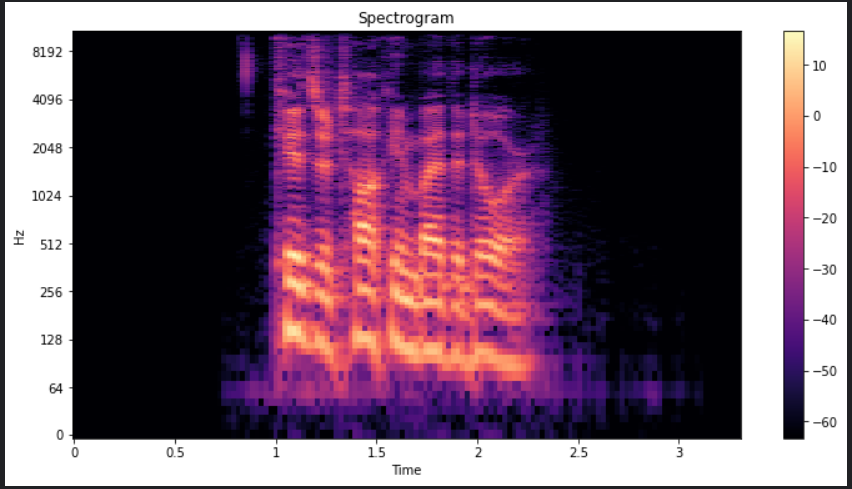
**Week 4**

1. **Speech Emotion Recognition using Spectrogram images:**

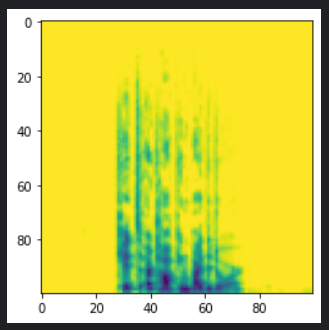
Pre-processing:

* The same dataset (Ravdess song+speech) was used to train a CNN model with spectrogram images.
* A basic visualisation of spectrogram images was done to visualise how the spectrogram images actually look.



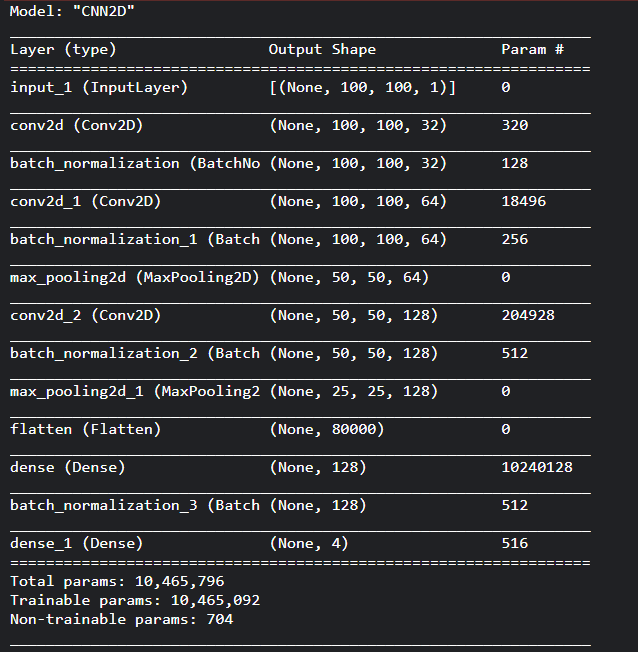
* The opensoundscape python module was used convert audio files into spectrogram images and extract these images and save them as numpy arrays.
* PIL (Python Imaging Library) module was used to resize/reduce images into the required size which helps reduce the computation time of the convolutions (images are converted to 100x100 size).
* The emotions were encoded (for example anger = [1,0,0,0] , happy = [0,0,0,1])

Sample spectrogram image after pre-processing:



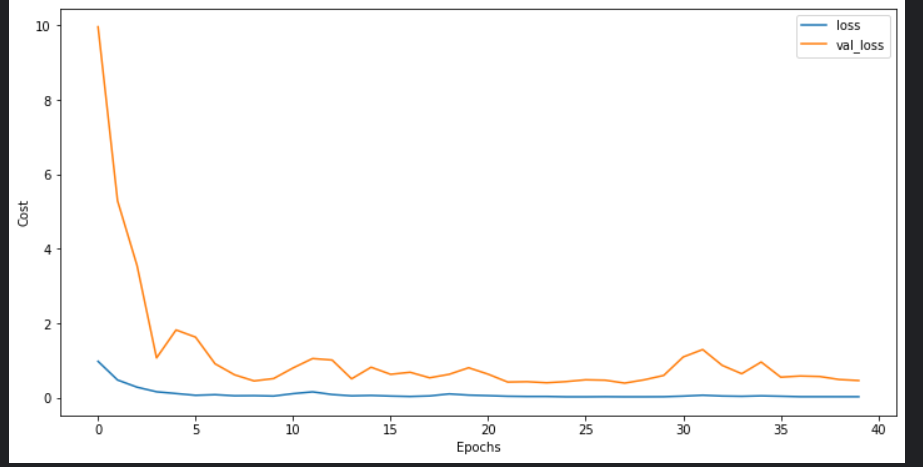
Modeling:

* A CNN-2D model was initialised with the following architecture:



* The data(images) were split into the training and the test set and the dimensions of the data was adjusted depending on the input size of the cnn model.
* The model was fit with a categorical cross entropy loss function and with an Adam optimizer with a learning rate of 0.0001 and trained .

Loss vs Val\_loss plot:



* A final validation accuracy of 82.5% was achieved with the model.

**2)Testing with a different dataset:**

* The dataset used for testing the trained models was the EMODB dataset which consists of audio samples of professional speakers (five males and five females). The database contains a total of 535 utterances. The EMODB database comprises of seven emotions: 1) anger; 2) boredom; 3) anxiety; 4) happiness; 5) sadness; 6) disgust; and 7) neutral.



Testing accuracies:

|  |  |  |
| --- | --- | --- |
| Model Name | Validation Accuracy | Testing accuracy with EMODB |
| Convolutional Neural Network | 86.4 | 43.1 |
| Multi Layer Perceptron | 86.3 | 54.5 |
| Support vector Machine | 72.3 | 19.1 |
| Random Forest Classifierr | 80.8 | 31.1 |
| CNN using Spectrogram images | 82.5 | 20.75 |

**3)Training using K Folds Cross Validation:**

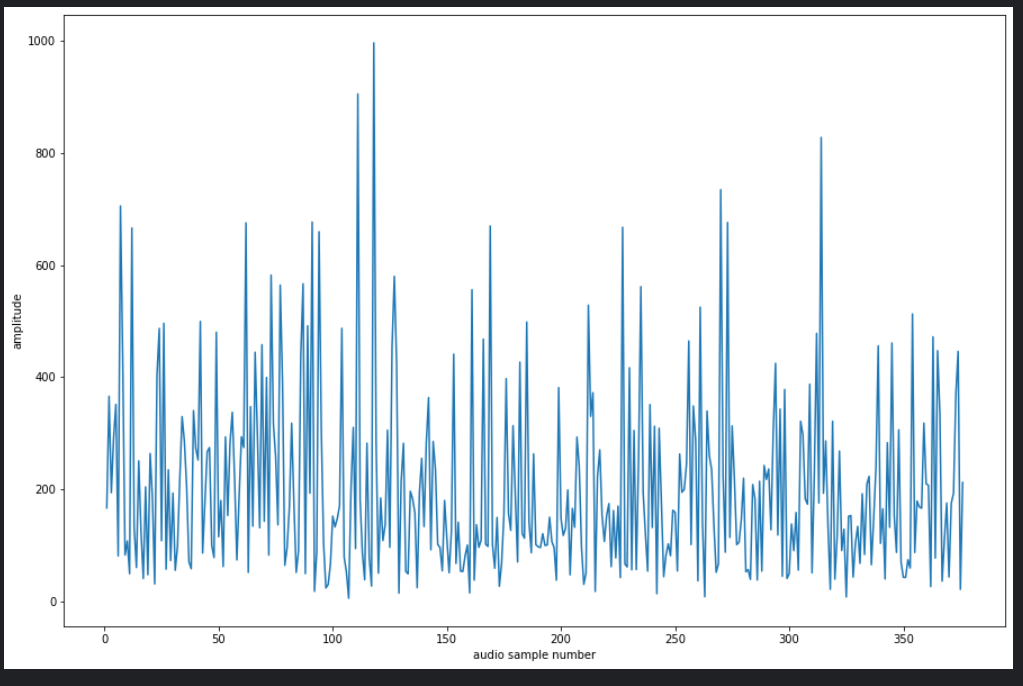
* K Folds Cross Validation is a technique to generalize the trained model to make sure that overfitting does not occur when there is a scarcity of training data.
* K Folds validation consists of splitting data into k splits or folds. In the first iteration the model is trained using k-1 folds and tested using the fold left out and this occurs for k iterations. (until the model has been validated with all folds)
* K Folds validation was performed to reduce overfitting and help generalize the model so that the model performs better on the different dataset(EMODB)

Results after 3 fold Cross Validation:

|  |  |  |  |
| --- | --- | --- | --- |
| Model Name | Validation Accuracy | Testing accuracy with EMODB before K folds cross validation | Testing accuracy with EMODB after K folds Cross validation |
| Convolutional Neural Network | 86.4 | 43.1 | 60.1 |
| Multi Layer Perceptron | 86.2 | 54.5 | 56.4 |
| Support vector Machine | 72.3 | 19.1 | 22.3 |
| Random Forest Classifierr | 80.8 | 31.1 | 30.4 |
| CNN using Spectrogram images | 82.5 | 20.75 | - |

**4)Use Case:**

* Writing a python function to evaluate the level of risk the driver is at.
* Fourier Transform was performed on speech samples , if the emotion identified is anger , the maximum amplitude at frequencies > 450Hz is identified from the fft results and if this value is greater than a threshold then the driver is identified as “at risk”.
* The threshold value was calculated by analysing the max amplitude values for different audio samples and the value at the 80th percentile was taken as the threshold .
* Visualisation of max amplitude for different audio samples:



Percentile vs Amplitude plot:

