CNN based Music Instrument Classification Model

Anirudh Sriram (ee18b073), Arjun Menon V (ee18b104), Srinivas Mareddi(ee18b057), Nithin Varma (ee18b052)
Electrical Engineering, IIT Madras

Abstract—In this project we use CNN based models to classify instruments using the Freesound audio data set. Our proposed architecture computes the Mel-spectrogram from the input audio data and feeds it to a CNN based model. To add robustness to the model, we use a novel data augmentation technique based on the CutMix Algorithm. We optimise the architecture using Hyperparameter Tuning and Pruning, and analyse the model by generating the Class Activation Maps and by conducting an Ablation Study.

1 Introduction

In Music Instrument Identification, while the features may be extracted using classical DSP techniques, decision functions to map the features to instrument classes typically rely on learning based approaches. Some popular approaches use SVMs, Hidden Markov Models (HMM) and Neural Networks for the classification task. In our work, we extract the features from audio samples by computing the Mel Frequency Cepstral Coefficients, and train a Convolutional Neural Network to classify the features to the corresponding labels.

In addition to the proposed CNN architecture, we make the following additions- (1) Using **CutMix Algorithm** on the Mel Spectrograms to augment input data and add robustness to the model; (2) **Hyper Parameter Tuning** using keras tuner with Random search algorithm. (3) **Model Pruning** to obtain a lighter model with similar performance metrics; (4) **Class Activation Maps** to identify neurons that play a significant role in identifying a class; (5) **Ablation study** to understand the importance of individual layer from the proposed model architecture. We have also done a thorough comparative analysis of our model with traditionally used instrument classification models.

Music instrument identification can be used in building Content Based Recommender Systems, in Genre Identification tasks and can be extended to provide solutions for similar problems in other domains, for example in anomaly detection for mechanical systems.

2 Related Works

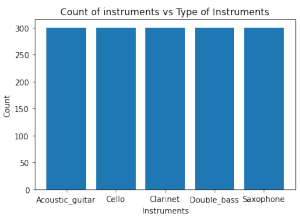
Prior work on music instrument identification tasks include SVM, HMM, Random Forest and CNN based models operating on feature vectors. Table 1 summarises the highlights of some of these works.

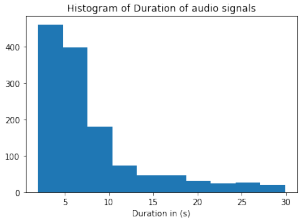
3 Dataset

We have taken an open source dataset called **'Free Sound Audio tagging data'** for classification of musical instruments. The unfiltered version of the dataset had 9400 unique audio samples stored in .wav format and there were 41 unique classes of instruments. We filter out the top 5 most commonly occurring classes to make a dataset with 1500 samples. The classes were **'Acoustic Guitar', 'Cello', 'Double Bass', 'Saxophone'** and **'Clarinet'.**

3.1 Preparation

The duration of the audio samples varies from 0.1 to 30s. Next we remove the samples with audio content less than 2s. This can be reasoned due to the fact that there will not be sufficient data for the model to learn from audio files. Next to make use of files with more than sufficient audio content we make use of random sampling algorithm to obtain random chunks of required threshold length from each data, and hence create 6000 more samples.

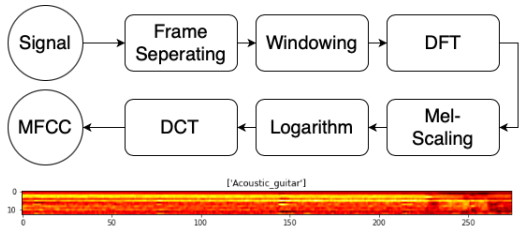




The Mel Frequency Cepstral Coefficients (MFCCs) of each music piece from the generated data is extracted using Librosa. For each audio file, its MFCCs are averaged to produce the final feature vector in which only 13 of the 26 DCT coefficients are kept. Now the shape of mfcc features represents the number of audio data and the heatmap image with the size of 275 times 13 produced using mfcc() function. The most time-consuming part of the system is the MFCC feature extraction. However, it still needs less than a second to process a one-second audio, which makes it feasible to do real-time instrument detection. Next we label-encode by one-hot-encoding the labels of each sample.

Paper Title	Models Proposed	Datasets	Highlights
Instrument classification using Hidden Markov Models.	Hidden Markov Models	Synthesized datasets	Novel Data creation method, Power spectrum coefficients as input features
Musical Instrument Classification	SVM, Random Forest classifier, Dense NN	Google Audio set	Comparative study among multiple ML classifiers
Predominant Musical Instrument Classification based on Spectral Features	SVM	IRMAS (Instrument recognition in Musical Audio Signals) - polyphonic data	Hierarchical Clustering methods
Musical Instruments Classification using Pre-Trained Model	SVM, KNN	RWC database, MINIM-UK database, IRMAS dataset	Usage of AlexNet to obtain the input features
Musical Instrument Classification Using Neural Networks	Probabilistic Neural Networks (PNN)	Experimental dataset: 4548 solo tones from 19 instruments of MIS database	Usage of MFCC as input features. Usage of both Hierarchical and Non-hierarchical classification methods
Learning Instrument Identification	SVM- various kernels, PCA, K Means	Experimental dataset: 1,455 samples, obtained from both online and live recordings	Propose a set of features that can be used to accurately classify musical instruments. Usage of supervised and unsupervised learning algorithms
Music instrument recognition using deep convolutional neural networks	Deep CNN	IRMAS	Usage of Techniques to avoid overfitting. Usage of MFCC features.
Music and Instrument Classification using Deep Learning Technics	CNN	AudioSet By Google	Usage of MFCC as feature inputs

Table 1: Summary of Prior Work on Music Instrument Classification

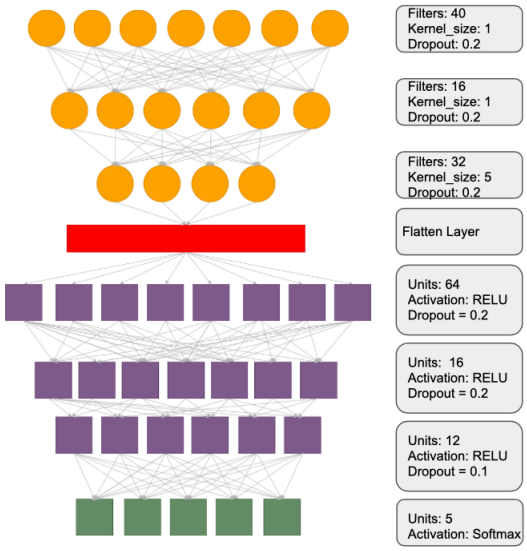


Schematic Representation of Preprocessing steps in computing the MFCC coefficients

4 Model

4.1 CNN Model Architecture

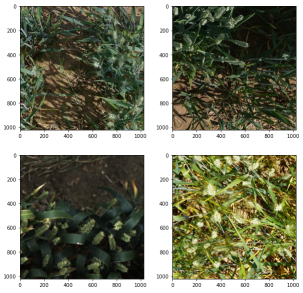
The proposed model comprises of 3 convolutional layers and 3 dense layers, taking in inputs of size 275 x 13 and generating 5 x 1 softmax predictions corresponding to the 5 classes. To prevent overfitting, the convolutional layers incorporate dropout with p= 0.2. Model checkpoints are saved and the weights corresponding to maximum validation accuracy obtained during training are used. All the layers use RELU activation and Adam optimizer with an initial learning rate of 0.001.



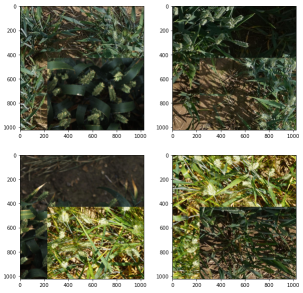
Model Summary: Circular Nodes correspond to Convolutional Layers while Rectangular Nodes correspond to Dense Layers. Parameters obtained after hyperparameter tuning using the Keras Tuner API

4.2 CutMix Algorithm

CutMix is an data augmentation technique predominantly used in computer vision tasks, were patches are cut and pasted among training samples and corresponding ground truth labels are also mixed proportionally to the area of the patches. By making efficient use of training data and retaining the regularization effect of regional dropout, CutMix consistently outperformed the state-of-the-art augmentation strategies.



a) Original Images from Training set



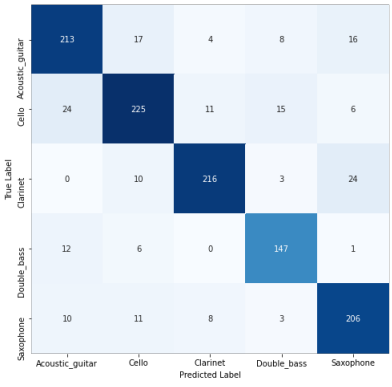
b) Augmented Image after CutMix

4.2.1 Implementation of CutMix Algorithm

[1] The preprocessed data represented as a Mel-Spectrogram has a dimension of (275,13), and contains spatial information which can be extracted using convolutional layers. Cut Mix increases localization ability by making the model to focus on less discriminative parts of the object being classified. The idea from mixing training pixels to make the models more robust is used with the input audio features in our case. The mixing of portions of training data also makes the model to be aware of handling reallife like audio feature data with multiple instruments.

4.2.2 Inferences from output of CutMix Algorithm

The following figure describes the confusion matrix obtained using the CNN + CutMix model on the validation dataset. The class-wise accuracies obtained for this were **85%** for **Acoustic Guitar**, **80%** for **Cello**, **85%** for **Clarinet**, **89%** for **Double Bass** and **87%** for **Saxophone**.



Confusion Matrix obtained on the validation set

It can be noted from the confusion matrix that instruments that are similar, as perceived by the human ear, are likely to be misclassified by the model; for example, a considerable number of misclassifications occur between the classes Guitar and Cello. In contrast, instruments that are dissimilar in characteristics seldom get misclassified. This result validates the choice of MFCCs as the feature fed to the CNN, as it effectively extracts the features that the human ear is sensitive to, and is also evidence that the CNN model is effective at learning the mapping from the features to the instrument classes.

4.3 Hyper-parameter Tuning

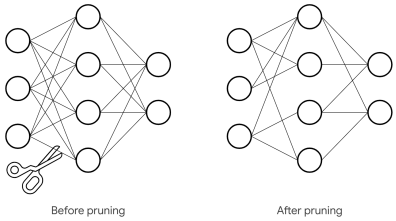
Optimal hyperparameters for the model were chosen using the Keras Tuner API. Hyperparameters that were tuned were the number of filters and kernel size for the convolution layers, and the number of units in Dense layers and the learning rate of the Adam optimizer. Since there were 1.4 million+ points in the search space, we used the Random Search algorithm. In contrast to GridSearch, which iterates over all possible points in the search space, Random Search goes through only a fixed number of hyperparameter settings in a random fashion, and guarantees faster convergence to the optimal set of hyperparameters.

Hyperparameter	Parameter Space	Optimal Parameter
Conv_1_filter	16 to 64 ; step_size=8	40
Convo_1_kernels	[1,3,5]	1
Conv_2_filter	16 to 32 ; step_size=4	32
Conv_2_kernel	[1,3,5]	5
Conv_3_filter	8 to 16 ; step_size=2	16
Convo_3_kernels	[1,3,5]	1
Dense_1_units	16 to 64 ; step_size=8	64
Dense_2_units	8 to 16 ; step_size=4	16
Dense_3_units	4 to 8 ; step_size=2	12
Learning Rate	[1,0.1,0.01,0.001]	0.001

Table 2: Parameters of Neural Layers

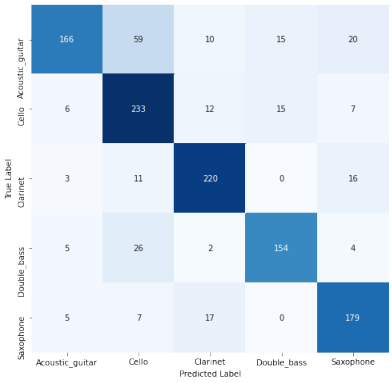
4.4 Model Pruning

One of the main applications of Music separation is real time music recognition in our mobile devices. Hence a light weight and efficient model is very essential for this. Pruning is used to reduce the number of parameters and operations involved in the computation by removing connections, and thus parameters, in between neural network layers. We start off with a 50% sparse neural network and keep increasing the sparsity till we reach the minimum performance threshold.In this work,by pruning our model we bring down the number of trainable parameters from 3.6 million to 1.8 million with the accuracy of the model decreasing only slightly from 86% to 83%



Representation of Pruning

The below figure represents the confusion matrix obtained after pruning the CNN Model and using it on the validation set.



Confusion matrix after Pruning

If we consider the two instruments, Guitar and Cello, we can observe that accuracy of Cello has slightly increased. Also misclassifications of cello as guitar have decreased and guitar as cello have increased. From this we can propose that a Cello is spectrally similar but less complicated than a Guitar. So after pruning more Guitar audio files have been classified as Cello and lesser Cello audio files have been classified as a Guitar. Similar conclusions can be drawn for other instruments.

5 Analysis of the Neural Network

5.1 Comparison with Other Models

We have compared our proposed model with other popular and commonly used models as mentioned from the section on related works. The algorithms used are Naive Bayes,Decision Tree classifier, SVM, Random Forest, Adaboost, MLP, Logistic Regression etc.The below table shows us the accuracy, precision, recall, F1 Score metrics for these classification models.

Model Name	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.68	0.68	0.68	0.68
Naive Bayes	0.58	0.62	0.6	0.58
SVM	0.71	0.71	0.71	0.71
Decision Tree Classifier	0.64	0.64	0.63	0.64
Random Forest	0.80	0.81	0.80	0.81
Adaboost	0.66	0.68	0.67	0.66
MLP Classifier	0.47	0.48	0.47	0.47
CNN (w/o CutMix)	0.79	0.79	0.80	0.79
CNN + CutMix	**0.86**	**0.86**	**0.85**	**0.85**
CNN + CutMix + Pruning	**0.83**	**0.83**	**0.82**	**0.82**

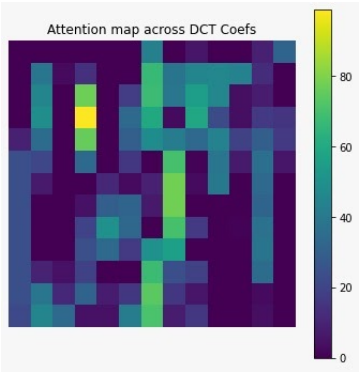
Table 3: Table of Comparisons

Insights

1) We can observe that the Random Forest Classifier is the best performing Machine learning algorithm with a pretty good accuracy of 80%.
2) The accuracy of our CNN Model has increased from 79% to 86% after implementation of Cut-Mix algorithm.
3) The accuracy obtained after pruning has only decreased slightly to 83% despite halving the number of parameters.

5.2 Class Activation Maps

Class Activation Maps help us visualise the significance of each individual features at the input to the model. We make use of Grad-CAM, which creates heatmaps by visualising the important of input features from the output of any convolutional layer in the network. The gradient of the output class value with regards to each channel in the feature map of a certain layer is calculated. This results in a gradient channel map with each channel in it representing the gradient of the corresponding channel in a feature map. The gradient channel map obtained is then global average pooled, and the values obtained henceforth are the weights of importance of each channel in the feature map. The weighted feature map is then used as a heat map.



A sub matrix of the attention map across the DCT coefficients

5.3 Ablation Study

To have have a better understanding of our model we do an ablation study to see how significant of a role different layers play in predicting the classes accurately.To estimate this

qualitatively, we remove some intermediate layers from the model and train the network and observe the corresponding accuracy.

Ablation study	Accuracy
No Conv Layer + 3 Dense Layers	0.4698
1 Conv Layers + 3 Dense layers	0.5727
2 Conv Layers + 3 Dense layers	0.8277
3 Conv Layers + 3 Dense Layer	0.8599
3 Conv Layers + No Dense Layer	0.8068
3 Conv Layers + 1 Dense Layer	0.8176
3 Conv Layers + 2 Dense Layers	0.8319

Table 4: Ablation Study Results

Insights

1) Removing convolutional layers takes a big hit on the accuracy of the model.This is as expected- conv layers identify the spatial information in the MFCC matrix.
2) Dense layer predominantly learn the approximation of the function mapping output of convolutional layers to softmax output to get the defined classes.
3) Improvement in model accuracys is much higher with addition of a convolution layer compared to the addition of Dense layers.

Note: We have added 3 dense layers instead of 2 or 1 to keep the accuracy of the model above **86%**, so that after pruning when the accuracy falls to **83%**, its still better than that of a Random Forest Classifier (**80%**) as seen in Table 2.

6 Conclusion and Future Work

From the thorough analysis of our proposed model, we identify the significance of both CNN and Dense layers for efficient prediction. Also, we observe that usage of CutMix algorithm helps us to gain a 7% improvement in accuracy. In the future, we can try and explore working with audio files with multiple instruments and also try out more sophisticated models like LSTM or Transformers based models for better model performance.

References

[1] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe and Youngjoon Yoo, *CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features*,Yonsei University, 2019
[2] Lara Haider Ahmad, *Music and Instrument Classification using Deep Learning Technics*, Stanford University, 2019
[3] Tim Woodford, Jared Pham and Jonathan Lam, *Musical Instrument Classification*, UC San Diego, 2019
[4] Karthikeya Racharla, Vineet Kumar, Chaudhari Bhushan Jayant, Ankit Khairkar and Paturu Harish, *Predominant Musical Instrument Classification based on Spectral Features*, Indian Institute of Technology, Kharagpur, 2020
[5] *Learning Instrument Identification*, Stanford University, 2015
[6] *Musical Instrument Classification*, UCSD, 2015