Melih Baştürk

18.02.2022

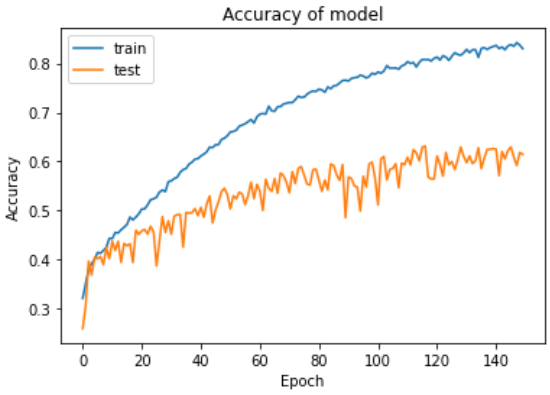
Internship Report

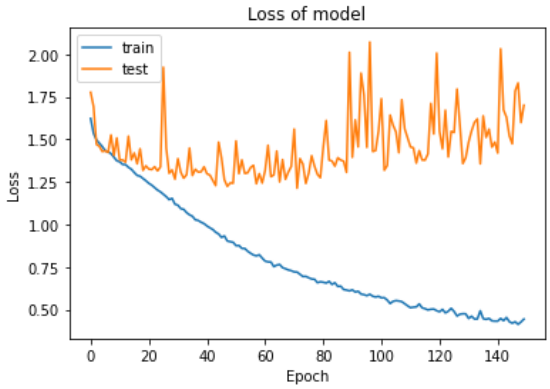
1D-Mel Spectrogram

In my first approach, because audio files have different length I took the mean in time domain for Mel Spectrograms. Thus all audio files have the same shape, which is 128x1. Summary of 1D ConvNet Model is given below figure.

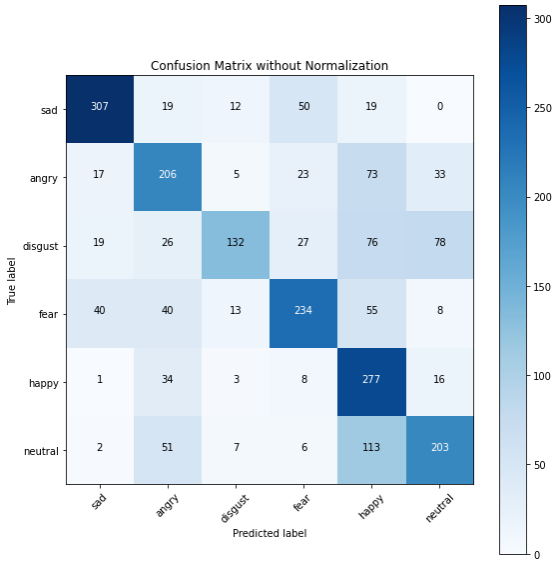
```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d (Conv1D)              (None, 128, 256)          1536

max_pooling1d (MaxPooling1D  (None, 64, 256)           0
)

batch_normalization (BatchN  (None, 64, 256)           1024
ormalization)

conv1d_1 (Conv1D)            (None, 64, 256)           327936

max_pooling1d_1 (MaxPooling  (None, 32, 256)           0
1D)

batch_normalization_1 (Batc  (None, 32, 256)           1024
hNormalization)

conv1d_2 (Conv1D)            (None, 32, 128)           163968

max_pooling1d_2 (MaxPooling  (None, 16, 128)           0
1D)

batch_normalization_2 (Batc  (None, 16, 128)           512
hNormalization)

conv1d_3 (Conv1D)            (None, 16, 64)            41024

max_pooling1d_3 (MaxPooling  (None, 8, 64)             0
1D)

batch_normalization_3 (Batc  (None, 8, 64)             256
hNormalization)

flatten (Flatten)            (None, 512)               0

dense (Dense)                (None, 32)                16416

dropout (Dropout)            (None, 32)                0

dense_1 (Dense)              (None, 6)                 198

=================================================================
Total params: 553,894
Trainable params: 552,486
Non-trainable params: 1,408
```

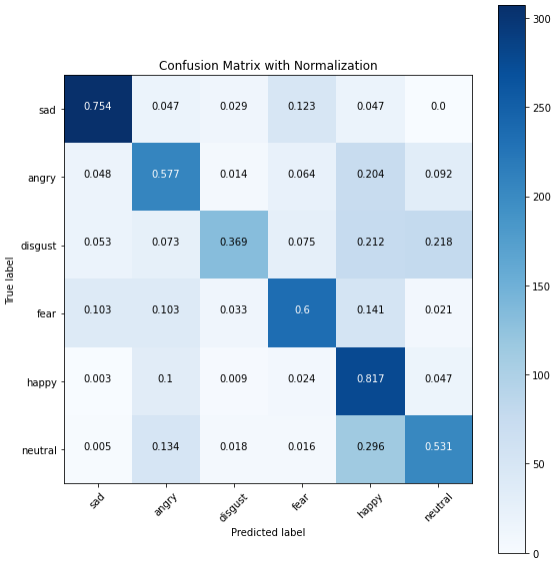
Also because there are not enough data for a good training process, I increased number of audio file by adding noise. This increasing number of data is known as data augmentation. After that point in this report, all results have obtained with augmented data. At below figures, you can see results for 1D ConvNet Model with Mel Spectograms.





	precision	recall	f1-score	support
sad	0.80	0.75	0.77	407
angry	0.55	0.58	0.56	357
disgust	0.77	0.37	0.50	358
fear	0.67	0.60	0.63	390
happy	0.45	0.82	0.58	339
neutral	0.60	0.53	0.56	382
accuracy			0.61	2233
macro avg	0.64	0.61	0.60	2233
weighted avg	0.64	0.61	0.61	2233



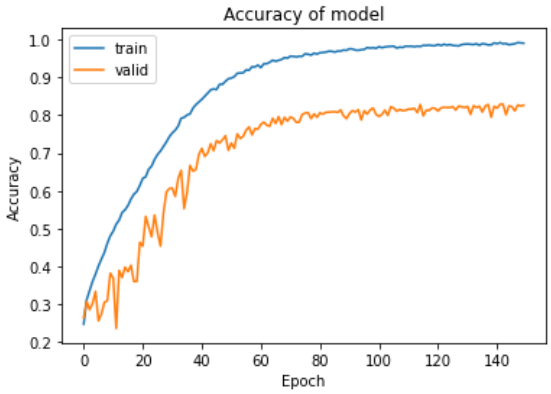


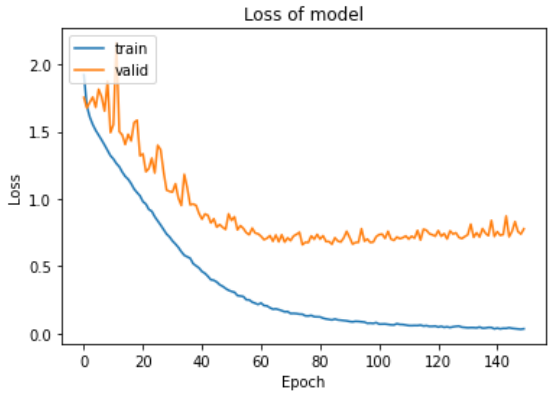
Obtained best accuracy is 64%. This value is not enough for a good classification. This is because taking mean causes loss of data. Therefore my second approach became to use 2D Mel Spectrograms.

2D-Mel Spectrogram

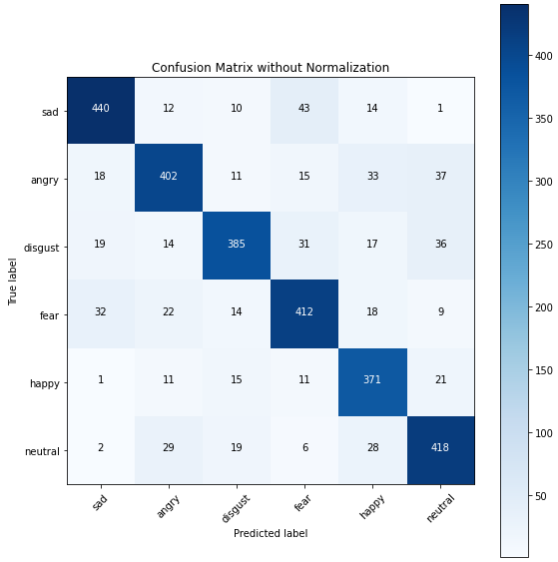
In this approach, I took actual Mel spectrograms of audio files, that's in two dimensions. Summary of my model and results for this model are given below figures.

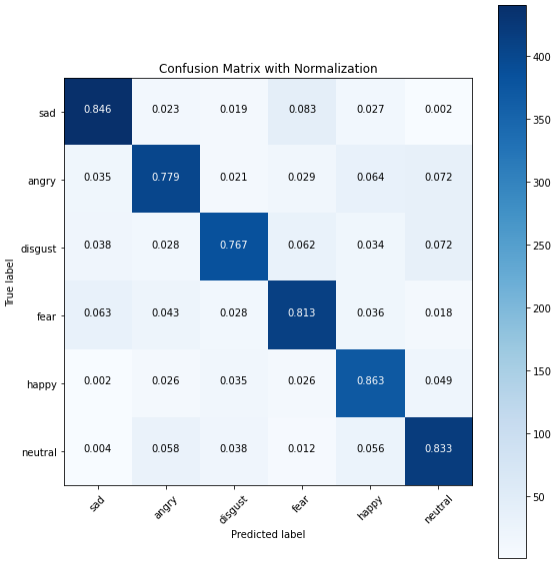
```
Model: "sequential_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_17 (Conv2D)          (None, 128, 63, 256)      2560

 max_pooling2d_17 (MaxPoolin  (None, 64, 31, 256)      0
 g2D)

 dropout_14 (Dropout)        (None, 64, 31, 256)       0

 conv2d_18 (Conv2D)          (None, 64, 31, 128)       295040

 max_pooling2d_18 (MaxPoolin  (None, 32, 15, 128)      0
 g2D)

 dropout_15 (Dropout)        (None, 32, 15, 128)       0

 conv2d_19 (Conv2D)          (None, 32, 15, 64)        73792

 max_pooling2d_19 (MaxPoolin  (None, 16, 7, 64)        0
 g2D)

 dropout_16 (Dropout)        (None, 16, 7, 64)         0

 flatten_5 (Flatten)         (None, 7168)              0

 dense_10 (Dense)            (None, 64)                458816

 batch_normalization_13 (Bat  (None, 64)               256
 chNormalization)

 dense_11 (Dense)            (None, 6)                 390

=================================================================
Total params: 830,854
Trainable params: 830,726
Non-trainable params: 128
_________________________________________________________________
```





	precision	recall	f1-score	support
sad	0.86	0.85	0.85	520
angry	0.82	0.78	0.80	516
disgust	0.85	0.77	0.81	502
fear	0.80	0.81	0.80	507
happy	0.77	0.86	0.81	430
neutral	0.80	0.83	0.82	502
accuracy			0.82	2977
macro avg	0.82	0.82	0.82	2977
weighted avg	0.82	0.82	0.82	2977



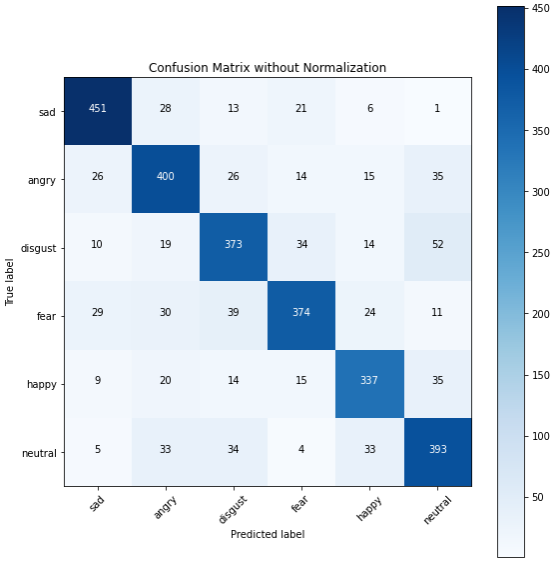


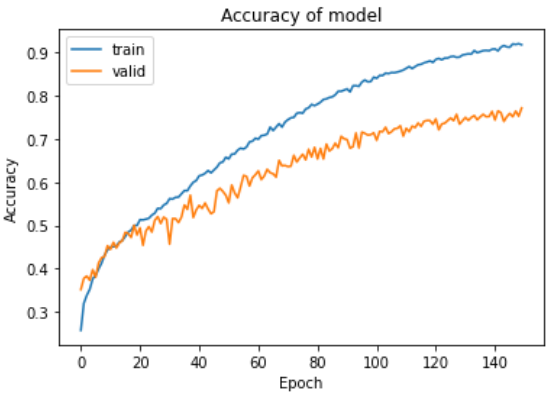
	precision	recall	f1-score	support
sad	0.85	0.87	0.86	520
angry	0.75	0.78	0.76	516
disgust	0.75	0.74	0.75	502
fear	0.81	0.74	0.77	507
happy	0.79	0.78	0.78	430
neutral	0.75	0.78	0.76	502
accuracy			0.78	2977
macro avg	0.78	0.78	0.78	2977
weighted avg	0.78	0.78	0.78	2977

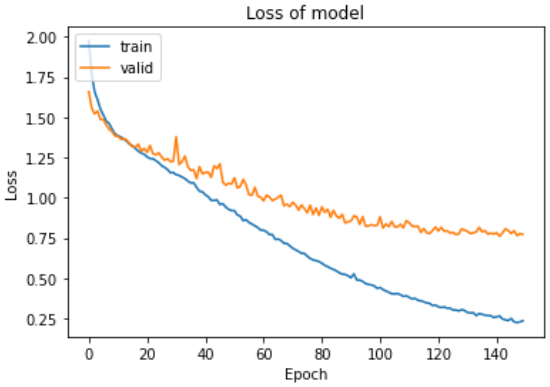
Obtained best accuracy is 82%.

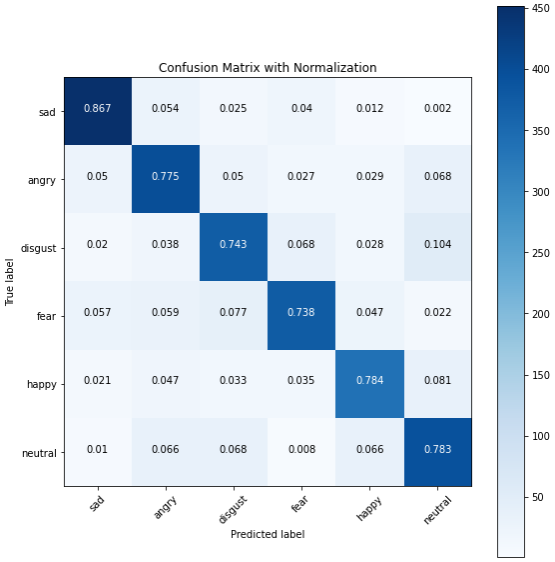
2D-MFCC

I also trained same model by using Mel Frequency Cepstral Coefficients, in short MFCC. This also gave similar results with previous model. **Obtained best accuracy is 78%**



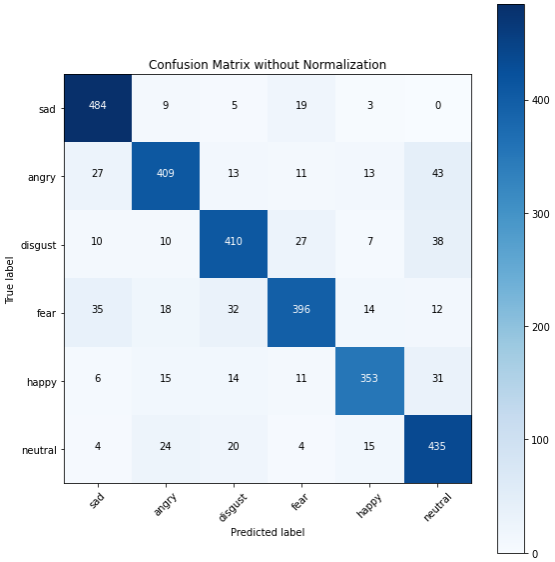


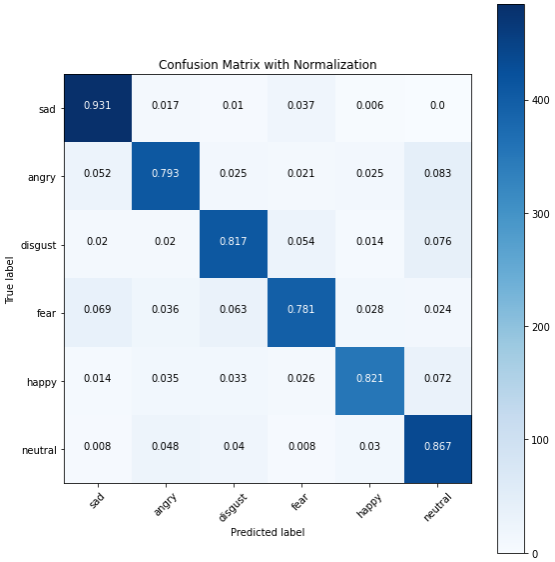


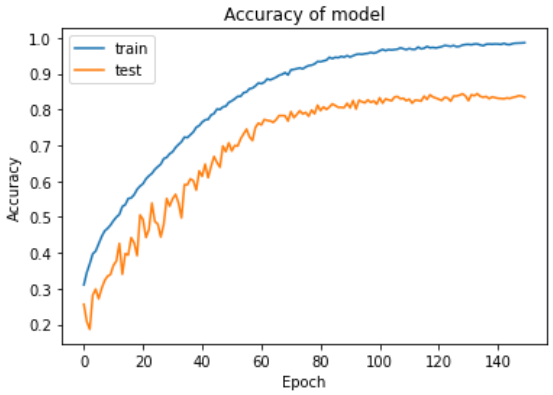


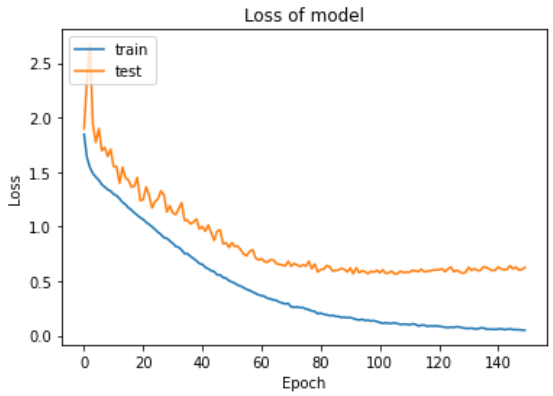
2D-Mel Spectrogram&MFCC

After these results, I tried to improve my models. To reach this aim, I made classification by using Mel spectrograms and MFCCs together. I used previous model again in this approach. Only difference is at last dense layer: I trained model with Mel spectrograms and MFCCs separately however I did not made classifications. To made classification I took outputs of last layers from both Mel spectrograms and MFCCs and gave them into a new model, which consists of 2 dense layer, one of them for classification. As a result of this combined model, **accuracy increased to 84%.** You can also see block schema of this model at next page.

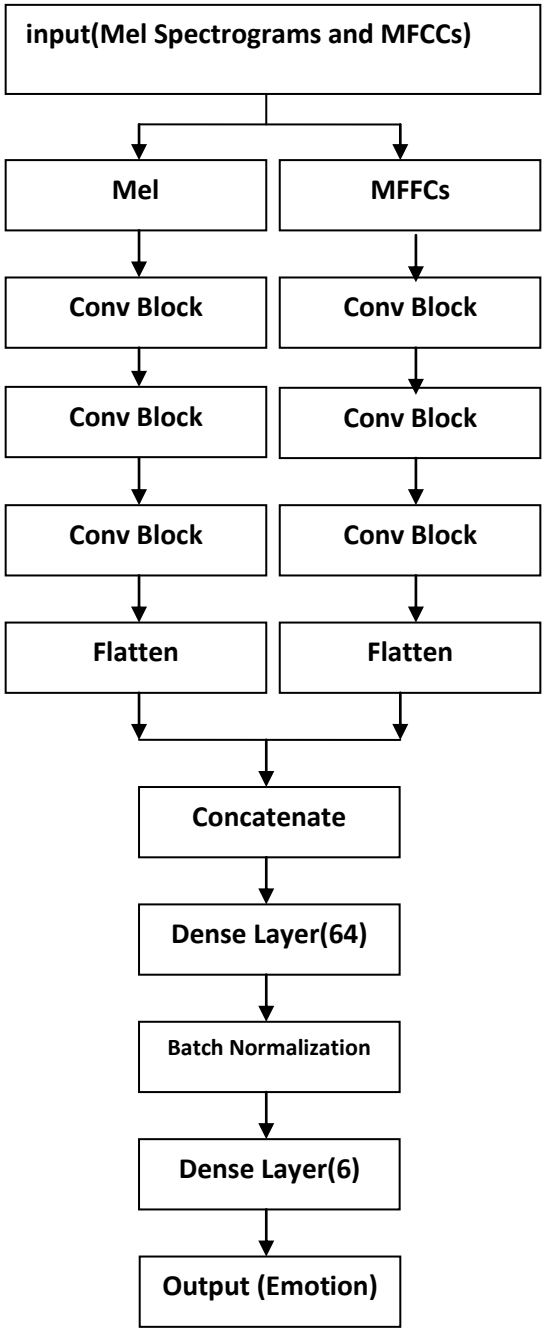




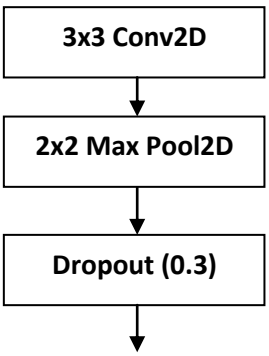




	precision	recall	f1-score	support
sad	0.86	0.93	0.89	520
angry	0.84	0.79	0.82	516
disgust	0.83	0.82	0.82	502
fear	0.85	0.78	0.81	507
happy	0.87	0.82	0.85	430
neutral	0.78	0.87	0.82	502
accuracy			0.84	2977
macro avg	0.84	0.83	0.83	2977
weighted avg	0.84	0.84	0.83	2977


input(Mel Spectrograms and MFCCs)
Mel
MFFCs
Conv Block
Conv Block
Conv Block
Conv Block
Conv Block
Conv Block
Conv Block
Conv Block
Flatten
Flatten
Concatenate
Dense Layer(64)
Batch Normalization
Dense Layer(6)
Output (Emotion)

Conv Block


3x3 Conv2D
2x2 Max Pool2D
Dropout (0.3)