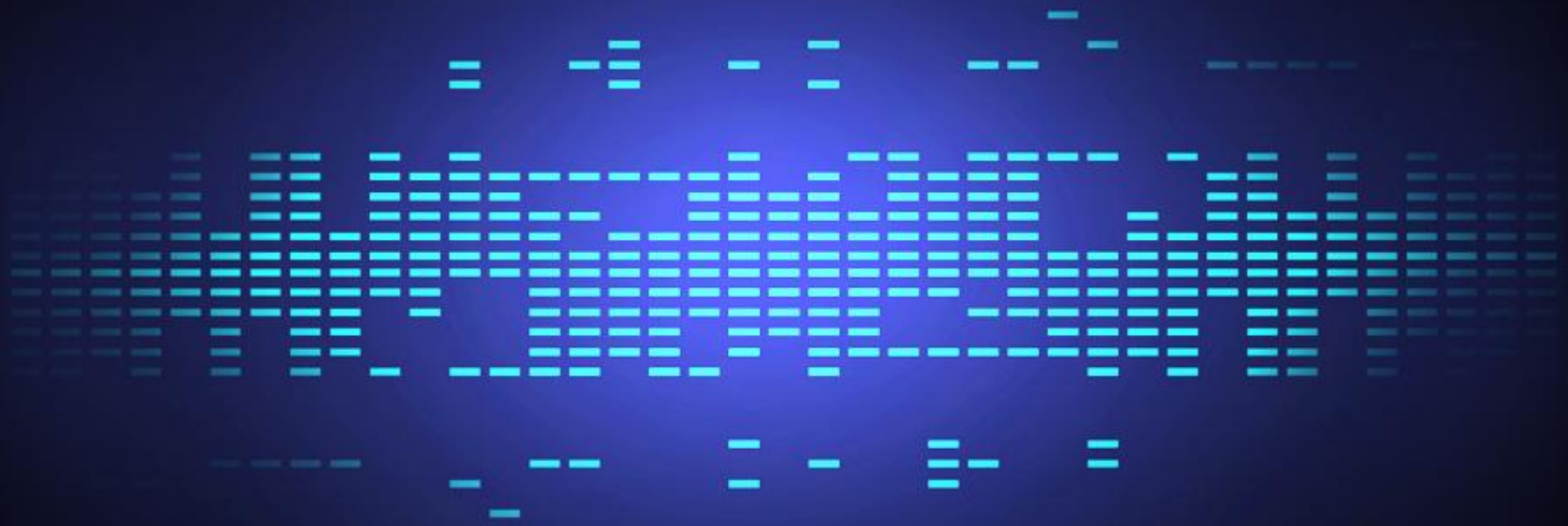
Music Analysis and Genre Classification using Deep Learning



CS 577 Deep Learning

Team:

Manoj Narayan Bisarahalli -A20452726

Vaishnavi Manjunath – A20446043

Abstract

In this project, we use CRNN architecture for music tagging. CRNNs take advantage of CNNs for local feature extraction and RNNs for temporal summarization of the extracted features. We see how to process audio files, train, test a CRNN network and examine the results.

Problem Statement

With an increase in the number of music platforms, it has become difficult to distinguish large number of music manually.

The manual classification involves a need for knowledge of many/all types of genres for a human classifying it. There is a need for quick and efficient methods to identify the genres of these large number of music on different platforms

Proposed Solution

- Pre-process audio files
  - Convert raw audio (30 secs) to mel spectrograms
- Use CRNN network configuration
  - Convolution blocks - extract features from mel spectrograms
  - Recurrent blocks - Learn temporal summarisation of the features
  - Fully connected block - Classify based on inputs

Dataset

Free Music Archive (FMA), an open and easily accessible dataset suitable for evaluating several tasks in MIR, a field concerned with browsing, searching, and organizing large music collections.

dataset	clips	genres	length	size	
			[s]	[GiB]	#days
small	8,000	8	30	7.4	2.8
medium	25,000	16	30	23	8.7
large	106,574	161	30	98	37
full	106,574	161	278	917	343

Table 5: Proposed subsets of the FMA.

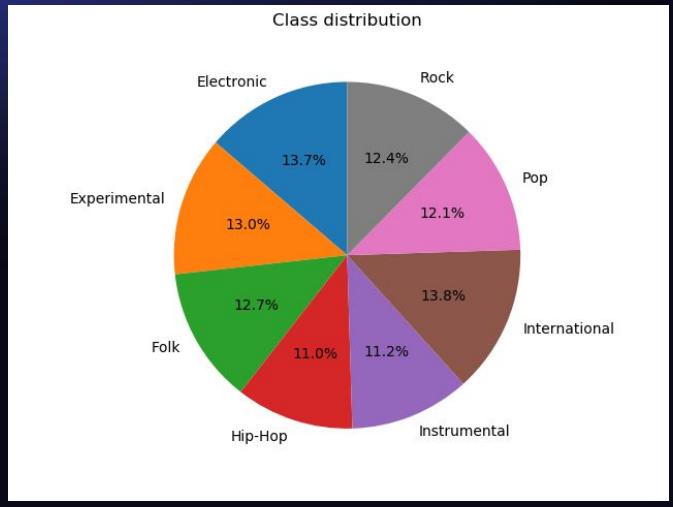
We make use of two subsets from table 5 shown above and focus on **four** top-level genres for each

- Small
- Medium

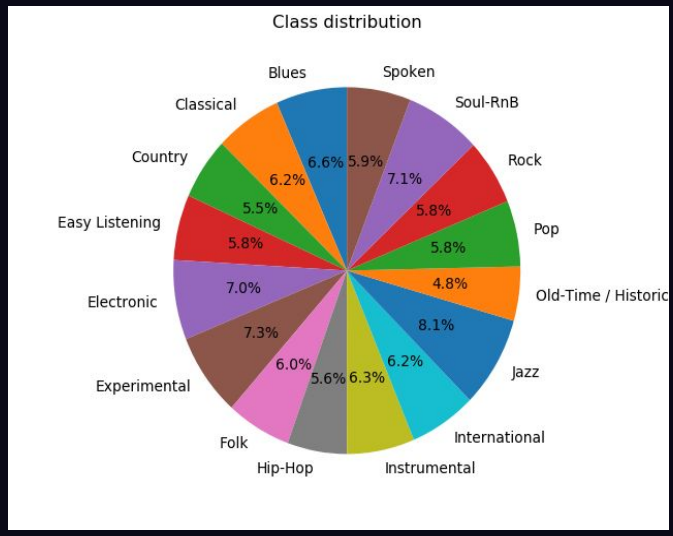
Class Distribution

- Audio Length - 30s for all classes
- Slight imbalance is often negligible as in this case
- We are going to choose a subset of four classes namely:
  - Electronic
  - Rock
  - Hip-Hop
  - Folk

SMALL



MEDIUM

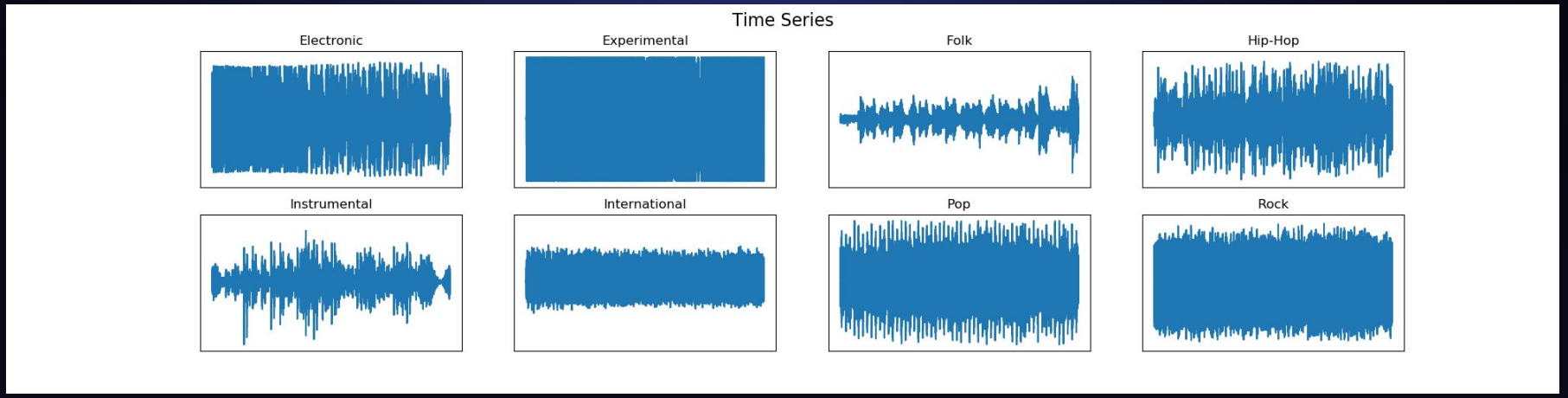


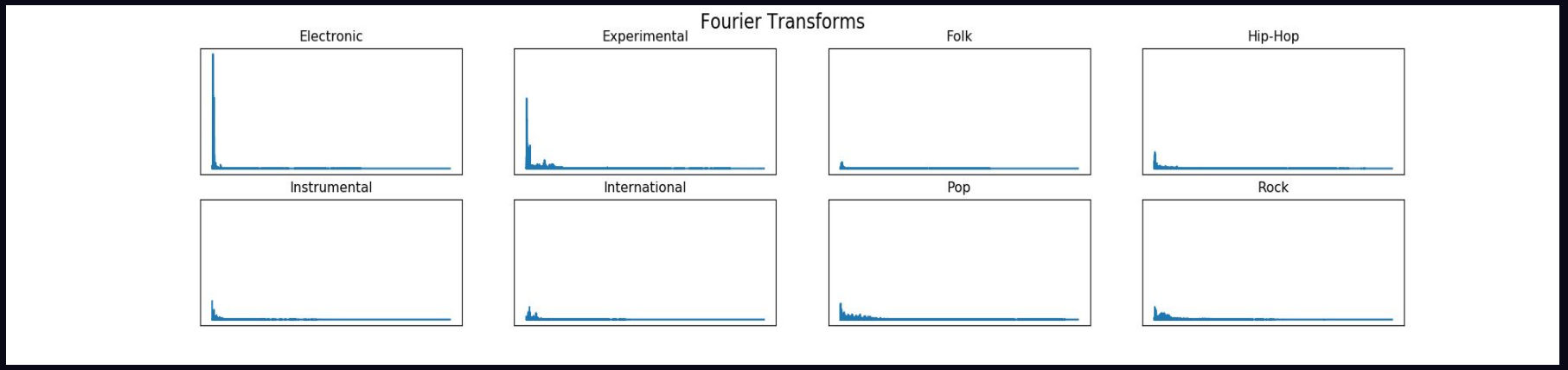
Audio Pre-Processing

Librosa helps with the following

- Retrieve audio files
- Convert raw audio into mel-spectrograms
  - Raw audio of 29.12 secs are used. Shorter clips are padded with zeros and larger ones are clipped
  - Frequency of the resulting me spectrogram is converted to db (decibels)

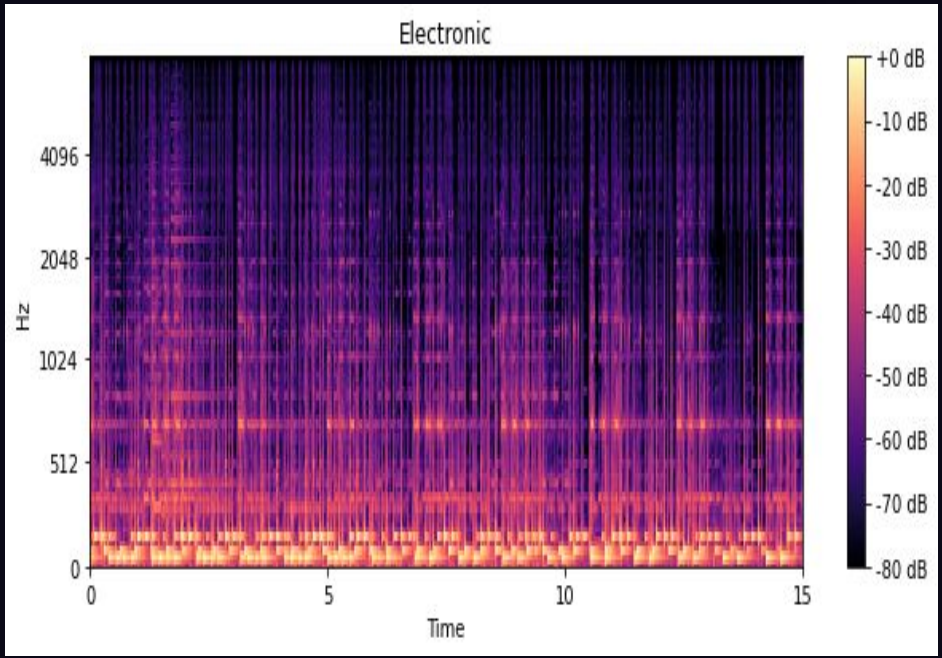
Raw audio signals

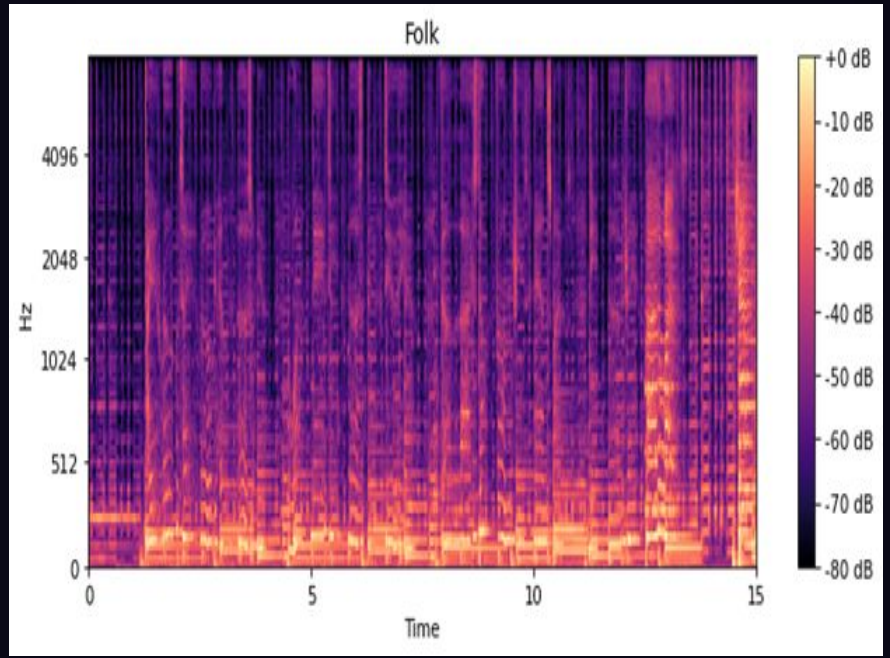




Converted Audio to Mel-Spectrograms

Mel-Spectrograms for Electronic and Folk Genre





Implementation

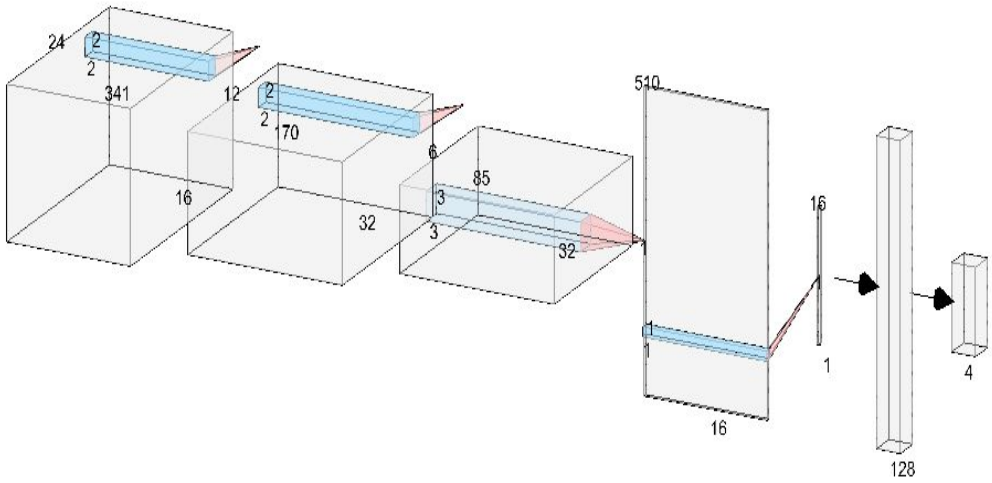
- Keras with tensorflow backend
- Librosa library to process audio files
- Keras' functional API for network building

Model: "model_1"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 1, 96, 1366)	0
conv1 (Conv2D)	(None, 16, 96, 1366)	80
bn1 (BatchNormalization)	(None, 16, 96, 1366)	64
pool1 (MaxPooling2D)	(None, 16, 24, 341)	0
dropout1 (AlphaDropout)	(None, 16, 24, 341)	0
conv2 (Conv2D)	(None, 32, 24, 341)	2080
bn2 (BatchNormalization)	(None, 32, 24, 341)	128
pool2 (MaxPooling2D)	(None, 32, 12, 170)	0
dropout2 (AlphaDropout)	(None, 32, 12, 170)	0
conv3 (Conv2D)	(None, 32, 12, 170)	9248
bn3 (BatchNormalization)	(None, 32, 12, 170)	128
pool3 (MaxPooling2D)	(None, 32, 6, 85)	0
dropout3 (AlphaDropout)	(None, 32, 6, 85)	0
permute_1 (Permute)	(None, 85, 32, 6)	0
reshape_1 (Reshape)	(None, 510, 32)	0
gru1 (GRU)	(None, 510, 16)	2352
gru2 (GRU)	(None, 16)	1584
final_drop (AlphaDropout)	(None, 16)	0
hidden1 (Dense)	(None, 128)	2176
output (Dense)	**(None, 4)**	**516**

Total params: 18,356
Trainable params: 18,196
Non-trainable params: 160

Network Architecture - High Level



Experimental Results- FMA Small Dataset

Network Architecture

- 3 Convolution Blocks (Conv2D + BN + MP + Dropout)
  - 16 , 32, 32 filters in each block respectively
- Permute and Reshape layers
- 1 RNN block - GRU with 8 units each
- 1 Dense layer with 128 units

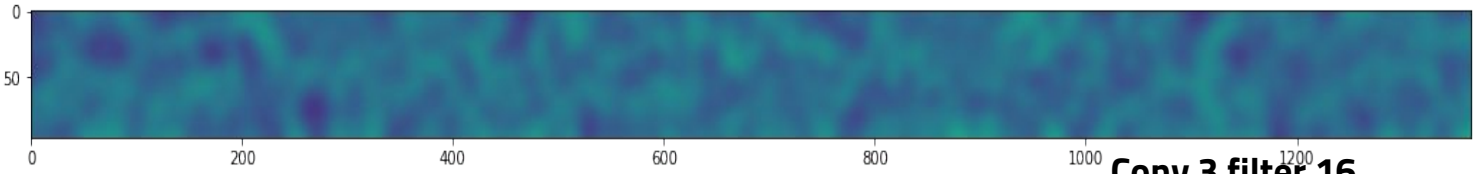
Parameters

- Trained for 20 epochs, Adam optimizer, lr = 0.0001
- loss = categorical cross entropy, batch size = 128

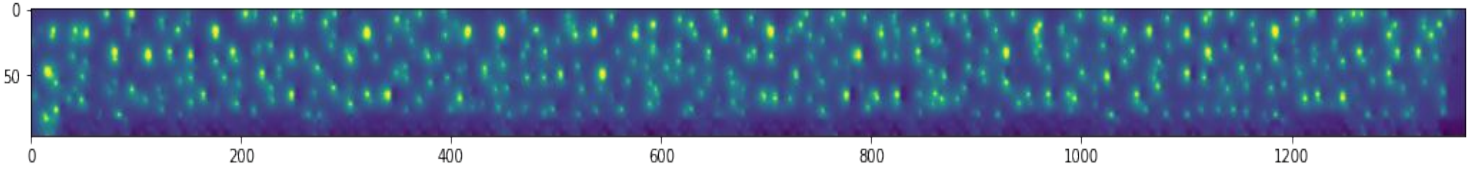
Test Accuracy of 50%

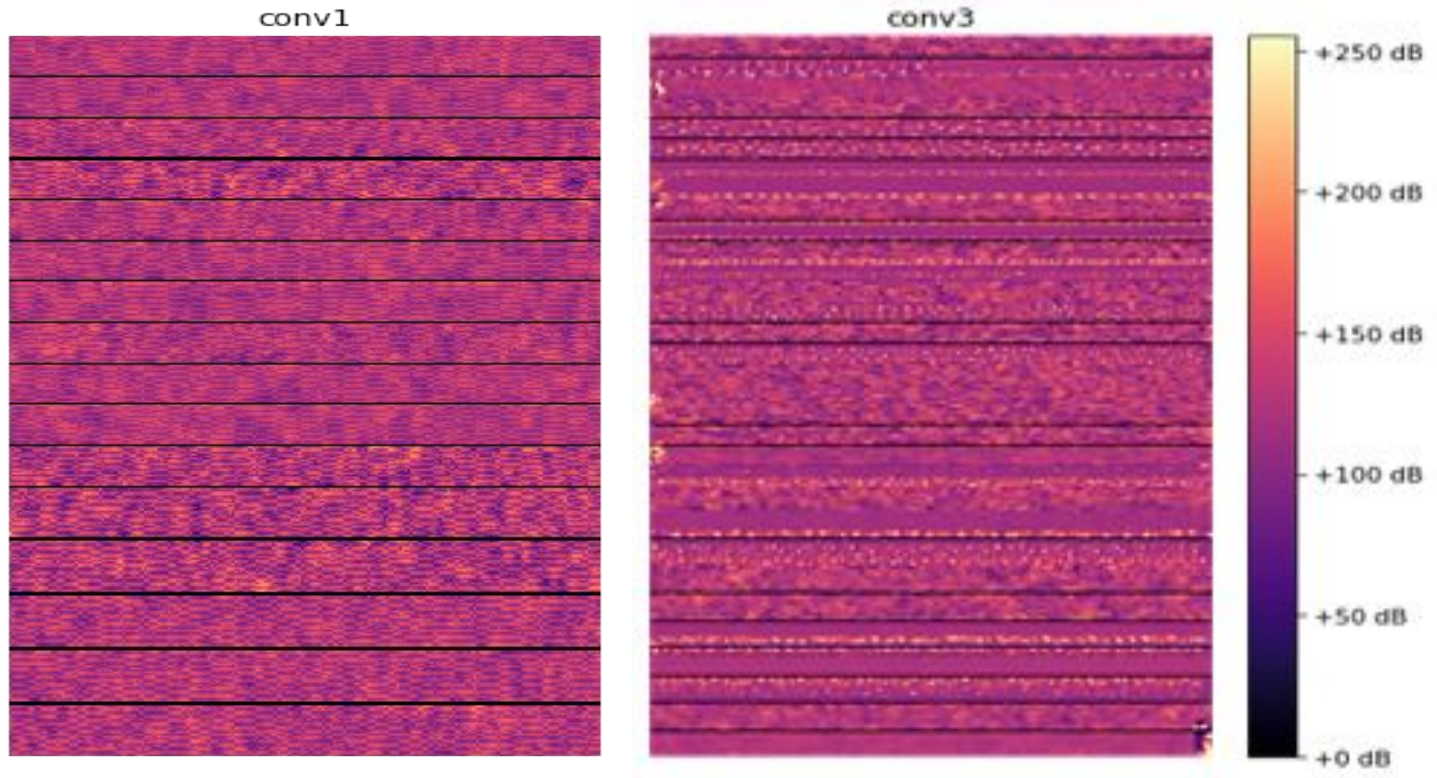
Visualizing

Conv1 Filter 1



Conv 3 filter 16

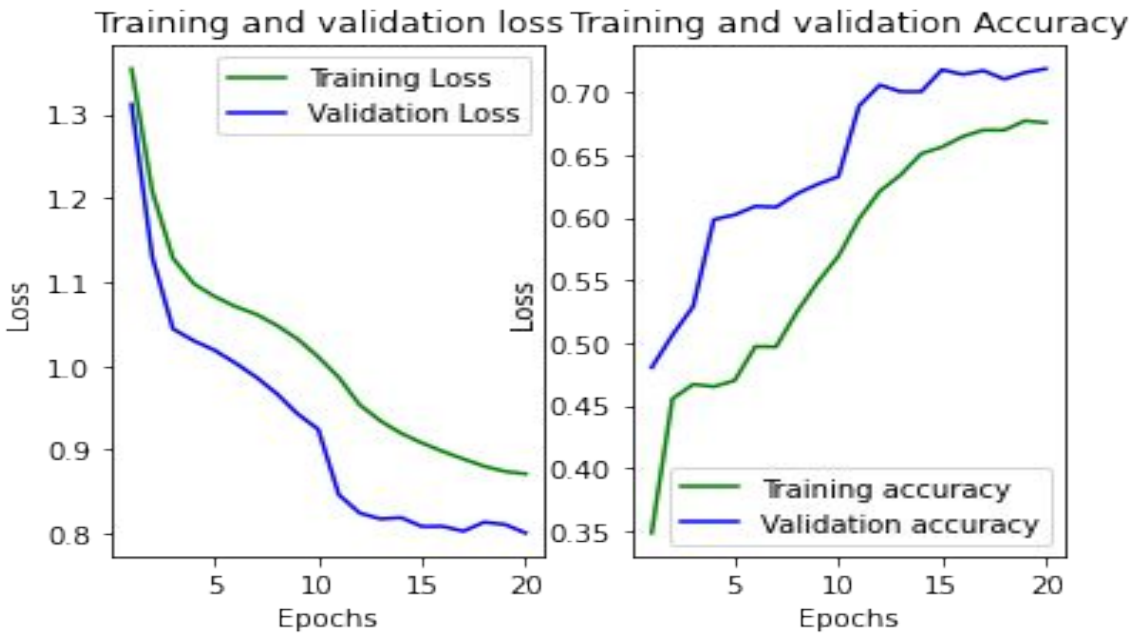


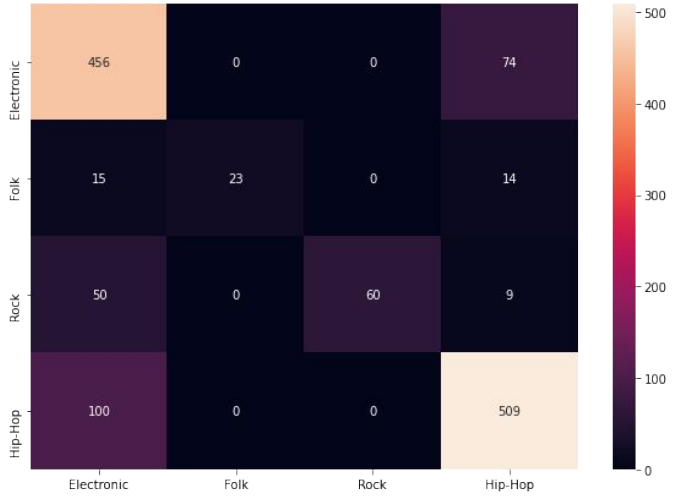


Experimental Results- FMA Medium Dataset

CNN Blocks	RNN Blocks	Dense Layer	Activation	Test Accuracy %
3 16 F (2,2) 32 F (2,2) 32 F (3,3)	1 - GRU 16	1 - 128 units	relu	**60**
3 16 F (2,2) 32 F (2,2) 64 F (3,3)	2 - GRU 8 units each	1 - 128 units	selu Alpha Dropout	**70**
3 16 F (2,2) 32 F (2,2) 64 F (3,3)	2 - GRU **16** units each	1 - 128 units	selu Alpha Dropout	**72**

Analyzing Results





Confusion Matrix

Conclusion

After examining several choices of datasets, pre-processing methods, neural network structures, and other factors, we found the optimal combination to be a convolutional recurrent neural network using mel-spectrograms of thirty seconds long samples of audio. A bigger dataset improves our final accuracy and f1 score but further tuning and class balancing techniques can further improve our results. Our final best test accuracy turned out to be 72%.

References

[1] Michal Defferrard, Kirell Benzi, Pierre Vandergheynst, Xavier Bresson. FMA: A Dataset For Music Analysis. arXiv:1612.01840

[2] Keras Library API Documentation - https://keras.io/

[3] Scikit Learn - https://scikit-learn.org/

[4] Tzanetakis, G. and Cook, P. (2002). "Musical genre classification of audio signals." IEEE Transactions on Speech and Audio Processing, 10(5), pp.293-302

[5] Ioffe, Sergey and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift." ICML (2015)

[6] https://github.com/derekahuang/Music-Classification/blob/master/CS229_Final_Report.pdf

[7] ybayle/awesome-deep-learning-music: List of articles related to deep learning applied to music