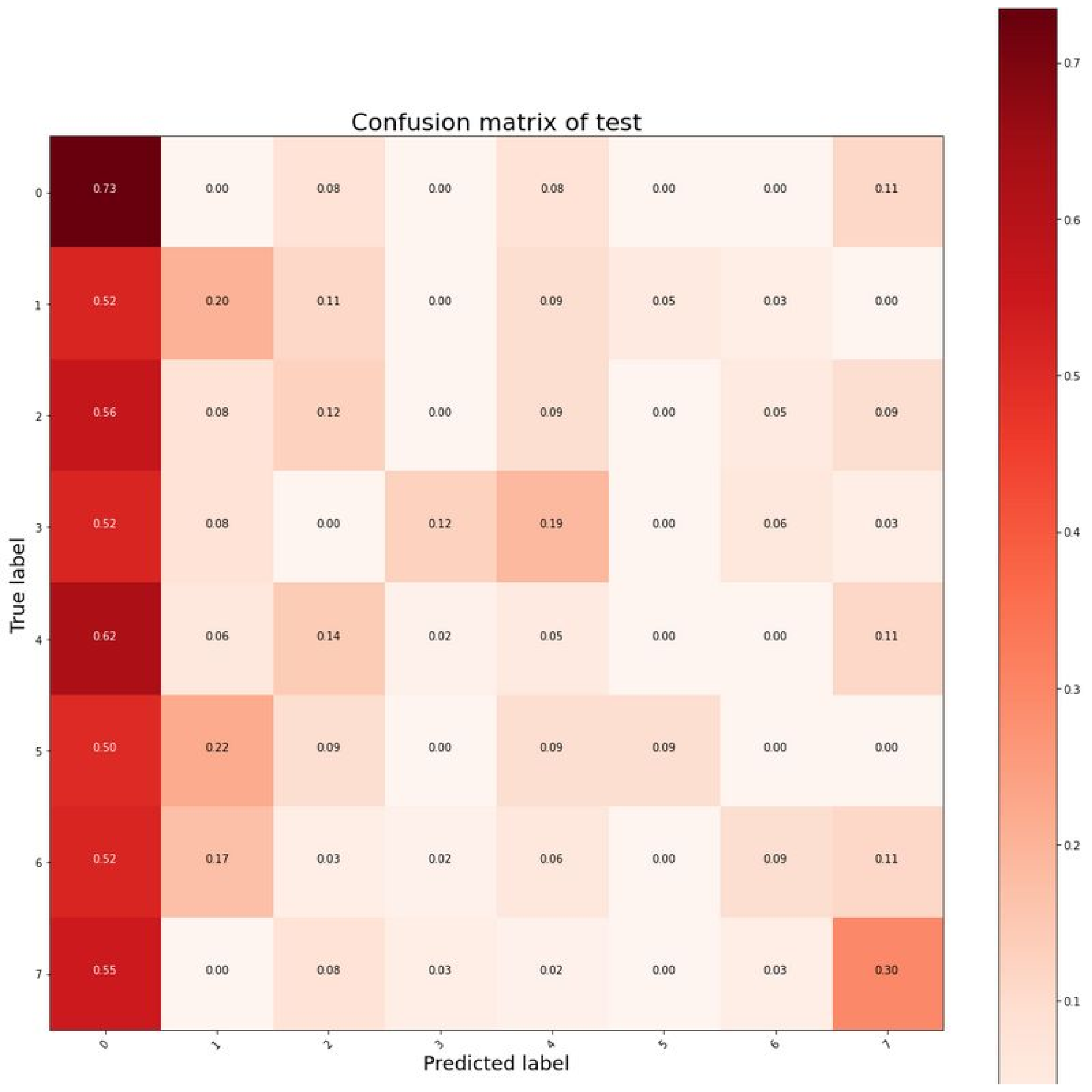
```
{'angry': 0, 'calm': 1, 'disgust': 2, 'fearful': 3, 'happy': 4, 'neutral': 5, 'sad': 6, 'surprised': 7}
{'angry': 0, 'calm': 1, 'disgust': 2, 'fearful': 3, 'happy': 4, 'neutral': 5, 'sad': 6, 'surprised': 7}
```

Using SVM, I achieved **22.89%** accuracy in the classification and below is the classification report**.**



```
              precision    recall  f1-score   support

           0       0.35      0.27      0.30        64
           1       0.33      0.05      0.08        64
           2       0.34      0.27      0.30        64
           3       0.47      0.14      0.22        64
           4       0.14      0.59      0.22        64
           5       0.17      0.06      0.09        32
           6       0.39      0.19      0.25        64
           7       0.32      0.17      0.22        64

    accuracy                           0.23       480
   macro avg       0.31      0.22      0.21       480
weighted avg       0.32      0.23      0.22       480
```

The **Confusion matrix** for the same is shown below:

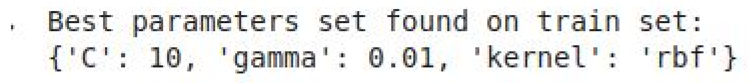


Part B.) **Extending with Some New Approach**

To extend to the new approach, I have used two modalities.
i.) Audio
ii.) Video

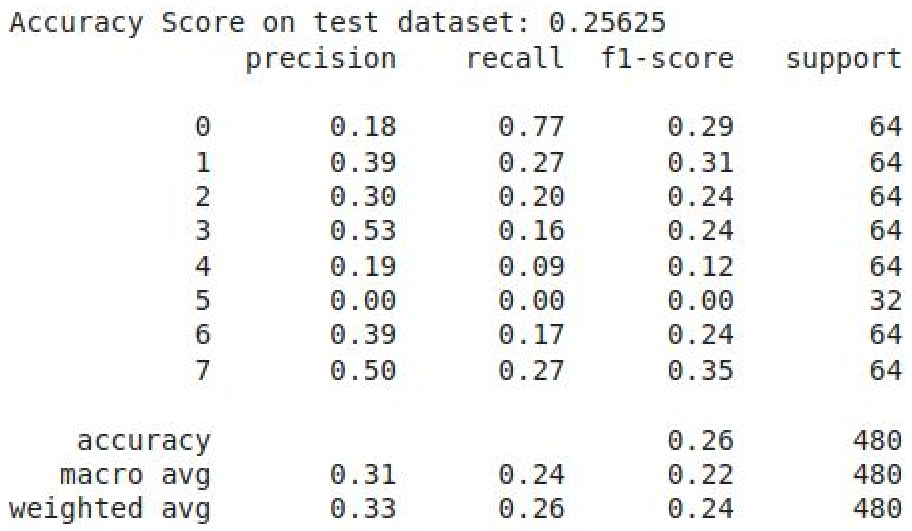
Audio
I have considered the same [2] approach, but the authors have not considered finding the best features out of all the features. Hence, I have used this fact and tried finding the best feature and then downsample the feature vectors dimension using PCA. I have used *SelectKBest* to find strong features from 'all'. Then reduced the dimensionality of the vector.

Using the results were as follows:
Best Parameter



```
. Best parameters set found on train set:
  {'C': 10, 'gamma': 0.01, 'kernel': 'rbf'}
```

Classification Report and Accuracy



```
Accuracy Score on test dataset: 0.25625
              precision    recall  f1-score   support

           0       0.18      0.77      0.29        64
           1       0.39      0.27      0.31        64
           2       0.30      0.20      0.24        64
           3       0.53      0.16      0.24        64
           4       0.19      0.09      0.12        64
           5       0.00      0.00      0.00        32
           6       0.39      0.17      0.24        64
           7       0.50      0.27      0.35        64

    accuracy                           0.26       480
   macro avg       0.31      0.24      0.22       480
weighted avg       0.33      0.26      0.24       480
```

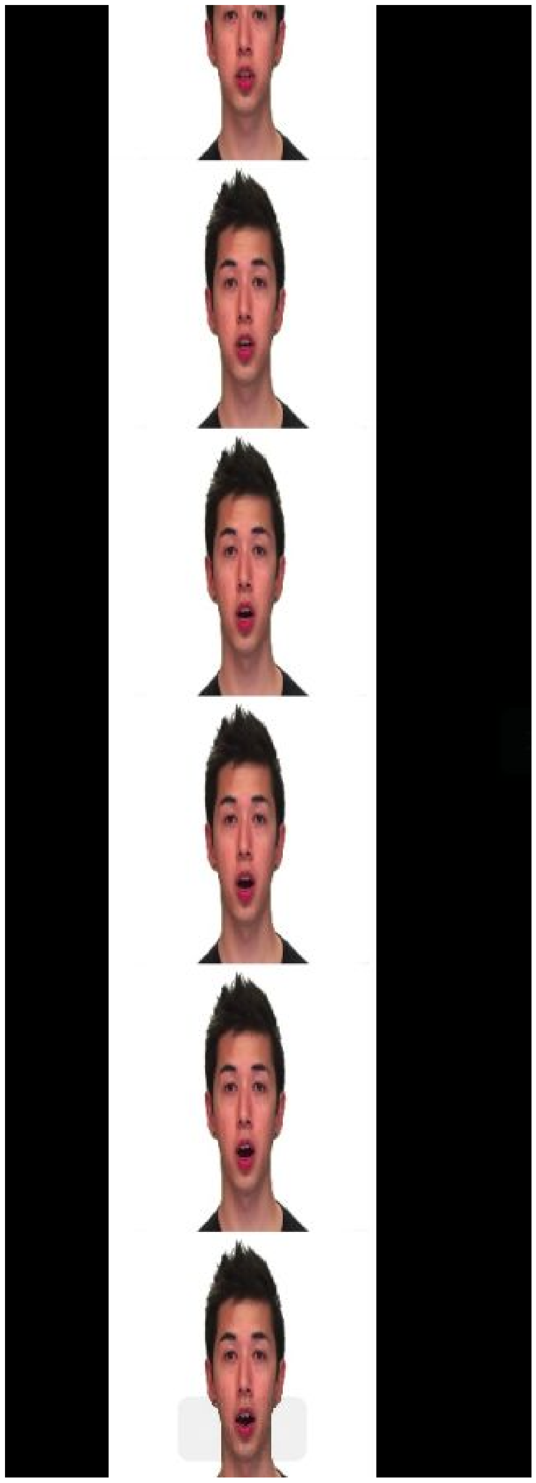
Using this I have managed to increase the accuracy by **3%** from the original approach.

Video

For analysis from the video, we first extract the frames from the video. Then, for each video we got nearly 90-150 frames. Since we had 24 actors and each actor had 120 videos, therefore, we stacked(vertically) images from a single video into a single jpg file.

For training we had, 2400 image which essentially had all the frames stacked vertically onto it and for testing, we had 480 images. Also, we put an upper cap of 120 frames in a single jpg to reduce abnormality in stack sizes.

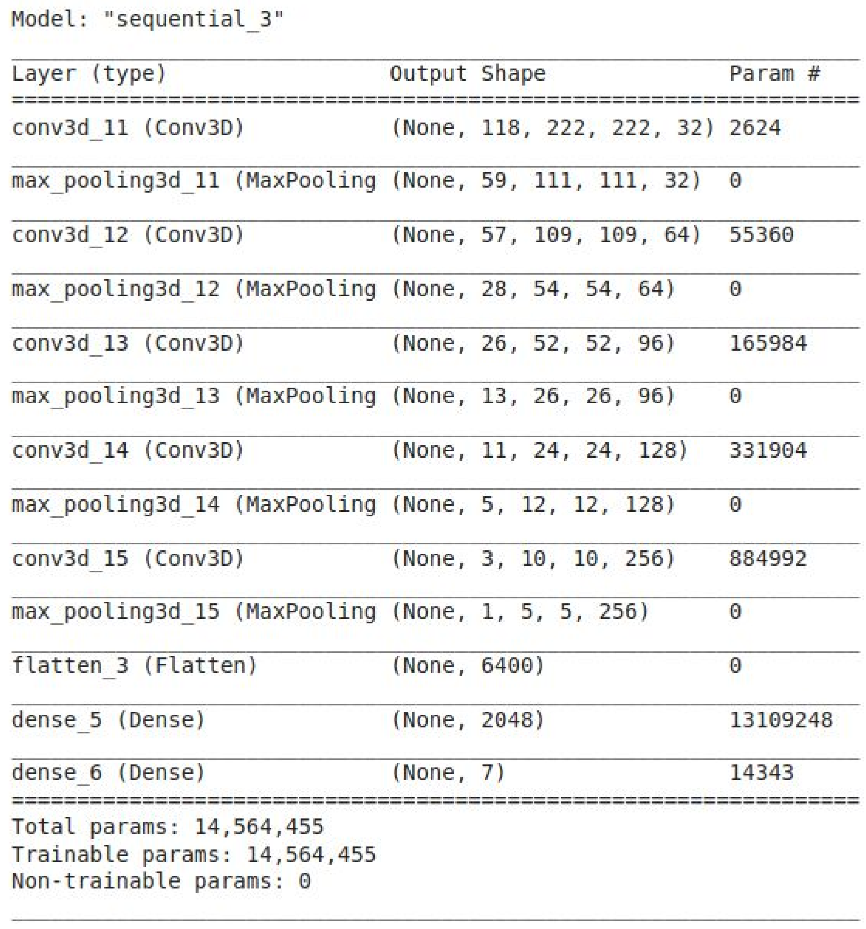
An example of frame is down below :



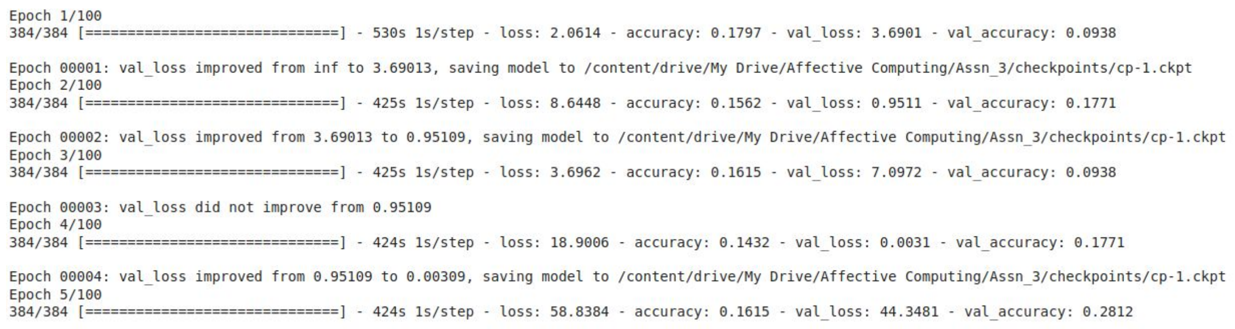
Model Description

For the learning part, here is the model that we are using:

We decided to implement a CNN architecture but with 3D convolutions which will help the model to not only understand the spatial feature (2D convolutions are well known for capturing these features) but also temporal features that will help the model to understand the correlation between each frame of a particular video.
The model takes an input of (1,120,224,224,3) where (224,224,3) is a particular frame from a video and we have 120 such frames stacked on top of each other and 1 denotes the batch size. After each 3D convolution we have use relu as an activation followed by a max pooling layer with stride 2 in order to reduce the width and increase channel as go on deep and finally a dense layer to pick out best relevant feature and classify the emotion .
We have used Adam as an optimizer and the last layer has softmax as an activation and loss function we have used is categorical cross entropy. And we are saving the checkpoints according to the validation loss.



```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv3d_11 (Conv3D)           (None, 118, 222, 222, 32) 2624
_________________________________________________________________
max_pooling3d_11 (MaxPooling (None, 59, 111, 111, 32)  0
_________________________________________________________________
conv3d_12 (Conv3D)           (None, 57, 109, 109, 64)  55360
_________________________________________________________________
max_pooling3d_12 (MaxPooling (None, 28, 54, 54, 64)    0
_________________________________________________________________
conv3d_13 (Conv3D)           (None, 26, 52, 52, 96)    165984
_________________________________________________________________
max_pooling3d_13 (MaxPooling (None, 13, 26, 26, 96)    0
_________________________________________________________________
conv3d_14 (Conv3D)           (None, 11, 24, 24, 128)   331904
_________________________________________________________________
max_pooling3d_14 (MaxPooling (None, 5, 12, 12, 128)    0
_________________________________________________________________
conv3d_15 (Conv3D)           (None, 3, 10, 10, 256)    884992
_________________________________________________________________
max_pooling3d_15 (MaxPooling (None, 1, 5, 5, 256)      0
_________________________________________________________________
flatten_3 (Flatten)          (None, 6400)              0
_________________________________________________________________
dense_5 (Dense)              (None, 2048)              13109248
_________________________________________________________________
dense_6 (Dense)              (None, 7)                 14343
=================================================================
Total params: 14,564,455
Trainable params: 14,564,455
Non-trainable params: 0
_________________________________________________________________
```

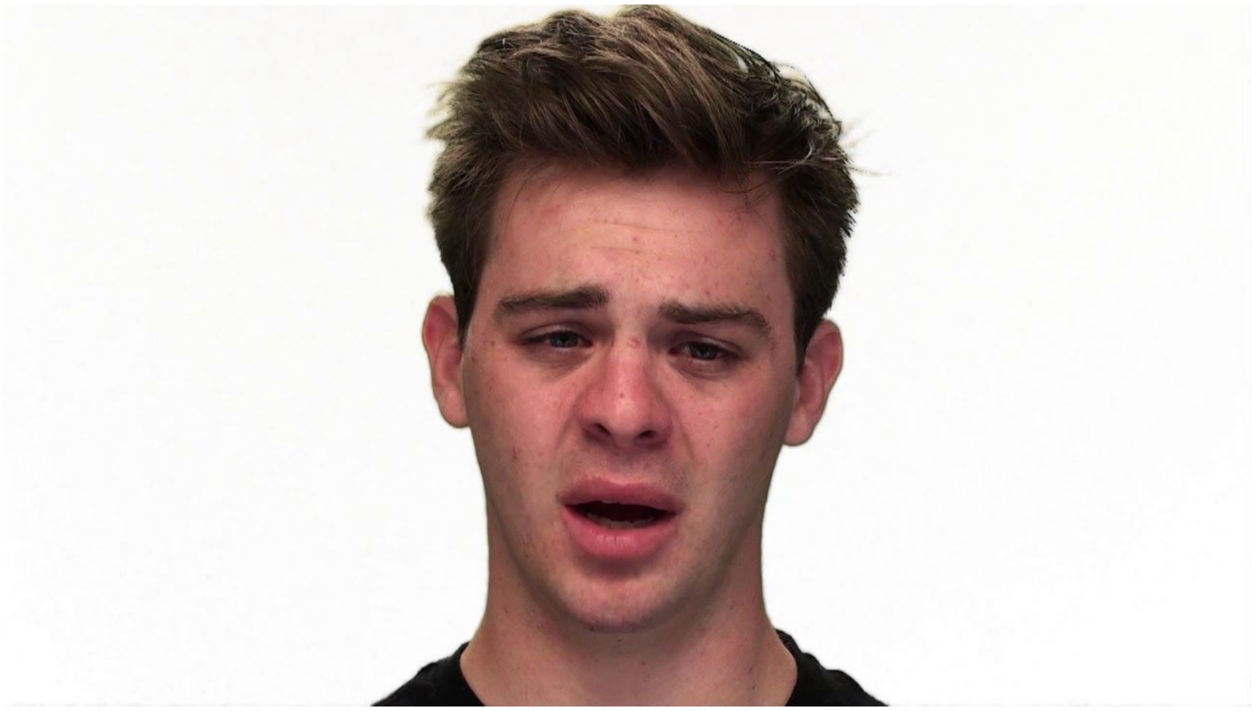


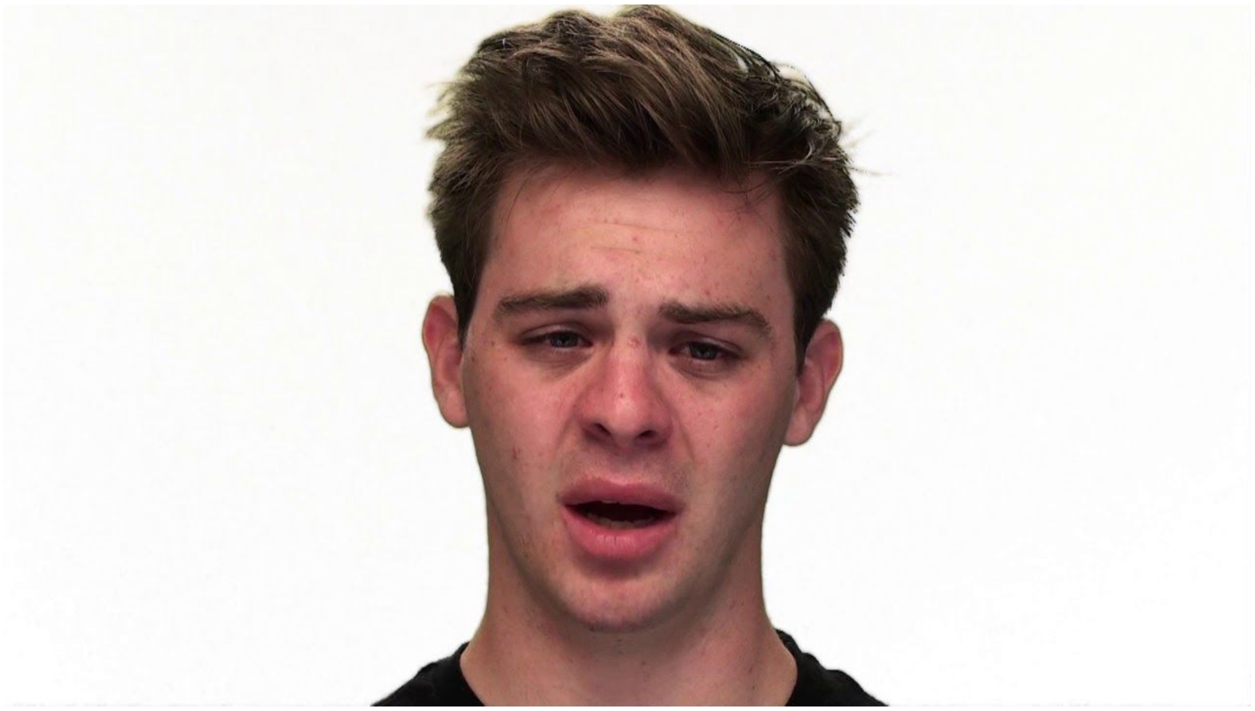
```
Epoch 1/100
384/384 [==============================] - 530s 1s/step - loss: 2.0614 - accuracy: 0.1797 - val_loss: 3.6901 - val_accuracy: 0.0938

Epoch 00001: val_loss improved from inf to 3.69013, saving model to /content/drive/My Drive/Affective Computing/Assn_3/checkpoints/cp-1.ckpt
Epoch 2/100
384/384 [==============================] - 425s 1s/step - loss: 8.6448 - accuracy: 0.1562 - val_loss: 0.9511 - val_accuracy: 0.1771

Epoch 00002: val_loss improved from 3.69013 to 0.95109, saving model to /content/drive/My Drive/Affective Computing/Assn_3/checkpoints/cp-1.ckpt
Epoch 3/100
384/384 [==============================] - 425s 1s/step - loss: 3.6962 - accuracy: 0.1615 - val_loss: 7.0972 - val_accuracy: 0.0938

Epoch 00003: val_loss did not improve from 0.95109
Epoch 4/100
384/384 [==============================] - 424s 1s/step - loss: 18.9006 - accuracy: 0.1432 - val_loss: 0.0031 - val_accuracy: 0.1771

Epoch 00004: val_loss improved from 0.95109 to 0.00309, saving model to /content/drive/My Drive/Affective Computing/Assn_3/checkpoints/cp-1.ckpt
Epoch 5/100
384/384 [==============================] - 424s 1s/step - loss: 58.8384 - accuracy: 0.1615 - val_loss: 44.3481 - val_accuracy: 0.2812
```

We achieved max accuracy of **28%**, using our approach. The only drawback to this approach is the scarcity of the data, due to very poor internet connection and limited download (I was not able to process the whole data, hence I used a small subset of it.)

Transfer Learning Approach

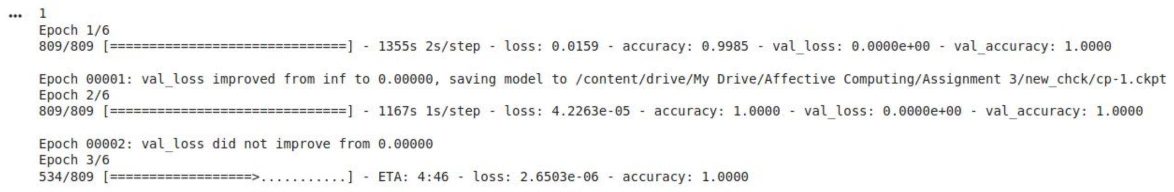
In this approach, inspired from the above, we decided to get time frames from particular intervals. Then we used those intervals to train two models say every 1 sec interval. As shown below:





One for ML methods, Dtree and Random Forest. We basically used an image array (flattened) as a feature. This method gave a very poor accuracy of 7% and 8% respectively.

For transfer learning, we used the Resnet pretrained weight to make the deep learning, and used similar frames data for this task. Surprisingly, Resnet could actually see every detailed point in the images and achieved 100% accuracy. This basically due to the reason that, data is small and really clean, the data is acted and white background, so the inner layers of the resnet actually find every minute detail from it.



```
... 1
    Epoch 1/6
    809/809 [==============================] - 1355s 2s/step - loss: 0.0159 - accuracy: 0.9985 - val_loss: 0.0000e+00 - val_accuracy: 1.0000

    Epoch 00001: val_loss improved from inf to 0.00000, saving model to /content/drive/My Drive/Affective Computing/Assignment 3/new_chck/cp-1.ckpt
    Epoch 2/6
    809/809 [==============================] - 1167s 1s/step - loss: 4.2263e-05 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000

    Epoch 00002: val_loss did not improve from 0.00000
    Epoch 3/6
    534/809 [==================>...........] - ETA: 4:46 - loss: 2.6503e-06 - accuracy: 1.0000
```

You can find the cpkt files here: https://drive.google.com/open?id=12QKYamKxT_yQGNEKGWNqkzOJUI_rYUlQ

References

[1] Dhall, A., Ramana Murthy, O. V., Goecke, R., Joshi, J., & Gedeon, T. (2015, November). Video and image based emotion recognition challenges in the wild: Emotiw 2015. In *Proceedings of the 2015 ACM on international conference on multimodal interaction* (pp. 423-426).

[2] Noroozi, F., Marjanovic, M., Njegus, A., Escalera, S., & Anbarjafari, G. (2017). Audio-visual emotion recognition in video clips. *IEEE Transactions on Affective Computing*, *10*(1), 60-75.