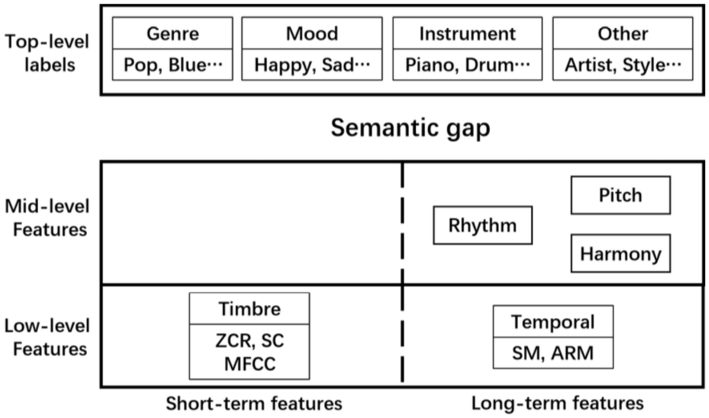
Music Genre Classification with LSTM, 2D-CNN and 1D-CNN based on Time and Frequency Domain Features

Anuraag Velamati, Naresh Kumar Kaushal, Vikraman Senthil Kumar

The Problem Statement

This project aims to classify music genre using deep feature extraction by LSTM. This problem is originally solved by Yinhui et al [8] and we made an attempt to implement this paper and compare it further with 1D and 2D CNN.

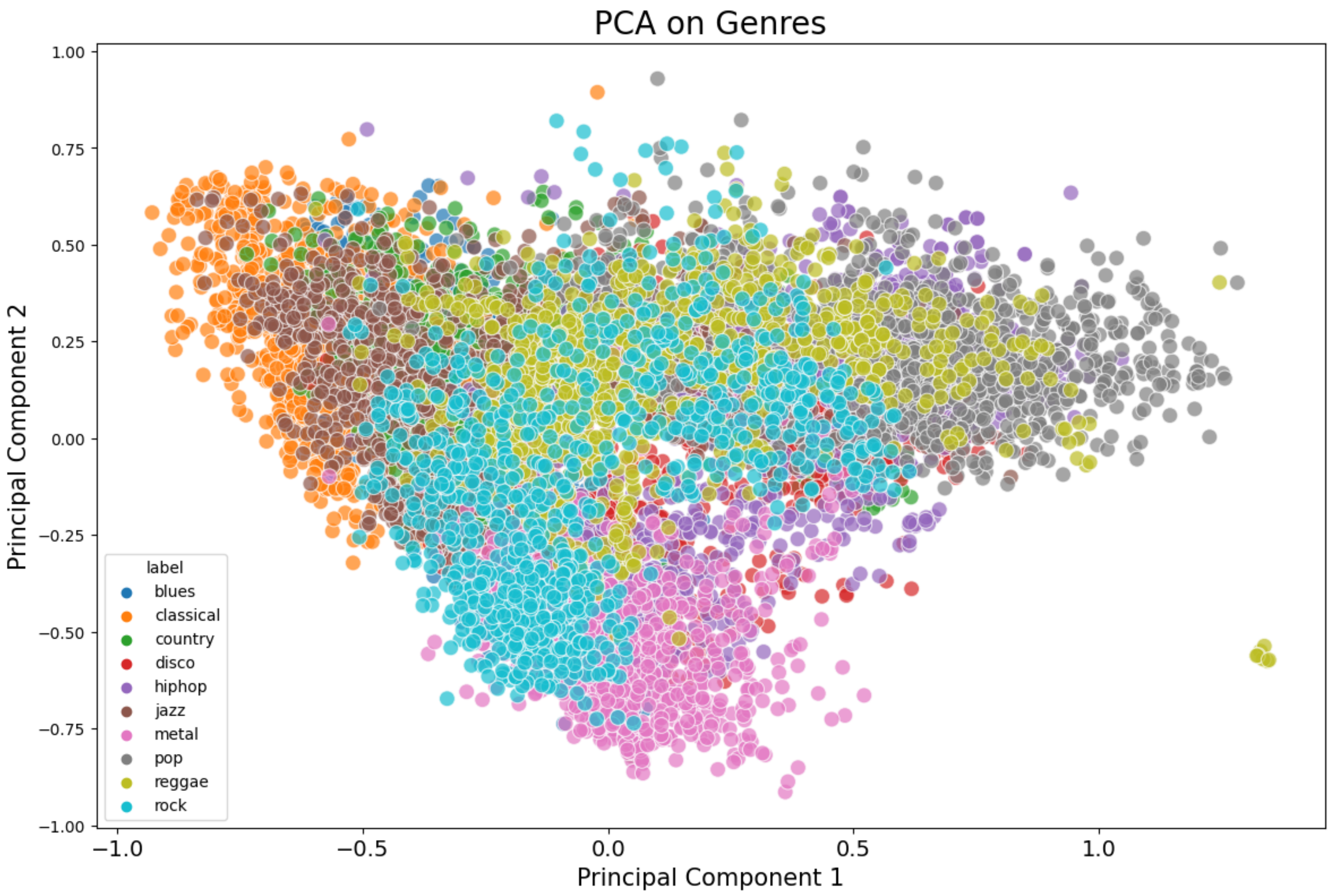
Motivation



Human's understanding of music is greatly influenced by semantic labels like genre, mood, instruments used and other features like artist, style etc. But they cannot be obtained from low level features or mid level features because of the semantic gap that exists. Hence, there is a need for an algorithm that can bridge this gap.

Short-term features are less useful for classifying music genres than deep features produced by deep learning models. So in order to produce deep features and classify music genres, this project employs an LSTM model and compares it with other models like KNN, SVM, 1-D CNN and 2-D CNN.

Dataset

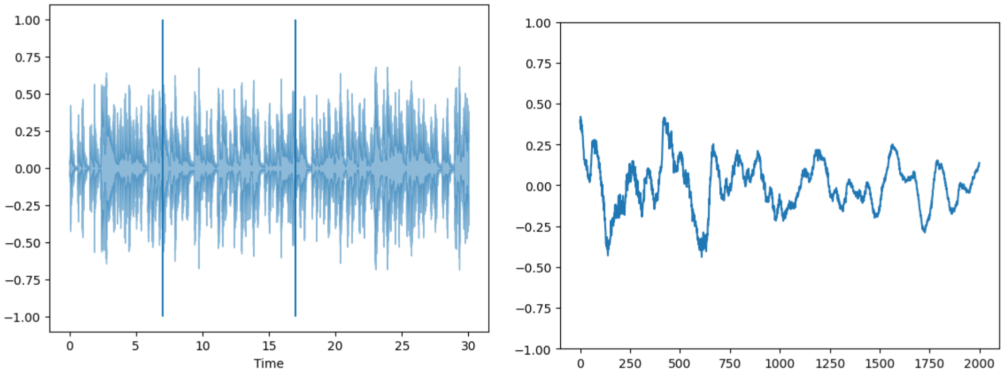


The dataset that we are using is available on Kaggle. It is a music collection consisting of 10 music genres with 100 audio files for each genre. The 10 music genres in the collection are blues, classical, country, disco, hiphop, jazz, metal, pop, reggae and rock. The audio files are saved in waveform with a length of 30 seconds.

Visualisation

By Vikraman Senthil Kumar

To visualise the audio dataset in time domain, we used *librosa.display.waveshow*.



(a) 30 sec Audio waveplot with 'blues' genre (b) 2000 samples from 7sec to 17sec

Similarly we can visualise time domain features like zero crossing rate, which tells us the rate at which the signal changes it sign. From figure below we can see that classical and jazz have lower zcr values compared to other genres.

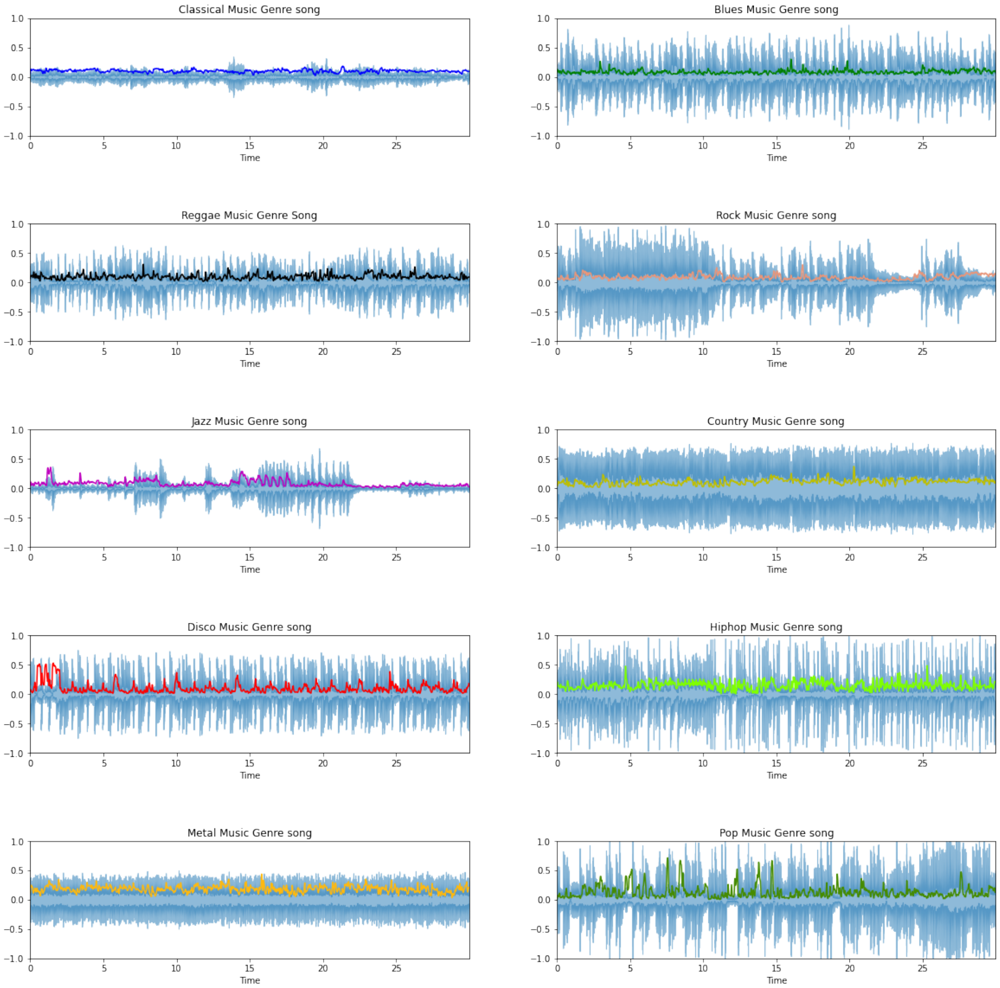
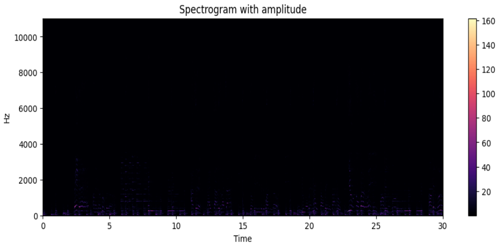
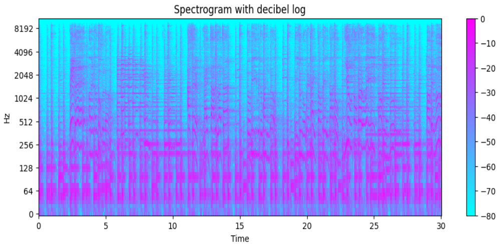


Figure 2: Zero crossing rate for music with different genres

To view frequency domain features we first take short time fourier transform (stft) of the audio files with window size of 2048 and hop length of 512 to get a spectrogram which you can think of as a bunch of FFTs stacked on top of each other. It is a way to visually represent a signal's loudness, or amplitude, as it varies over time at different frequencies. But unfortunately this spectrogram doesn't have much information for us to analyse this is because we humans preceive sounds which is concentrated in narrow range of frequencies and amplitudes so we have to convert the linear scale to log scale to clearly see audible frequencies.

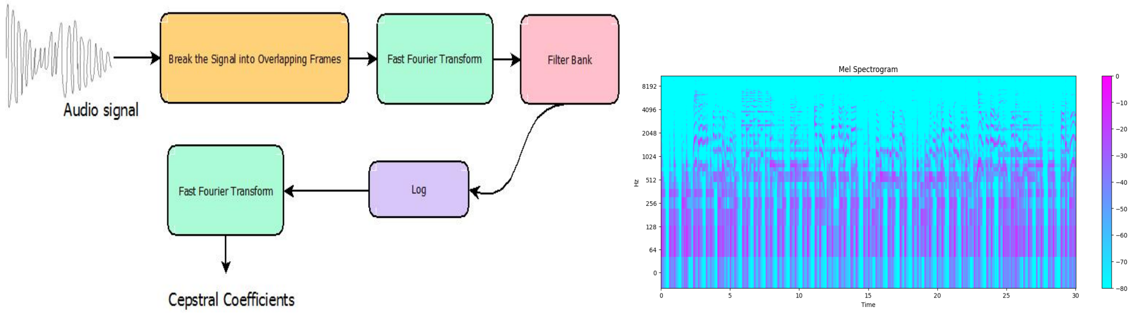


(a) Spectrogram with amplitude



(b) Log spectrogram

Since human's perception of sound is not linear i.e. we can easily tell the difference between lower frequencies but not between the ones on higher end. So we used a Mel-scale to obtain a set of features called Mel frequency cepstral coefficients or MFCCs for our classification task. Figure 2 shows how 30 MFCC's evolve with time.



(a) Procedure to extract MFCC's

(b) 30 MFCC's

Pre-processing

(By Naresh Kumar Kaushal, Anuraag Velamati, Vikraman Senthil Kumar)

For pre-processing we perform data normalisation, noise-filtering, silence reduction and removing any inconsistent values. Since GTZAN dataset has only 1000 audio files which is not enough to train our model efficiently so we perform data augmentation by splitting our 30 sec audio files into 10 segments of 3 sec each and then we divide this 3 sec file into 30 time steps and for each time step we calculated zcr values which finally gave us the feature set with dimension of 30 x 1, similarly we can calculate 30 mfcc's for each segment. The output MFCC feature set is a two-dimensional array with 30 x 1293. But since we cannot input this to LSTM. Therefore, the mean and variance of each row were calculated as features and the final shape of MFCC feature set is obtained i.e. 60 x 1.

Methodology

1. Long Short Term Memory (LSTM)

(By Naresh Kumar Kaushal)

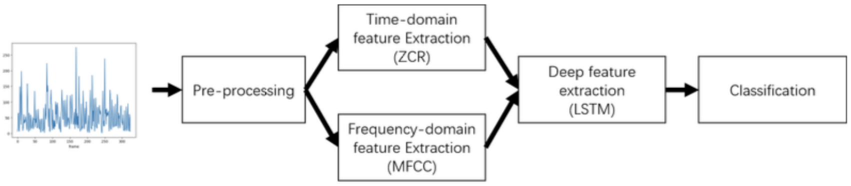

Pre-processing
Time-domain feature Extraction (ZCR)
Frequency-domain feature Extraction (MFCC)
Deep feature extraction (LSTM)
Classification

Figure 5: *Genre Classification Using LSTM*

The two short-term features i.e. zero crossing rate (ZCR) (time domain feature) and mel-frequency spectral coefficients or MFCC's (frequency domain feature) that we calculated in pre-processing stage are then fed to the LSTM with 128 units. Then we train our LSTM network and extracted the output of the LSTM layer which has the dimension of (Number of samples X 128). There are two kinds of outputs depending on the input feature set: (1) deep feature generated from ZCR; (2) deep feature generated from MFCC; finally we used this output as feature set to classify the genre of music using support vector machine (SVM) and k-nearest neighbors (KNN).

The benefit of LSTM is that, on the one hand, music features are time-serial, making them appropriate for LSTM. On the other hand, LSTM was created to address the vanishing gradient issue with recurrent neural networks (RNN). The LSTM's memory cells provide it the ability to retain and access data for extended periods of time and produce better long-term features. LSTM's usually tend to overfit and to overcome this issue we used batch normalization and dropout layers to bring down the complexity of the network.

2. 2D-Convolutional Neural Network (CNN)

(By Anuraag Velamati)

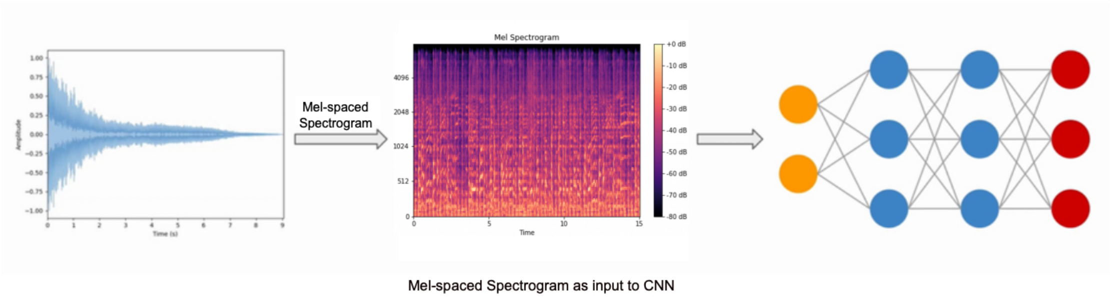


Figure 6: *Genre Classification Using 2D-CNN*

For further analysis we compared the LSTM model with the baseline 2D-CNN model. For the classification purposes, we have used Mel-Spectrogram as our primary time-frequency domain feature. The primary reason for deciding on this is the mel-scale, which is a perceptual scale of pitches judged by listeners to be equal in distance from one another. The mel-scale groups higher frequencies exponentially, thereby keeping track of frequencies at a higher level, which are not perceivable to the normal human ears. Moreover, as Mel-spectrograms yields a visual representation of intensity and spectrum of the frequency, it will be a good choice as CNN's are good at learning from visual representations. The generated mel-spectrograms are fed then to the network for classification.

After training the model on the dataset, we realised that the model was overfitting easily. So, in order to increase the robustness of the model, data augmentation has been done to the GTZAN dataset. We have made pitch shifts and the addition of white noise, along with splitting the audio files into slices. As deep learning models have a tendency to overfit, we also have had to take care of the complexity of the model. In order to achieve that, we employed dropout layers to make sure that the model was not complex.

3. 1D-Convolutional neural Network (CNN)

(By Vikraman Senthil Kumar)

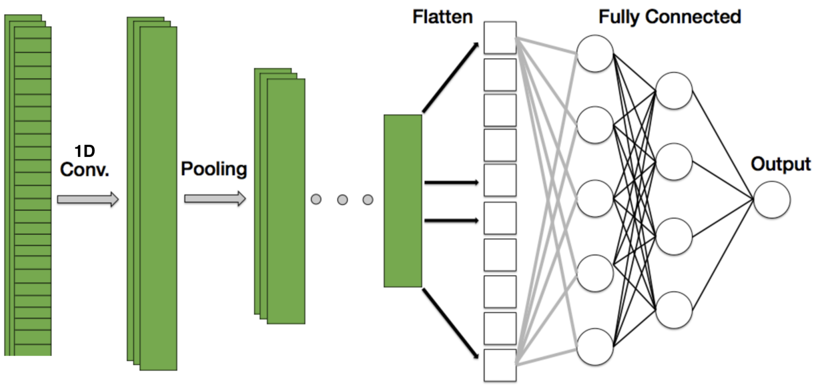


Figure 7: *Genre Classification Using 1D-CNN*

Apart from 2D-CNN we compared our LSTM model with baseline 1D-CNN architecture. One of the key challenges of CNN architectures in context to genre classification is variable-length signals which occurs when the inputs are not of constant length. We tackled this challenge by limiting our audio files to 15 sec duration. Then we calculated 30 MFCCs which outputs a 2D array (30 X 646) which we then feed into our 1D-CNN model by transposing it (646 x 30) where 646 are time steps and 30 is the count of MFCCs.

Our model consist of 3 1D-CNN layers with ReLU activation, dropout layer (To avoid overfitting), average pool layer (to down sample input) and batch normalisation (to speed up the training proces and make it more stable) which is then stacked with 1 flatten and 2 Dense layers.

Experimental results and Analysis

Classification accuracy without LSTM		
Short-term Feature	ZCR	MFCC
SVM	23.67%	64.33%
KNN	19%	31.67%

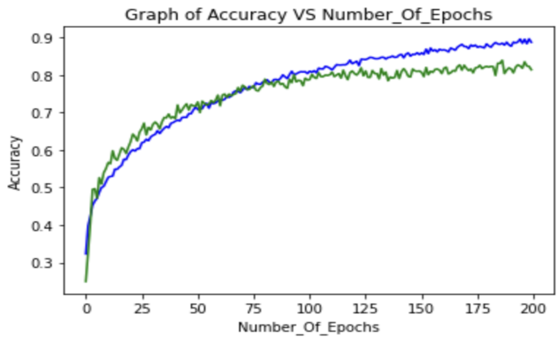
Table above shows the accuracy of music classification without deep feature extraction layer (LSTM layer). The lowest classification accuracy is 23.67% when music is classified by SVM based on ZCR. While changing the classifier to KNN, the accuracy is still low. However, The classification accuracy significantly increases to 64.33% when music is classified by SVM based on MFCC. Table also shows that compared to KNN, SVM has better performance in all aspect.

Classification accuracy with LSTM		
Short-term Feature	ZCR	MFCC
SVM	44.4%	81.8%
KNN	41.25%	77.8%

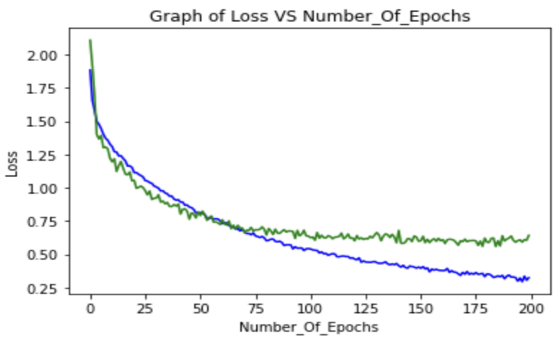
Table above shows the classification accuracy using deep feature extraction layer. Compared to previous table, classification accuracy increases greatly in most cases regardless of SVM classifier based on ZCR. While classifier changes to KNN, the accuracy of the model based on ZCR increases by more than 117%, from 19.0 to 41.25. The highest classification accuracy appears on SVM classifier based on the MFCC feature with 81.8.

Classification accuracies for 1D and 2D CNN	
1D-CNN	2D-CNN
62.1%	75.8%

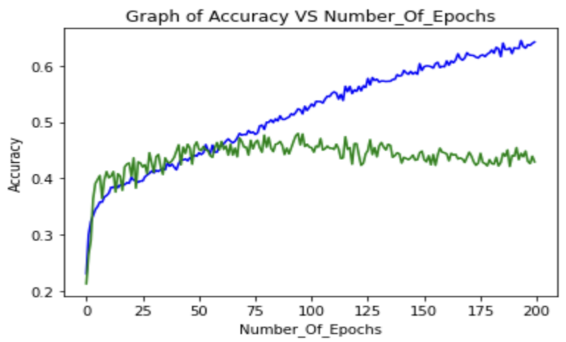
Table above shows accuracies for 2D-CNN and 1D-CNN and we can observe that the LSTM model that is proposed in this research paper is performing better than the above two. But we observed that the 2D-CNN was pretty close to our LSTM model. We wanted to experiment further with 2D-CNN, but we could not do it due to computational constraints.



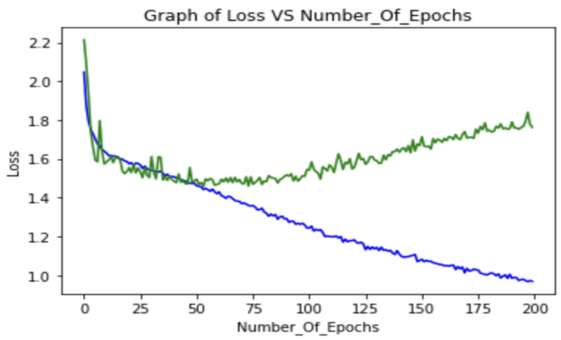
(a) Accuracy curve for LSTM training with only MFCC features



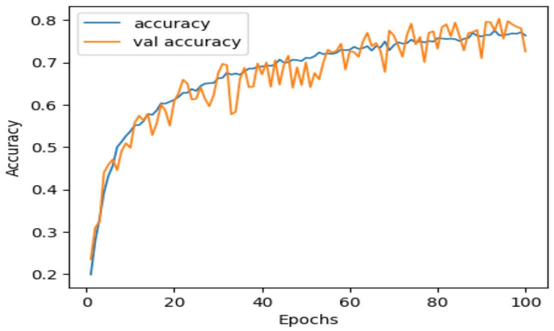
(b) Loss curve for LSTM training with only MFCC features



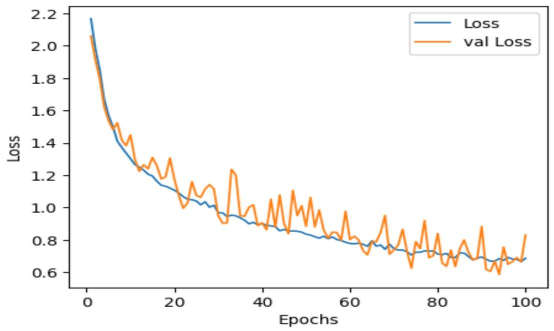
(a) Accuracy curve for LSTM training with only ZCR features



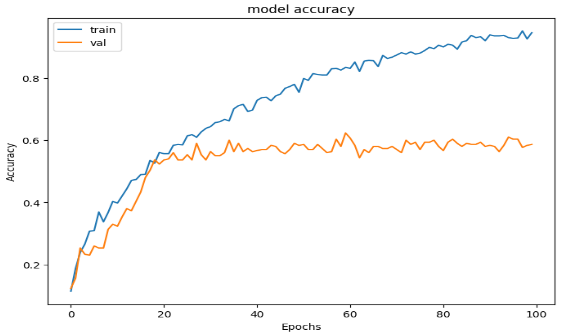
(b) Loss curve for LSTM training with only ZCR features



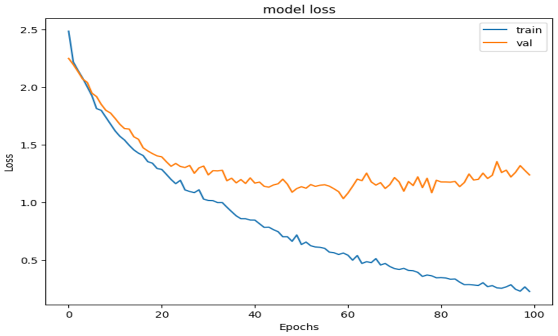
(a) Accuracy curve for 2D-CNN training



(b) Loss curve for 2D-CNN training



(a) Accuracy curve for 1D-CNN training



(b) Loss curve for 1D-CNN training

Future work

1. Test the current model on more robust dataset like FMA [4]
2. Use more kinds of low-level features like spectral centroid, RMS, chroma etc.
3. Optimising the deep learning model by using bi-directional LSTM or advance LSTM.
4. Further explore the relationship between deep features and mid-level features.

The Team

For further queries feel free to email : Vikraman Senthil Kumar | Anuraag Velamati | Naresh Kumar Kaushal

References

[1] Namrata Dutt. *Music genre classification using CNN: Part 2- Classification.*

[2] Priya Dwivedi. *Using CNNs and RNNs for Music Genre Recognition.*

[3] Lin Feng, Shenlan Liu, and Jianing Yao. *Music Genre Classification with Paralleling Recurrent Convolutional Neural Network.*

[4] FMA. *https://www.kaggle.com/datasets/imsparsh/fma-free-music-archive-small-medium.*

[5] Safaa Allamy; Alessandro Lameiras Koerich. *1D CNN Architectures for Music Genre Classification.*

[6] Yanxiong Li, Xianku Li, Yuhan Zhang, Wucheng Wang, Mingle Liu, and Xiaohui Feng. *Acoustic Scene Classification Using Deep Audio Feature and BLSTM Network.*

[7] Eva Garcia Martin. *Music Genre Classification using CNNs.*

[8] Yinhui Yi, Xiaohui Zhu, Yong Yue, and Wei Wang. *Music Genre Classification with LSTM based on Time and Frequency Domain Features.*

[8] [5] [1] [6] [3] [2] [4] [7]