ISSN No. 0976-5697

Volume 8, No. 5, May – June 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

SVM Based Classification of Sounds from Musical Instruments using MFCC Features

Sayantani Nandi, Madhura Banerjee, Parangama Sinha, Jayati Ghosh Dastidar*
Department of Computer Science
St. Xavier's College, Kolkata (Autonomous)
India

Abstract: This paper aims at classifying sounds obtained from different musical instruments. The proposed methodology works by extracting the Mel Frequency Cepstral Coefficients (MFCC) of a given sound signal. The extracted features are considered to be vectors input to a Support Vector Machine (SVM). The SVM classifies the MFCC feature vector of the given sound signal by using a Minimum Distance Classifier (MDC) based classification scheme which operates by calculating the Euclidean distances of the given vector from the representative pattern vectors of the different pattern classes that the SVM has been trained with. The given sound signal is then identified as being a member of the class for which the Euclidean distance is the minimum; and is thus classified.

Keywords: MFCC; pattern vectors; pattern recognition; Support Vector Machine; Minimum Distance Classifier; sound classification

I. INTRODUCTION

As a sub-field of computational linguistics, the field of sound/voice recognition has undergone tremendous research through the ages. It was first implemented as single-speaker digit recognition systems developed by Bell Labs (*The Bell Labs record-The Vocoder*). The two main aspects of the work involve audio content and speaker/instrument identity. In this paper we propose a scheme that takes audio inputs which are strictly musical in nature, and identifies which musical instrument the sound belongs to. The objective is to design a system that will make the recognition of musical instruments fast and easy.

Like all classification schemes, the success of our method too depends upon the creation and use of a quality training data set. To this end we create a database containing multiple sound samples from different musical instruments such as guitar, violin, flute, piano, etc. After subjecting these samples to a few pre-processing routines such as removal of noise, trimming them so as to remove the static contents, etc. we compute the Mel Frequency Cepstral Coefficients (MFCC) for them. The coefficients thus obtained for a sound sample is a training pattern vector representing the sample. Several such vectors for a particular instrument form a pattern class. These pattern classes would constitute the training data set that would be used by the Support Vector Machine (SVM).

Once the database has been created, a sound sample from an unknown source is then classified by first subjecting it to pre-processing methods of the likes mentioned above and then computing the MFCC for the same. The final step involves the computation of the Euclidean distance of this input pattern vector (consisting of the MFCC of the sound sample) from each of the pattern classes in the training database. The input sound is classified as belonging to that musical instrument from which the Euclidean distance is the minimum.

II. LITERATURE SURVEY

An exhaustive review of automatic classification of sounds from musical instruments was studied in 2003 [1]. Two different but complementary approaches were examined, the perceptual approach and the taxonomic approach. The former is targeted to derive perceptual similarity functions in order to use them for timbre clustering and for searching and retrieving sounds by timbral similarity. The latter is targeted to derive indexes for labeling sounds after culture- or user-biased taxonomies. A neutrally inspired musical instrument classification scheme was suggested by the authors in [2]. The classification scheme proposed uses a time-domain neural network – the echo state network. The authors in [3] have discussed the Multidimensional Gauss, KNN and LVQ algorithms in their paper. They have further suggested improvements by introducing an efficient process for Gradual Elimination of Descriptors using Discriminant Analysis (GDE) which improves a previous descriptor selection algorithm. An integral step during the computation of MFCC is the application of Discrete Fourier Transformation (DFT). The same has been discussed in details by the authors in [4]. Each of the sub-processes required for MFCC and the method to implement them has been described in [5]. The vulnerability of MFCC to accents and noise was pointed out in [6]. The use of MFCC to form the classification vector was first suggested by the author in [7].

III. DESIGN METHODOLOGY

The proposed system works through three broad phases. The first phase works deals with the acceptance of an input sound file and subjecting it to certain pre-processing operations. The pre-processing operations involve removal of noise and static from the signal. The second phase deals with the extraction of features from the processed sound signal. These features are supposed to be distinctive features of the sound signal. The third and final phase deals with the use of the extracted features (from phase 2) for the purpose of recognition and classification. This conceptual view of the phases has been depicted in Fig. 1.

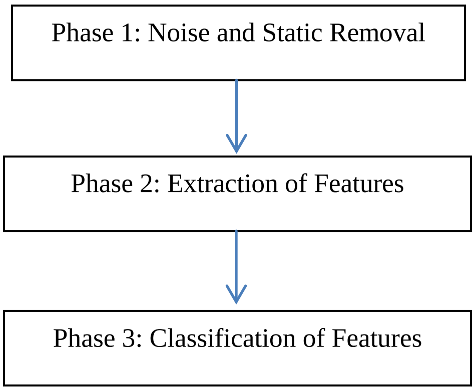
Phase 1: Noise and Static Removal
Phase 2: Extraction of Features
Phase 3: Classification of Features

Figure 1: Conceptual View of the entire process

A. *Phase 1: Noise and Static Removal*

The first phase begins by accepting a sound file and then smoothing out the noises in the sample. For this we chose to apply the Butterworth Low Pass Noise Removal Filter of order 1 [8]. This filter did not exhibit any ringing effect. A Butterworth low pass filter of order 2 displayed negligible ringing effect. However, higher order Butterworth low pass filter was rendered useless due to marked ringing effect. Hence the choice of a first order Butterworth low pass filter was justified. Once a smoothed out sound signal was obtained, the next step was to remove the presence of static from it. The expression used for the transfer function, H (u) of the filter is shown as in (1). Here, ω_u is the angular frequency at the point u, ω_c is the cut-off angular frequency and n is the order of the filter.

$$H\,(u) = \frac{1}{1 + \left({}^{\omega_u}/_{\omega_c}\right)^{2n}} \qquad (1)$$

Static or white-noise with respect to an audio signal is any signal that has a similar hissing sound. Technically, a random signal is considered as "white noise" if it is observed to have a flat spectrum over the range of frequencies that is relevant to the context. For an audio signal, for example, the relevant range is the band of audible sound frequencies, between 20 and 20,000 Hz. Such a signal is heard as a hissing sound. This static was dealt with in the following manner:

The entire audio signal was divided into several frames. In this case, the frame length was considered to be 0.05 times the sampling rate. Accordingly, total number of frames was also calculated depending on the frame length. Successively, each frame was analyzed in a loop and the maximum amplitude of that frame was determined. If the max amplitude of the frame was less than 0.03, then that particular frame was considered static and hence was discarded. A trimmed and noise-free signal is thus obtained in this phase.

B. *Phase 2: Extraction of Features*

The signal obtained after the pre-processing operation is now used to extract features. The feature extraction is usually a non-invertible (lossy) transformation. For this we calculate the Mel Frequency Cepstral Coefficients (MFCC) of the signal. Each step in the process of MFCC is motivated by perceptual or computational considerations [9]. The following steps are related to the feature extraction process [10]:

a) *Take waveform and Convert to Frames*

b) *Compute the Discrete Fourier Transformation*

c) *Compute the Log of Amplitude Spectrum*

d) *Do Mel scaling and smoothing*

e) *Apply Discrete cosine transform*

The signal is sampled by considering frames in the waveform. The frames in time domain are then transformed into frequency domain by applying the Discrete Fourier Transformation (DFT). The Amplitude Spectrum of the obtained DFT is considered and passed through a Mel filter bank. The output obtained from the filter is known as Mel spectrum. The logarithm operation is then applied on the obtained Mel spectrum. Mel Frequency Cepstral Coefficients (MFCC) is finally obtained by applying the Discrete Cosine Transformation (DCT) to the log of Mel Spectrum [11]. The feature extraction process using MFCC has been shown in Fig. 2.

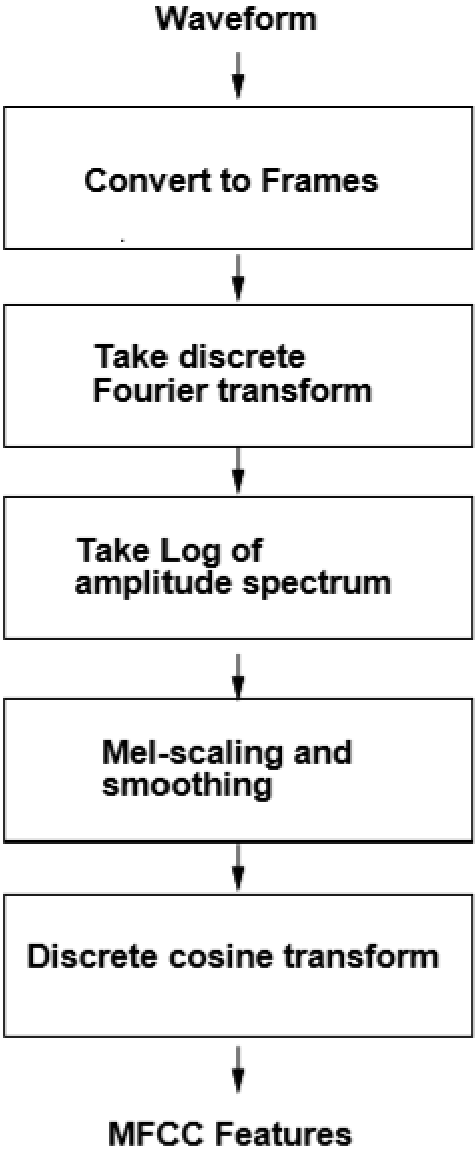
Waveform
Convert to Frames
Take discrete Fourier transform
Take Log of amplitude spectrum
Mel-scaling and smoothing
Discrete cosine transform
MFCC Features

Figure 2: MFCC steps [11]

C. *Phase 2: Classification of Features*

Like any other pattern recognition systems, audio recognition systems also involve two phases namely, training and testing. Training is the process of familiarizing the system with the characteristics of the different musical instruments. Testing is the process of matching and recognizing input audio signals with audio signals stored in the database during the training phase. Fig. 3 depicts the training phase.

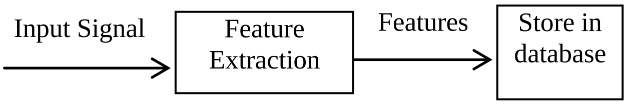

Figure 3: Training Phase

Fig. 4 shows the Testing Phase wherein the signal to be classified is taken, its features extracted, matched with the references stored in the database and finally a decision is taken to determine whether the input signal belongs to one of known musical instruments or otherwise.

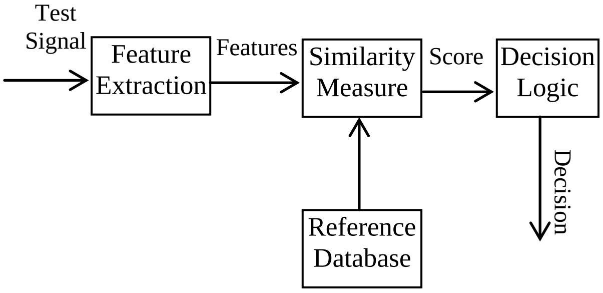

Figure 4: Testing Phase

The success of the Testing Phase depends upon the choice of a precise *Similarity Measurement* routine along with a deterministic *Decision making Logic*. We have designed the Testing Phase along the lines of a Support Vector Machine (SVM). Using a supervised learning model like SVM, aids in the classification of an unknown pattern into an appropriate pattern class [12]. In the proposed system, separate pattern classes have been defined for the different musical instruments. Each class contains multiple vectors from the same musical instrument. The attributes of the vectors are the MFCC coefficients for the sound samples. Our SVM works as follows [8]:

a) All pattern vectors of each class are extracted from the database and stored in respective variables.

b) The mean vector for each class is calculated.

c) The decision making function that has been used by us is a Minimum Distance Classifier (MDC). This function makes use of the Euclidean distance to arrive at the classification decision. The Euclidean distance is found for each pair: mean vector of ith class, m_i and unknown pattern vector, x. It is given by $||x - m_i||$. Given two vectorn A and B containing n attributes each, the Euclidean distance between A and B is calculated using (2),

$$||A - B|| = \sqrt{\sum_{i=1}^{n} (A_i - B_i)^2} \quad (2)$$

where A_i and B_i are the $i^{th.}$ attributes of the vectors A and B respectively. In our case the attributes would be the MFCC coefficients which have been computed previously.

d) The unknown pattern belongs to the class for which the distance is the least.

This method of classification is known as the Minimum Distance Classification scheme. Lesser the value of the Euclidean Distance of the input signal from a class, greater is the chance of the signal belonging to the class; as the resemblance of the signal with the class is more. Thus the class for which the distance is the minimum may be class to which the input signal belongs to. The input signal may be assumed to come from the musical instrument the class is representing.

IV. Results and Discussions

We have implemented the system using the MATLAB software. Multiple samples from several different types of instruments were subjected to the processing and heartening results were obtained. Table I shows the first six MFCC coefficients (rounded off to 3 decimal places) obtained for five different samples for the instrument, guitar.

Table I. MFCC Coefficients for Guitar

Samples	*Coeff1*	*Coeff2*	*Coeff3*	*Coeff4*	*Coeff5*	*Coeff6*
1	-14.054	11.959	6.262	4.699	3.383	1.826
2	-22.821	16.786	5.212	3.429	3.651	0.826
3	-20.360	10.468	3.885	3.683	3.529	1.569
4	-21.697	10.704	3.327	3.119	3.488	1.682
5	-25.086	14.113	7.894	5.498	3.346	1.400

Table II below shows the first six coefficients for a test sample of the same instrument, guitar.

Table II. MFCC Coefficients of the test sample

	Coeff1	*Coeff2*	*Coeff3*	*Coeff4*	*Coeff5*	*Coeff6*
Test Sample	-23.209	17.705	6.428	4.246	3.837	0.837

Table III below shows the Euclidean distance of the test sample from its own class (Guitar) and other instrument classes such as Flute, Trumpet, etc. The figures clearly indicate that with a distance of 5.403 the test sample is closest to the Guitar class over other classes.

Table III. Euclidean distances

Instrument Name	*Euclidean Distance*
Flute	14.6544
Guitar	5.4030
Trumpet	16.2555
Piano	14.7756
Sitar	15.2919
Harmonium	15.9362

It can thus be decisively concluded that the test sample belongs to the Guitar class. The designed system has behaved relatively well with most samples baring a few exceptions. Table IV below shows the False Reject Ratio (FRR) computed for a population size of over 100. The FRR was computed by finding the ratio of the total number of samples which were falsely rejected to the total number samples as shown in (3).

$$\text{FRR} = \frac{\text{Number of instruments which were falsely rejected}}{\text{Total number of samples tested}} \quad (3)$$

Table IV shows the calculation for the FRR for different types of musical instruments:

Table IV. FRR Calculation for instruments.

Instrument	*Number of rejected samples*	*Total number of tested samples*	*FRR of each instrument*
Flute	*7*	*18*	*0.389*
Guitar	*6*	*20*	*0.3*
Trumpet	*5*	*16*	*0.3125*
Piano	*7*	*18*	*0.389*
Sitar	*2*	*15*	*0.133*
Harmonium	*1*	*19*	*0.057*

The effective weighted FRR (eFRR) taking all the instruments into consideration is calculated using (4)

$$eFRR = \frac{\sum n_i FRR_i}{\sum n_i} \qquad (4)$$

where, n_i is the number of rejected samples for instrument i and FRR_i is the FRR of the instrument i. The eFRR was thus computed to a value of 0.326. This figure may not look too promising; however taking into consideration that the timbre, texture, etc. of the musical instruments were not taken into account the figure is definite good. This performance can be improved by incorporating other features along with the MFCC to get better results.

V. Conclusion

In this paper we have presented a scheme for classification and identification of sounds from musical instruments. We have used MFCC to obtain features of the sound samples. These features were then used to train an SVM. The final classification was done by using Minimum Distance Classifier. We have analysed the results obtained by calculating the FRR. The values show that while the technique shows promising results for some musical instruments such as Harmonium, it has also give mixed outcome for some other musical instruments such as piano and flute. An explanation for the mixed results is that MFCC is not robust enough against noises. Thus, the quality of the sound samples effect the outcome to a large extent. Sound samples with such compromised quality, may not be suitable for being used for the MFCC process even after the application of the Butterworth filter. Use of other advanced filtering techniques along with the Butterworth filter may help. These results may be improved further by using other feature extraction methods such as Linear Predictive Coding (LPC).

After multiple decades of research, the process of audio recognition has witnessed a major evolution. It has found its usage in various fields and is now a part of our daily lives. This also calls for developing software for recognizing musical instruments. This gap is expected to be filled up by our work. Amongst other things the proposed system may be used in the field of music industry, radio, Karaoke devices, machine learning, teaching and cultural research.

VI. References

[1] P. Herrera, G. Peeters and S. Dubnov, “Automatic Classification of Musical Instrument Sounds”, Journal of New Music Research, Vol. 32, 2003.

[2] M. J. Newton and L. S. Smith, “A neurally inspired musical instrument classification system based upon the sound onset”, The Journal of Acoustic Society of America, 131(6):4785-98. doi: 10.1121/1.4707535, June 2012.

[3] A. Livshin, G. Peeters and X. Rodet, “Studies and Improvements in Automatic Classification of Musical Sound Samples”, ICMC 2003, Singapore. pp.1-1,October 2003.

[4] C. M. Bishop, “Pattern Recognition and Machine Learning”, Springer, 2007.

[5] R. O. Duda, P. E. Hart, and D. G. Stork, “Pattern Classification”, (2nd Ed). John Wiley & Sons, 2000.

[6] K. Kido, “In Digital Fourier Analysis: Advanced Techniques”, Springer, 2014.

[7] Klautau and Aldebaro, “The MFCC”, In Technical report, Signal Processing Lab, UFPA, Brasil, 2005.

[8] R. C. Gonzalez and R. E. Woods, “Digital Image Processing”, (3rd. ed): Pearson Education, 2009.

[9] F. Zheng, G. Zhang and Z. Song, “Comparison of Different Implementations of MFCC”, In Journal of Computer Science & Technology, vol. 16: 582–589, 2001.

[10] B. Logan, “Mel Frequency Cepstral Coefficients for Music modeling”, In International Symposium on Music Information Retrieval, 2000.

[11] S. Muhury, G. Neogi, P. Debnath and J. Ghosh Dastidar, “Design of a voice-based system by recognizing speech using MFCC”, Computational Science and Engineering Proceedings of the International Conference on Computational Science and Engineering (ICCSE2016), CRC Press , pp 77–80, DOI: 10.1201/9781315375021-16, October 2016.

[12] D. O’Shaughnessy, “Pattern Recognition”, Volume 41 Issue 10: 2965-2979, 2008.