Music Instruments Classification Using Signal Processing and Machine Learning

Vivek Mittal
School of Computing and Electrical Engineering
Indian Institute of Technology, Mandi
b18153@students.iitmandi.ac.in

***Abstract*—The goal for this project is to develop a machine learning model for string instruments classification. We use various signal processing techniques to calculate useful features from instruments sound recordings. These calculated features are then used for training a support vector machine classifier. We achieved 75% accuracy on a subset of IRMAS dataset.**

***Index Terms*—Signal Processing, Instruments Classification, Machine Learning**

I. Introduction

With the increase of on-demand music it is very important to have better classification systems that can serve as a backbone of good recommendation systems so in this project, we work on the problem of instrument music classification. We approach the classification problem in a data driven way. We focus on signal processing technique mainly on the Mel-frequency Cepstrum (MFCC) features for classification. We use the support vector machines (SVM) algorithm for doing the classification.

Currently deep learning methods for music classification using convolutional and graphical neural networks [1]–[3] are on the peak in terms of accuracy but they require a lot of data to train. In this work, we emphasis on the signal processing techniques to calculate meaningful features which can be used for classification with a lightweight machine learning algorithm like SVM.

Now, we discuss about some of the earlier work done by the community. *Essid et al* [4] presented MFCC based method for classification of wind instruments, they used GMM and SVM for classification purpose. *Deng et al.* [5] studied the properties of various instruments described by their temporal and spectral features, they found only first five - seven coefficients are enough for good classification results. *Eronen et al.* [6] used a lot of features covering temporal as well as spectral information of a sound. They designed extraction algorithms for extracting useful temporal and spectral features. Recent papers treats the MFCC as an image and apply convolutional neural networks on them achieving state-of-the-art on various benchmarks.

In section II, we provide details of the method used for feature extraction and classification. In section III, we provide a description about your experiments.

II. Methodology

For implementing our classification algorithm, we first need to extract some useful features from the given instrument sound recording. Audio signals are continuously changing, so to simplify things we assume that signal is not changing too much in between small frames usually 20-40ms. We divide the whole signal in overlapping frames with each frame containing N samples. *Frame Blocking* is done such that the first frame contains samples from 0 to $N-1$, the next frame from M to $M+N-1$ and so on until every sample in the signal have entered into one or more frame. The next step in processing the signal is *Windowing*, we window each frame using hanning window, equation (1). Windowing minimizes the signal discontinuities at the start and end of each frame. Now our frames are ready to be processed using fourier transform for calculating the spectral features. We use fast fourier transform (FFT) which is a fast algorithm to implement *Discrete Fourier Transform*. In all the equations, i refers to the frame index, k refers to the samples in a frame therefore k goes from 0 to $N-1$.

$$w[n] = 0.5(1 - cos(2\pi n/N)), \quad n = 0, \ldots, N. \tag{1}$$

$$X_i[k] = \sum_{n=0}^{N-1} x_n e^{-2\pi i k n/N}, \quad k = 0, \ldots, N-1. \tag{2}$$

where $x_n = w[n]x[n]$, $x[n]$ is the given discrete sequence from 0 to $N-1$ and i indexes over the frames.

We extract temporal as well as spectral features. To be specific we extract Zero Crossing Rate, Root Mean Square Energy, Spectral Centroid, Spectral Roll-off, Spectral Bandwidth, Mel-frequency cepstral coefficients (MFCC).

- **Zero Crossing Rate:** It is the rate at which the signal crosses zero.
- **Root Mean Square Energy:** This is the root mean square energy of a given frame.
- **Spectral Centroid (SC):** This defines the centroid of the frequency spectrum. SC indicates the frequency at which energy of the spectrum is concentrated.

$$f_{c,i} = \frac{\sum_{k=0}^{N-1} X_i[k] f_i[k]}{\sum_{k=0}^{N-1} X_i[k]} \tag{3}$$

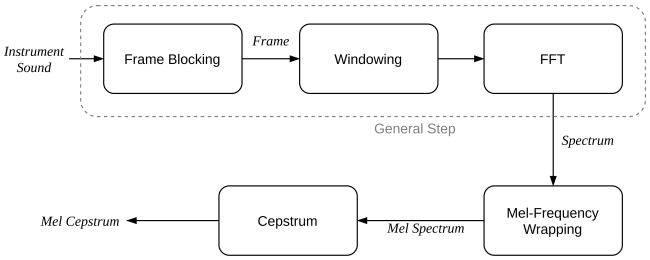


Fig. 1. Block Diagram of MFCC Calculations.

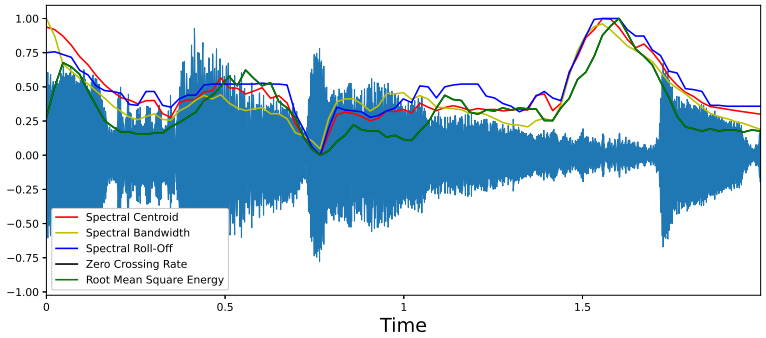


Fig. 2. Features and waveform of a sample.

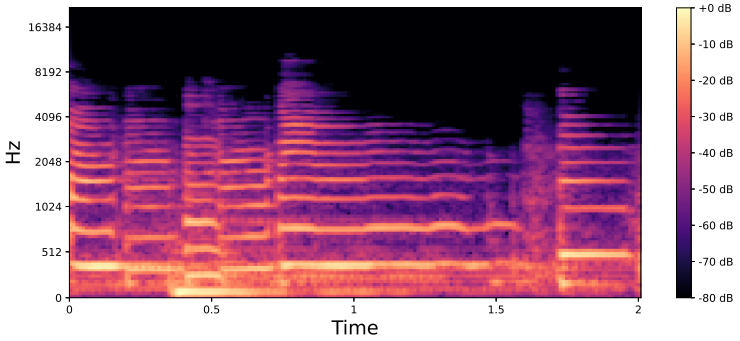


Fig. 3. Mel Scale Frequency spectrogram.

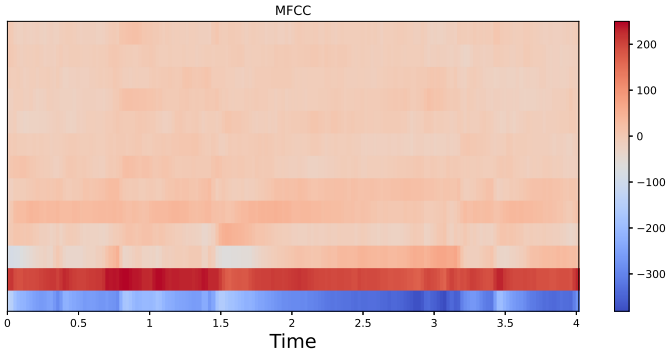


Fig. 4. MFCC for the above waveform.

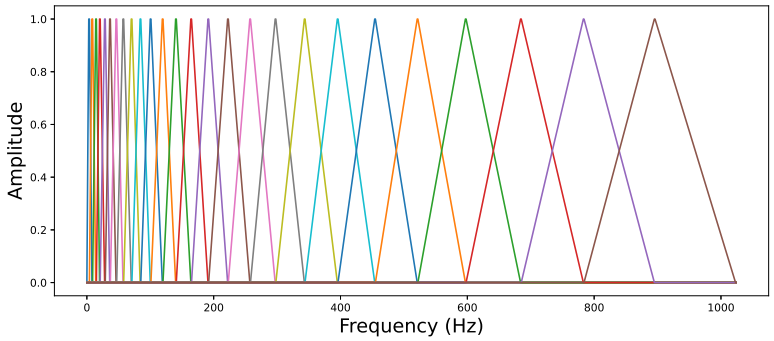


Fig. 5. Mel Filter Bank.

where $f_i[k]$ is the frequency at the kth bin in frame i.

- **Spectral Roll-off:** Frequency below which a certain proportion of spectral energy lies. We can also define the spectral roll off as the pth percentile of spectral power distribution. Here we considered the frequency below which 85% of the energy lies.
- **Spectral Bandwidth (SBW):** This is the weighted frequency deviation from the centroid frequency (SC).

$$SBW_i = (\sum_k (X_i[k](f_i[k] - f_{c,i}))^2)^{1/2} \tag{4}$$

- **Mel-frequency cepstral coefficients (MFCC):** To model the perception of the human ear to incoming sound usually Mel-scale is followed. The mel scale is linear for low frequency (below 1000Hz) and is logarithmic for high frequency (above 1000 Hz).

$$F_{mel} = \begin{cases} F_{Hz} & F_{Hz} \leq 1000Hz \\ 2595 \log_{10}(1 + \frac{F_{Hz}}{700}) & F_{Hz} > 1000Hz \end{cases} \tag{5}$$

where F_{mel} is the mel-frequency for the frequency F_{Hz}. For calculating the MFCC coefficients we first need the power spectral estimate for every frame which is given by

$$P_i[k] = \frac{1}{N}|X_i[k]|^2 \tag{6}$$

Next, we convert our lower and upper frequency in mel scale using equation (5). After that, we compute 26 equally spaced point in between the upper and lower frequency, so now we have 26+2 mel scale frequency points. Now, convert these mel scale back to frequency scale in Hz again using equation (5) and then round off these frequencies to the nearest FFT bin, let's call this sequence of frequencies as $g[m]$ where m ranges from 0 to 27. Next, we create our triangular shaped filter banks. The first filter bank starts at $g[0]$ reaches it's maximum at $g[1]$ and decreases to zero at $g[2]$, using similar procedure we get 26 triangular shaped filter banks as shown with equation (7). In equation (7) k is the frequency in Hz.

$$H_m(k) = \begin{cases} 0 & k < g[m-1] \\ \dfrac{k - g[m-1]}{g[m] - g[m-1]} & g[m-1] \leq k \leq g[m] \\ \dfrac{g[m+1] - k}{g[m+1] - g[m]} & g[m-1] \leq k \leq g[m] \\ 0 & k > g[m+1] \end{cases} \tag{7}$$

We multiply each filter from the filter bank with the power spectrum and sum up the multiplication to get the energy value of that filter. After multiplying all 26 filter banks, we have 26 energy values for every frame. We take log of these 26 energy values to get the log filterbank energies. At last, we take Discrete Cosine Transform (DCT) of these 26 log filterbank energies which give us 26 cepstral coefficients, we only keep the first 13 coefficients.

For classification we used SVM, which is kernel based classification technique that fits a best separating hyperplane in between the classes. We used 'radial basis function' kernel for our project. The value of parameters like 'C' and 'gamma' were finetuned using using a holdout validation set from the training set.

III. Experiments

In this section, we provide details of the experiments done for this project.

A. Dataset

For this project we use IRMAS (Instrument recognition in Musical Audio Signals) dataset. This dataset comes in two part, for this project we only use the training part from which we construct our own testing data. This dataset consists of 3 sec .wav files for eleven type of instruments, in total we have 6705 audio files in 16-bit stereo format sampled at 44.1kHz. The data consists of various genres like country, rock, latin, etc. We only used five instruments for our classification purpose those are flute, acoustic guitar, organ, piano and trumpet.

Instrument	Number of Samples
Cello	388
Clarinet	505
Flute	451
Acoustic Guitar	637
Electric Guitar	760
Organ	682
Piano	721
Saxophone	626
Trumpet	577
Violin	580
Human Singing Voice	778

TABLE I
INSTRUMENTS IN IRMAS DATASET

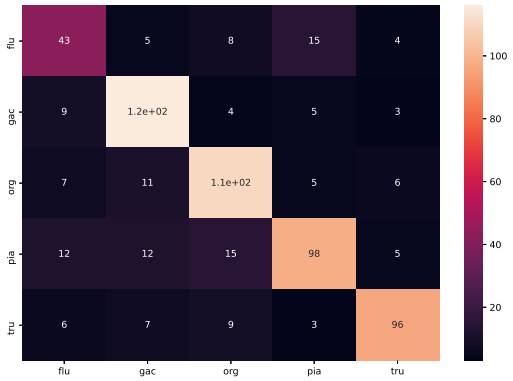
B. Implementation Details

We crop the first 0.5 sec of the audio signal to reduce the noise in the extracted features and limit the length of 2 sec for audio signals. For all the features we take the mean of all the frames to get a feature value representing the whole signal. We obtain 18 features for every audio sample. We split the data in 80:20 for training and testing respectively. We use frame length of 2048 samples and the hop length of 512 samples, hanning window is used for windowing, zero padding is used where ever needed. All of the feature extraction is done using librosa library, the SVM algorithm is implemented using sklearn library, and the code is implemented in Python.

C. Evaluation Metrics

- **Accuracy** is ratio of number of correctly predicted samples to the number of total samples.
- **Precision** is the ratio $\frac{tp}{(tp+fp)}$, where tp is number of true positive samples and fp is number of false positive samples.
- **Recall** is the ratio of $\frac{tp}{(tp+fn)}$, where tp is number of true positive samples and fn is number of false negatives samples.
- **F1-Score** is the harmonic mean of precision and recall.

D. Results



Instrument	Precision	Recall	F1-Score
Flute	0.67	0.53	0.59
Acoustic Guitar	0.71	0.78	0.74
Organ	0.79	0.82	0.80
Piano	0.72	0.80	0.76
Trumpet	0.80	0.72	0.76

TABLE II
RESULTS ON TESTING SET

In table II, we provide the summary of results on the testing set, we observe classifying flute is difficult than classifying other instruments, it could be because of less number of samples for flute. It is also observed organ and flute are getting mixed up more during classification the could be because of the nature of these instruments, both are wind instruments. The accuracy of our classification algorithm on the test set is 75%. We also tried classification without any MFCC feature then the classification accuracy dropped to 55% (keeping all other parameters same), this suggests that the MFCC features are very important for classification.

IV. CONCLUSION

In this project, we designed a instruments classification system. We used spectral as well as temporal features for making the decision in between the classes. We used SVM for separating the samples, using this we achieved 75% on a subset of IRMAS dataset. We also found that Mel-frequency cepstral coefficients are the main and the most important features for our classification algorithm.

REFERENCES

[1] M. Defferrard, S. P. Mohanty, S. F. Carroll, and M. Salathé, "Learning to recognize musical genre from audio," *arXiv preprint arXiv:1803.05337*, 2018.

[2] L. Franceschi, M. Niepert, M. Pontil, and X. He, "Learning discrete structures for graph neural networks," *arXiv preprint arXiv:1903.11960*, 2019.

[3] C. Liu, L. Feng, G. Liu, H. Wang, and S. Liu, "Bottom-up broadcast neural network for music genre classification," *Multimedia Tools and Applications*, pp. 1–19, 2020.

[4] S. Essid, G. Richard, and B. David, "Musical instrument recognition on solo performances," in *2004 12th European signal processing conference*, pp. 1289–1292, IEEE, 2004.

[5] J. D. Deng, C. Simmermacher, and S. Cranefield, "A study on feature analysis for musical instrument classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 2, pp. 429–438, 2008.

[6] A. Eronen and A. Klapuri, "Musical instrument recognition using cepstral coefficients and temporal features," in *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100)*, vol. 2, pp. II753–II756, IEEE, 2000.