Audio Event Detection Using Adaptive Feature Extraction Scheme

Selver Ezgi Küçükbay[§] and Mustafa Sert[*]
Department of Computer Engineering
Başkent University
Ankara 06810 Turkey
Email: [§]seyalniz@baskent.edu.tr, [*]msert@baskent.edu.tr

***Abstract*—Audio event detection is one of the important tasks of multimedia content analysis. The noise like characteristics and the diversity of audio events make the recognition task difficult when compared with music and speech sounds. Therefore, proper application of feature extraction methods is very crucial, as well as feature selection and machine learning algorithms. Here, we propose a novel adaptive feature extraction scheme along with Support Vector Machine (SVM) learner in recognizing audio events. In our scheme, we propose to apply the widely used Mel frequency cepstral coefficients (MFCCs) feature to the problem in an adaptive way. To this end, we analyze each audio event in its frequency space to obtain a dominant frequency and then make use of the determined dominant frequency in the feature extraction phase. Extensive experiments have been conducted on sixteen (16) different audio events namely *alert*, *clear throat*, *cough*, *door slam*, *drawer*, *keyboard*, *keys*, *knock*, *laughter*, *mouse*, *page turn*, *pen drop*, *phone*, *printer*, *speech*, and *switch* using the IEEE AASP CASA Challenge Dataset to demonstrate the performance of the proposed scheme. The results show that our adaptive feature extraction scheme achieves significantly higher recognition accuracy than traditional feature extraction method with an average F-measure value of 72%.**

Keywords–Audio event detection; Audio content analysis; Environmental sound detection; MFCC; SVM;

I. Introduction

Over the last decade, there has been an increased interest in the audio community for detecting acoustic events (also called as audio events) in audio signals. The main motivation is to develop automatic methods for recognizing sounds of particular events in any environment. However, the problem is challenging for two reasons when compared with speech and music sounds: (a) the variability and (b) the diversity of audio events (AEs). The former describes the dynamic nature of AEs and may lead to the perception of an AE as a different sound at distinct location/times; the latter is about the diversity of these sounds in the environments [12]. As a result, studies in AE recognition have received some interests in the last few years [1]–[4].

Cai *et al.* [1] work on the problem of highlight sound effects detection. They used Hidden Markov Models (HMM) with different feature extractors such as Mel Frequency Cepstral Coefficient (MFCC), Zero Crossing Rate (ZCR), sub band energies, brightness and bandwidth features in their study. They combined all features in one feature vector to achieve better results during the experiments. Their system gives Precision and Recall values of 90%. Wang *et al.* [2] present an audio event sound classification system to recognize 12 different audio events. In their study they combine Support Vector Machine (SVM) and k- Nearest Neighbor (kNN) classifier. In feature selection, they use MPEG-7 audio low level descriptors, spectrum centroid (SC), spectrum spread (SS) and spectrum flatness (SF). The classification accuracy is 85.1%. Chu *et al.* [3] propose a new method based on matching pursuit (MP) algorithm for analyzing audio events. They use 14 different audio scenes. The tests are applied through using 4 fold cross validation. Their overall accuracy is 72% for MP-based feature.

Lee *et al.* [4] present a method in order to identify and segment the frames in to regions. They used Markov model based clustering algorithm. For the dataset they download 1873 video for 25 different concepts from YouTube. They evaluate their study using average precision for each class. They yield best result for cheering segments. Beritelli et al. [15] work on a pattern recognition system for background sounds such as bus, car, construction, dump, factory, office and pool. Their classifier is Neural Networks (NN) and feature extractor is MFCC. They evaluate their systems in terms of percent misclassification and indicate accuracy between 73% and 95% depending on the duration of decision window. Muhammad et al. [8] studied on an environment recognition system. They use selected MPEG-7 audio low level descriptors and MFCC feature. In their method they eliminate MPEG-7 descriptor using Principal Component Analysis (PCA) and combine with MFCC feature. In this work, restaurant, crowded street, quiet street, shopping mall, car with open window, car with closed window, corridor of university campus, office room, desert and park are used for evaluation. For only MFCC, full MPEG-7, selected MPEG-7 and their method, the system gives accuracies of 85%, 89%, 91% and 93%, respectively. Schrder et al. [7] propose an audio-event detection system. Their system consists of two–layered hidden Markov Model as backend classifier. The system is evaluated with the materials provided in the AASP Challenge on Detection and Classification of Acoustic Scenes and Events [5]. For event-based results, the optimization applied on the dataset returns Precision, Recall and F-measure value of 66%, 58% and 62% respectively. Vuegen et al. [9] design a system based on MFCCs to train a Gaussian Mixture Models (GMM) classifier and make use of the same AASP Challenge dataset for the evaluations. The reported event-based performances for precision, recall, and F-measure are 68%, 33%, and 43%, respectively. Kucukbay et al. [10] propose a system for detection the audio events in office live environment. They propose efficient representation of MFCC features using different window and hop sizes by changing the number of Mel coefficient and also they optimize

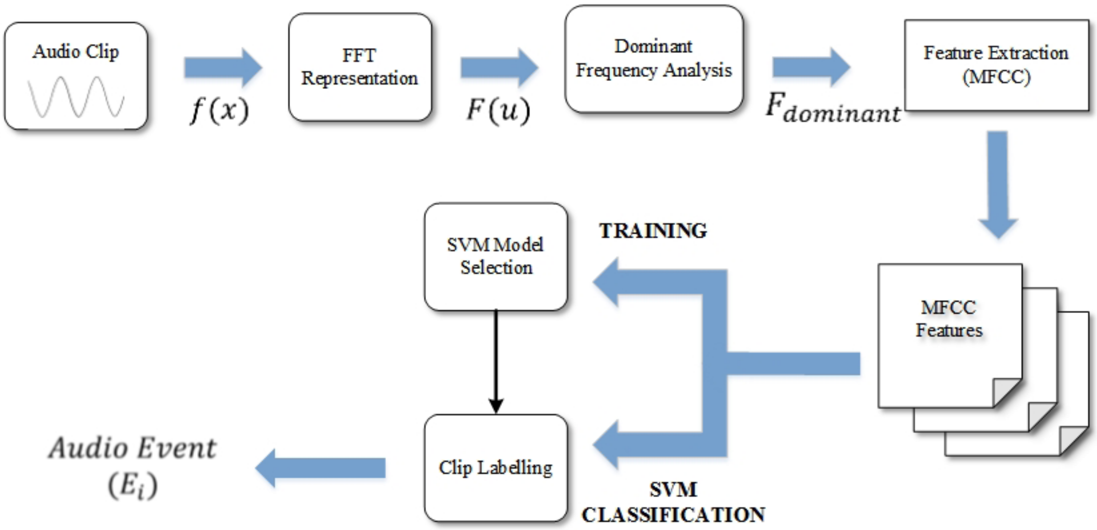

Figure 1. Block diagram of the proposed framework.

SVM parameters. The dataset are provided from subtask Office Live Environment of AASP Challenge. In the work, they use 16 distinct audio events. The tests conduct through using 5-fold cross validation gives the result of 62%, 58%, and 55% for Precision, Recall and F-measure.

Recent research shows that the performance of audio event recognition can be enhanced using suitable machine learning algorithms along with robust features [7], [11]. However, most of these studies make use of standard procedures during the feature extraction phase. For instance, in MFCC feature extraction, we obtain the coefficients from a given frequency interval, namely low- and high-frequency bounds. Using the fixed frequency bounds in the analyses of different types of sounds may lead to miss some important frequency components, since each sound source may have different bounds.

In this paper, we present a novel adaptive feature extraction scheme to recognize audio events by capturing each sound by its own frequency bounds along with SVM classifier. We consider sixteen distinct audio events from IEEE AASP CASA Challenge, namely alert, clear throat, cough, door slam, drawer, keyboard, keys, knock, laughter, mouse, page turn, pen drop, phone, printer, speech, and switch [5].

The paper is outlined as follows: In Section 2, the proposed recognition system is introduced. Empirical analysis and recognition performance are presented in Section 3 and finally, concluding remarks are given in Section 4.

II. The Proposed Scheme

The presented system consist of 7 main blocks, namely Audio Clip, FFT Representation, Frequency Analysis, Feature Extraction, MFCC Audio Features, SVM Model Training and SVM Classification. The block diagram of the proposed system is depicted in Figure 1.

A. *Adaptive Feature Extraction Scheme*

Each audio event conveys different information, i.e., comprise of different frequency components. Although the sampling rate of a signal defines the upper frequency bound of a signal in the analyses, each audio event may have its dominant frequency. In our proposed scheme, we aim to analyze each audio clip in its own frequency range during the MFCC feature extraction. Thus, we intend to capture the specific frequency range of each sound.

Specific frequency range (also referred to as dominant frequency) can be analyzed through complex methods but we verify our consideration using a fast, yet simple algorithm. In order to capture the characteristics of AEs in audio signals and to prove the effectiveness of our adaptive scheme, we use the MFCC feature due to its success in speech recognition applications. On the other hand, our proposed scheme is flexible and hence can be applied to other frequency-domain audio features. We used the standard MFCC feature extraction algorithm in [14]. In order to extract the MFCC features, we need to know the lower- and upper-frequency bounds. If we use the default values in the standard, which are defined as $300Hz$ for the lower-bound (LF) and $3700Hz$ for the upper-bound (HF), we can miss some important frequency components of an audio clip having different frequency bounds.

To solve this problem, we analyze the signal to determine its dominant frequency component. Let E denotes an audio event class (e.g., alert), then our scheme for determining the dominant frequency is given in (1).

$$f_{dominant}(E_i) = \frac{1}{N} \sum_{(k=0)\in E_i}^{N-1} f_k(idx(\max(|F_k|))) \qquad (1)$$

where $1 \leq i \leq \sharp of audio events$, E_i is the ith audio event, N is the number of audio clips in E_i, and F_k is the Fourier transformation of the k^{th} audio clip, $idx(y)$ represents the index number of y, and $f_k(z)$ denotes the frequency value of the k_{th} audio clip at index z. The main idea behind this formula is to define a dominant frequency for each AE class and make use of it in the feature extraction phase. We assume that, the most frequent frequency appeared in the signal is the dominant frequency of this clip.

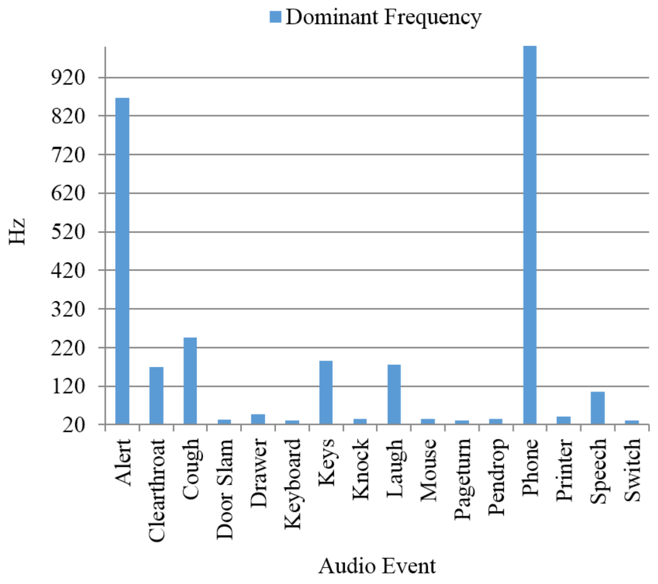

Figure 2. Dominant frequency of each audio event (AE).

TABLE I. Structure of the utilized dataset.

Audio Event Name	Duration
Alert	40 sec
Clearthroat	23 sec
Cough	23 sec
Door Slam	44 sec
Drawer	33 sec
Keyboard	1 min 16 sec
Keys	41 sec
Knock	26 sec
Laugh	30 sec
Mouse	29 sec
Pageturn	1 min 03 sec
Pendrop	16 sec
Phone	3 min 05 sec
Printer	7 min 01 sec
Speech	1 min
Switch	10 sec
TOTAL	**18 min 49 sec**

In this formula, following the Fourier transformation of the signal, which is denoted by F, we pick the frequency with the maximum magnitude as the dominant frequency and calculate the dominant frequency of each class by calculating the mean of dominant frequencies of the clips occurred in that class. Eventually, we obtain sixteen distinct dominant frequency corresponding to each AE class. We used dominant frequencies for the LF value in the feature extraction process. For the value of HF, we specified $22050Hz$ according to Nyquist theorem since the sampling rate of the clips in the dataset is $44100Hz$. Each class and their dominant frequencies are given in Figure 2. Once we find the proper frequency bounds of each class, we extract MFCC feature of each audio clip in the dataset using these frequency bounds. In our study, we choose a clip-based decision strategy for evaluating the results. When a clip is assigned to a particular class tag during the testing phase, the system selects a distinct class out of 16 different options for each frame of MFFCs that has been extracted for this particular clip.

B. *Classifier Design*

For audio event detection, we classify sounds with SVM classifier with radial basis function (RBF). This method is selected owing to achievement results in pattern recognition applications. We use LIBSVM library for the implementation [13]. Our multiclass evaluation strategy is defined as the one-versus-all approach. For each class, a separate SVM model is built such that every single SVM are trained to detect the features of particular classes and distinguish them from the others. In order to optimize the SVM parameters γ and C, we performed the grid search algorithm. Consequently, 16 different model files belonging to a particular class are created. In testing phase, the experiments conducted through using 5–fold cross validation.

III. Experimental Results

In model training and testing, we use audio event clips that are collected from the publicly available dataset of the sub-task Event Detection Office Live of the IEEE AASP Challenge Detection and Classification of Acoustic Scenes and Events [5]. These 16 distinct audio events include *short alert-beeping*, *clearing throat*, *cough*, *door slam*, *drawer*, *keyboard clicks*, *keys clinging*, *door knock*, *laughter*, *mouse click*, *turning page*, *object hitting table*, *phone ringing*, *speech*, *printer*, and *switches*.

Each class contains 20 recordings. Durations of recordings are changing because recording are collected from real-world environment. The dataset contains non-overlapping events from the office live environments. Class durations are presented in Table I.

In order to evaluate the proposed scheme, we prepared three scenarios. In the first scenario, we tested the standard MFCC implementation along with the SVM classifier. In the second one, we considered the standard MFCC feature extraction along with *optimized* SVM, and in the last scenario, we applied the proposed adaptive feature extraction scheme to the MFCC feature along with the optimized SVM. In the evaluations, we used 5–fold cross validation method and never mixed the train and the test datasets.

When we applied the proposed scheme, which considers adaptive feature extraction scheme using the dominant frequency for each class, we obtain an F-measure value of 72%.

In the second scenario in which we use fixed frequency bounds during the MFCC implementation, the recognition performance decreases to an F-measure value of 55%. And lastly, the first scenario that uses standard methods, we note an F-measure value of 48%. Our empirical results clearly show that, the proposed adaptive feature extraction scheme is superior to the standard methods and yields 17% increase in the recognition performance. Figure 3 provides a comparison for these three approaches. In addition, the proposed scheme also improves the confusion of similar AEs, such as pen drop and page turn sounds. The confusion matrices of the proposed scheme (the 3rd scenario) and the 2nd scenario are depicted in Table II and Table III, respectively. In both tables, each column of the matrix represents the instances in a predicted

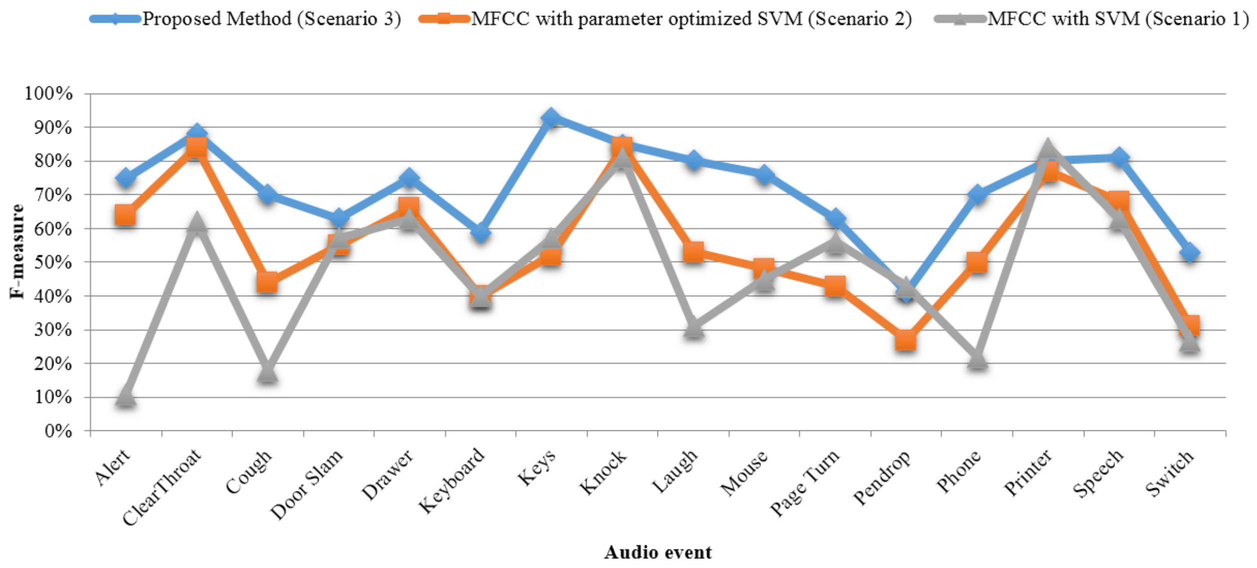

Figure 3. Recognition performances of the proposed scheme and the others.

TABLE II. Confusion matrix for 16–class classification using the proposed method (5–fold)

	Alert	Clear Throat	Cough	Door Slam	Drawer	Keyboard	Keys	Knock	Laughter	Mouse	Page Turn	Pen Drop	Phone	Printer	Speech	Switch
Alert	**13**	0	0	0	0	0	0	0	1	0	0	0	4	0	2	0
Clear Throat	0	**18**	1	0	0	0	0	0	0	0	0	0	0	0	1	0
Cough	0	0	**13**	1	0	0	2	0	0	0	0	2	1	1	0	0
Door Slam	0	0	0	**12**	4	0	0	0	0	0	1	0	0	3	0	0
Drawer	0	0	0	2	**17**	0	0	0	0	0	0	0	0	0	1	0
Keyboard	0	0	0	1	0	**11**	0	0	0	1	4	1	0	0	0	2
Keys	0	0	0	0	0	0	**19**	0	0	0	0	1	0	0	0	0
Knock	0	0	0	1	0	0	0	**17**	0	2	0	0	0	0	0	0
Laughter	0	1	1	0	0	0	0	0	**15**	0	0	0	1	1	1	0
Mouse	0	0	0	1	0	0	0	1	0	**15**	2	0	0	0	0	1
Page Turn	0	0	0	0	1	3	0	0	0	0	**15**	0	0	0	0	1
Pen Drop	0	0	1	0	1	1	0	0	0	1	4	**7**	0	3	1	1
Phone	1	1	0	0	1	0	0	1	0	0	0	0	**14**	0	2	0
Printer	0	0	0	0	1	0	0	1	0	0	0	0	0	**18**	0	0
Speech	0	0	0	0	1	0	0	0	0	0	0	0	0	0	**19**	0
Switch	0	1	0	0	0	3	0	0	1	1	2	3	0	0	0	**9**

class, whereas the rows represent the instances in an actual class. In Table III, we note that some of the sound classes such as *pen drop*, *switch*, *cough*, and *phone*, are mixing with other classes and decreases the overall performance. When used our method, we can read from Table II that the correct hit of *pen drop* increases by 3, *switch* and *phone* sounds increase by 5, *cough* sound increases by 6, and like many others increase the overall recognition performance dramatically.

This improvement can be described as using the own frequency spectrum of each sound provides the utilized frequency-spectrum feature to capture the characteristics of sounds better than using a fixed frequency range. Specifically,

TABLE III. Confusion matrix for 16–class classification using the MFCC with parameter optimized SVM (5–fold)

	Alert	Clear Throat	Cough	Door Slam	Drawer	Keyboard	Keys	Knock	Laughter	Mouse	Page Turn	Pen Drop	Phone	Printer	Speech	Switch
Alert	**15**	0	0	0	0	0	0	0	0	0	0	0	3	0	1	1
Clear Throat	0	**17**	0	0	1	0	0	0	0	1	0	0	0	0	1	0
Cough	1	1	**7**	0	2	1	2	0	4	0	2	0	0	0	0	0
Door Slam	0	0	0	**12**	3	0	0	1	0	0	1	0	0	3	0	0
Drawer	0	0	0	3	**15**	0	0	0	0	0	1	0	0	0	1	0
Keyboard	0	0	0	1	0	**9**	3	0	0	2	5	0	0	0	0	0
Keys	0	0	0	1	0	3	**12**	0	0	0	3	1	0	0	0	0
Knock	0	0	0	0	0	0	0	**18**	0	2	0	0	0	0	0	0
Laughter	0	0	3	0	1	0	1	0	**10**	1	0	0	1	1	2	0
Mouse	1	0	0	0	0	1	0	2	0	**10**	3	0	0	0	3	0
Page Turn	0	0	0	0	1	5	2	0	0	1	**11**	0	0	0	0	0
Pen Drop	0	0	1	3	1	2	3	0	0	1	4	**4**	0	1	0	0
Phone	6	0	0	0	0	0	0	1	1	1	0	1	**9**	0	1	0
Printer	0	0	0	2	1	0	0	1	0	0	0	0	0	**16**	0	0
Speech	1	1	0	0	0	0	1	0	2	0	0	0	0	0	**15**	0
Switch	2	1	0	0	0	3	3	0	0	2	3	2	0	0	0	**4**

we assign the computed dominant frequency as the low frequency bound and perform the frequency analyses using the range of $[f_{dominant}, 22050Hz]$. Another option might be to use the computed dominant frequency as the high frequency, but in this case we have to compute the low frequency bound by introducing additional computation cost, since we do not know it in advance. In our case, we know the high frequency bound in advance (i.e., 22050 *Hz* by the *Nyquist* theorem). We can read from the Figure 3 that in some cases (e.g., *printer*, *knock*, and *pendrop*), the proposed method performs similar success rates as the standard methods. To the best of our knowledge, this is because of the short durations used in the training and/or the characteristics of these sounds are quite hard to capture for the MFCC.

IV. Conclusion

This paper introduce a novel adaptive feature extraction scheme for the recognition of sixteen distinct audio events namely *alert*, *clear throat*, *cough*, *door slam*, *drawer*, *keyboard*, *keys*, *knock*, *laughter*, *mouse*, *page turn*, *pen drop*, *phone*, *printer*, *speech*, and *switch* from audio clips. In the experiments, clips are recognized and tested using the proposed scheme based on the MFCC feature and the SVM classifier.

Our study shows that, when we apply specific frequency limits for each class, we attain 72% F-measure score, which is better than both the standard methods (F-measure value of 48% and 55%) and the event-based results of the IEEE AASP Challenge (61.52% F-measure value) [8]. Based on the experiments, the proposed scheme outperforms the standard methods by 17% and the IEEE AASP Challenge results by 10.48%.

Our feature work lies on the detection of audio scenes using the audio events.

References

[1] R. Cai, L. Lie, Z. Hong-Jiang, and C. Lian-Hong, “Highlight sound effects detection in audio stream,” Multimedia and Expo (ICME’03), International Conference on, 2003, pp.37–40.

[2] W. Jia-Ching, W. Jhing-Fa, H. Kuok Wai, and H. Cheng-Shu, “Environmental Sound Classification using Hybrid SVM/KNN Classifier and MPEG-7 Audio Low-Level Descriptor,” Neural Networks, IJCNN ’06. International Joint Conference on, 2006, pp.1731–1735.

[3] S. Chu, S. Narayanan, and C.-C.J. Kuo, “Environmental sound recognition using MP-based features,” Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on, 2008, pp.1–4.

[4] K. Lee, D. Ellis, and A. Loui, “Detecting Local Semantic Concepts in Environmental Sounds using Markov Model based Clustering,” Proc. IEEE ICASSP, 2010, pp.2278–2281.

[5] D. Giannoulis, E. Benetos, D. Stowell, and M. D. Plumbley, “IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events - Development Dataset for Event Detection Task, subtask 1 - OL,” Applications of Signal Processing to Audio and Acoustics (WASPAA), 2013 IEEE Workshop on, 2013, pp.1–4.

[6] C.C. Chang and C.J. Lin, “LIBSVM: A library for support vector machines,” ACM Transactions on Intelligent Systems and Technology, Vol. 2, No. 3, 2011, pp.1–27.

[7] J. Schroder et al., “On the use of spectro-temporal features for the IEEE AASP challenge detection and classification of acoustic scenes and events,” Applications of Signal Processing to Audio and Acoustics (WASPAA), 2013 IEEE Workshop on, 2013, pp.1–4.

[8] G. Muhammad, Y. A. Alotaibi, M. AlSulaiman, and M.N. Huda, “Environment Recognition Using Selected MPEG-7 Audio Features and Mel-Frequency Cepstral Coefficients,” Digital Telecommunications (ICDT), 2010 Fifth International Conference on, pp.11–16, 2010.

[9] L. Vuegen, B. Van Den Broeck, P. Karsmakers, J. F. Gemmeke, B. Vanrumste, and H. Van hamme, “An mfcc-gmm approach for event detection and classification,” IEEE AASP Challenge on Detection and Classification Acoustic Scenes and Events, 2013.

[10] S. E. Kucukbay and M. Sert, “Audio-based event detection in office live environments using optimized mfcc-svm approach,” IEEE International Conference on Semantic Computing (ICSC’15), 2015, pp. 475–480.

[11] L. Chen, S. Gunduz, and M. T., “Mixed Type Audio Classification with Support Vector Machine,” Multimedia and Expo, 2006 IEEE International Conference on, 2006, pp.781–784.

[12] C. Okuyucu, M. Sert, and A. Yazici, “Audio Feature and Classifier Analysis for Efficient Recognition of Environmental Sounds,” Multimedia (ISM), 2013 IEEE International Symposium on, 2013, pp.125–132.

[13] L. Rabiner and J. Biing-Hwang, “Fundamentals of Speech Recognition.” Alan V. Oppenheim, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[14] C. C. Chang and C.J. Lin, “LIBSVM: A library for support vector machines,” ACM Transactions on Intelligent Systems and Technology, Vol. 2, No. 3, 2011, pp. 1–27.

[15] F. Beritelli and R. Grasso, “A pattern recognition system for environmental sound classification based on mfccs and neural networks,” IEEE International Conference on Signal Processing and Communication Systems (ICSPCS 2008), 2008, pp. 1-4.