Pattern Recognition
Assignment 3
Speech Emotion Recognition

Team Members

Name	ID
Ramy Ahmed ElSayed	19015649
Zyad Samy Ramadan	19015720
Abdelmoniem Hany	19017359

Problem Statement

Speech is the most natural way of expressing ourselves as humans. It is only natural then to extend this communication medium to computer applications. We define speech emotion recognition (SER) systems as a collection of methodologies that process and classify speech signals to detect the embedded emotions.

Data Set

The dataset used in the assignment is from the ”CREMA” dataset which contains 7000+ entries of .wav audio files that express the following emotions:

- SAD - sadness;
- ANG - angry;
- DIS - disgust;
- FEA - fear;
- HAP - happy;
- NEU - neutral.

Loading The Data

```
def loadData(datasetPath, samplingRate, length):
  audios = []
  emotions = []
  duration = length * samplingRate
  for audioFile in os.listdir(datasetPath):
    path = datasetPath + "/" + audioFile
    audio, samplingRate = librosa.load(path, sr = samplingRate)

    if len(audio) < duration:
      padding = duration - len(audio)
      offset = padding // 2
      audio = np.pad(audio, (offset,padding - offset), 'constant')
    else:
      audio = audio[:length * samplingRate]

    audios.append(audio)
    info = audioFile.partition(".wav")[0].split("_")
    emotions.append(emotionsMap[info[2]])
  return np.array(audios), np.array(emotions)
```

Creating The Feature Spaces

Time Domain

For the 1D feature spaces, the features extracted were as follows:

- Energy.
- Zero Crossing Rate.

Note: Other features were added in experimenting like the RMSE and the Spectral Centroid but that led to severe underfitting and overall worse results.

Frequency Domain

For the 2D feature space, the features extracted were as follows:

- Mel Spectrogram.

Creating The Models

1D CNN Model

The model consists of 4 Convolution layers and 2 FD classification layers. The initial model chosen was the one mentioned in the assignment report but the average best accuracy obtained from such model was around 35~40% so we added an additional conv layer to allow the features to be more concrete and retain a decent part from the original input.

2D CNN Model

The model consists of 5 Convolution layers and 2 FD classification layers. Unlike the 1D model, there are a lot more features to retain/propagate through the layers so it was found that adding an additional layer increased our overall accuracy.

Notes:

- Batch normalization was used to avoid overshoot in the output values.
- Dropout layers were used to deal with the problem of overfitting.
- In the 2D CNN model, the number of filters was decreased drastically than what was mentioned in the assignment report, as it introduced a lot of learnable parameters which took a lot of time to train and contributed more to the problem of overfitting.

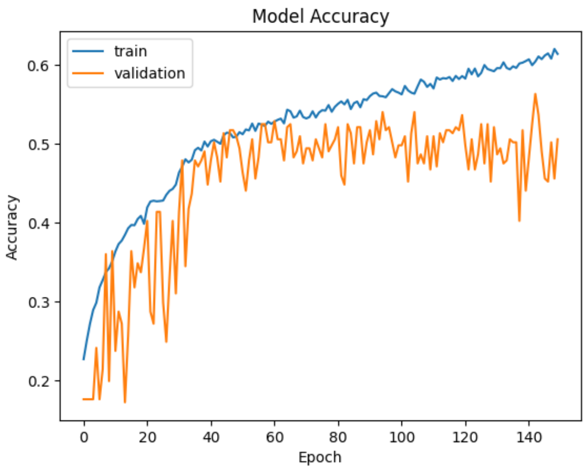
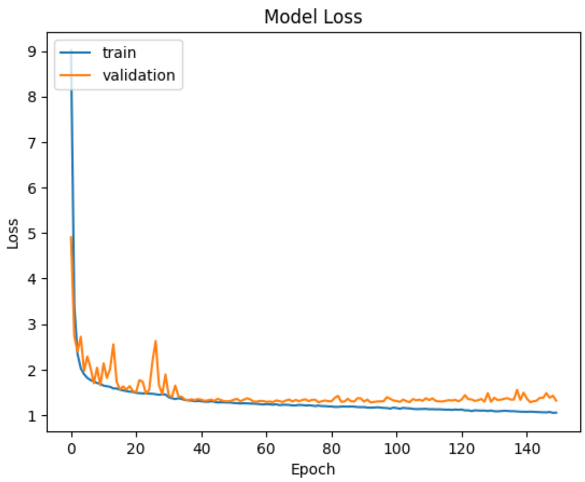
Results

Epochs = 150

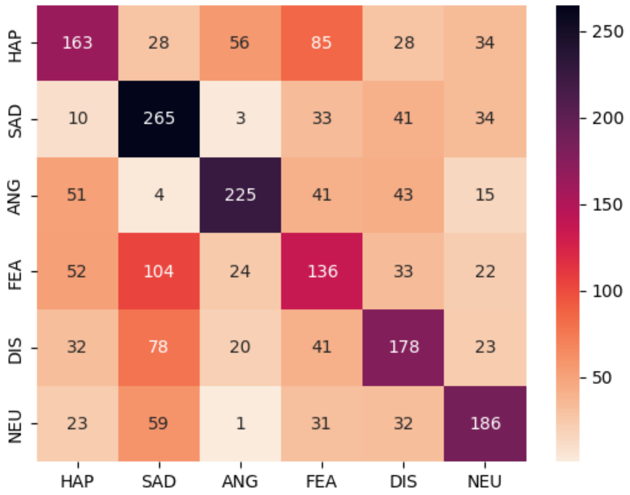
Model	1D (512, 512, 256, 128)	2D (64, 64, 16, 16, 128)
Trainable Parameters	2,269,318	1,695,718
Training Accuracy (Best)	61,39%	99.4%
Validation Accuracy (Best)	50.5%	55.9%
Test Accuracy (Best)	51.6%	51.9%

Model Evaluation

1D CNN Model

Accuracy	Loss
	

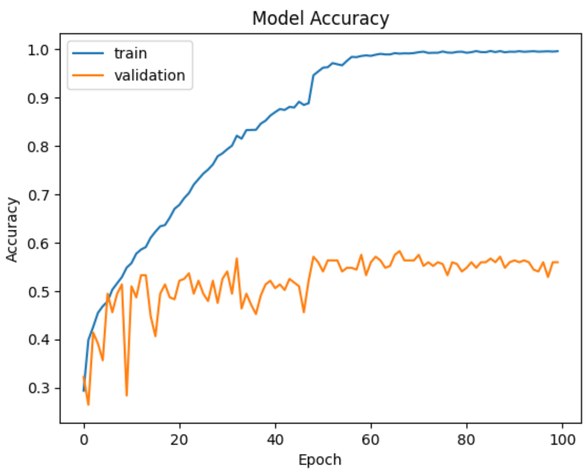
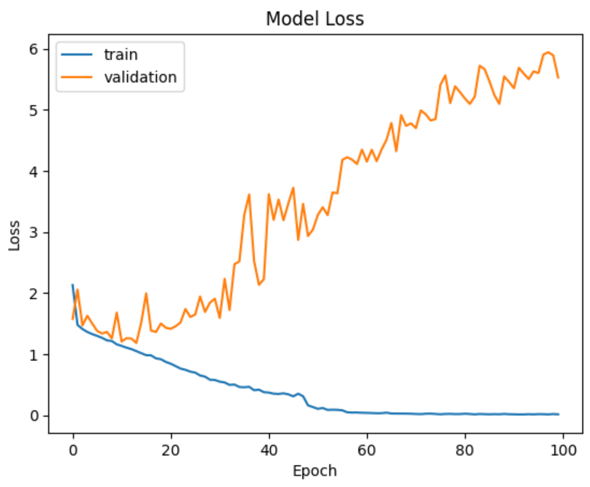
Confusion Matrix



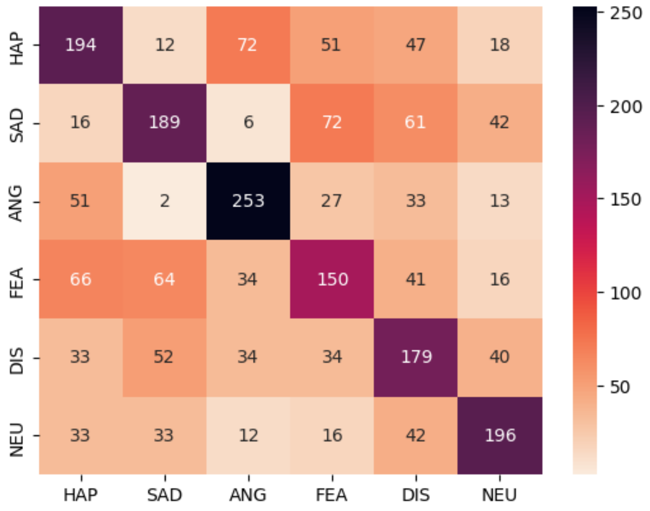
	HAP	SAD	ANG	FEA	DIS	NEU
HAP	163	28	56	85	28	34
SAD	10	265	3	33	41	34
ANG	51	4	225	41	43	15
FEA	52	104	24	136	33	22
DIS	32	78	20	41	178	23
NEU	23	59	1	31	32	186

It can be seen that the fear class is the one causing the most problems in our model and the one that has the least accuracy.

2D CNN Model

Accuracy	Loss
	

Confusion Matrix



Conclusion

The overall accuracy of the model is on the lower side and that’s for 2 main reasons:

- The lack of data used for training/testing the model which causes overfitting in some cases (2D)
- CNNs are not that great when it comes to audio recognition while other models are much better (RNNs, LSTMs, etc…)