Sound-Event Partitioning and Feature Normalization for Robust Sound-Event Detection

Baiying Lei
Department of Biomedical Engineering, Shenzhen University, Shenzhen, China
leiby@szu.edu.cn

Man-Wai Mak
The Hong Kong Polytechnic University, Hong Kong SAR, China
enmwmak@polyu.edu.hk

***Abstract*—The ubiquitous of smartphones has opened up the possibility of mobile acoustic surveillance. However, the continuous operation of surveillance systems calls for efficient algorithms to conserve battery consumption. This paper proposes a power-efficient sound-event detector that exploits the redundancy in the sound frames. This is achieved by a sound-event partitioning (SEP) scheme where the acoustic vectors within a sound event are partitioned into a number of chunks, and the means and standard deviations of the acoustic features in the chucks are concatenated for classification by a support vector machine (SVM). Regularized PCA-whitening and L2 normalization are applied to the acoustic vectors to make them more amenable for the SVM. Experimental results based on 1000 sound events show that the proposed scheme is effective even if there are severe mismatches between the training and test conditions.**

Keywords—Scream sound detection, PCA whitening and regularization, Feature normalization, Sound event partitioning.

I. INTRODUCTION

Unlike video-based surveillance, audio-based surveillance can make effective use of mobile devices. The recent ubiquity of smartphones has opened up new applications of sound event detection. For example, any smartphones can be turned into a personal security device, allowing users to detect any hazardous situations around them 24 hours a day. Abnormal sound events such as screaming can be detected and emergency phone calls can be automatically made. Ideally, such a system should be able to detect and classify a variety of sound events 24 hours a day. To prolong battery life, it is important to minimize the power consumption of the sound-event classifier. This paper proposes a sound-event partitioning scheme that can greatly reduce the computation time of the classifier.

In spite of the previous efforts in sound-event detection [1–4], the high detector error and false alarm rate remain a challenging issue, especially when the detectors are operated under severely noisy conditions. Ideally, detectors should be able to (1) detect scream sounds in very noisy environments (with SNR as low as −5dB), (2) detect very short sound events (less than 1s), (3) function properly even if the operating conditions are different from the training conditions, and (4) conserve battery power.

To address the above challenges, it is necessary to determine the acoustic features that can identify the unique scream signatures efficiently. Research has demonstrated that time-frequency representation is very useful for the classification of sound and speech signals [5-8]. While Mel-frequency cepstral coefficients (MFCCs) [9] are one of the most popular time-frequency representation, it is well-known that MFCCs are not very robust under noisy conditions. Recently, an enhanced cepstral feature, namely gammatone frequency cepstral coefficient (GFCC), is proposed for speaker recognition and speech segregation [10-12]. It was found that GFCCs are more robust than MFCCs in noisy environments. In this work, we explored the application of GFCCs to sound-event detection and compared their performance with the conventional MFCCs.

Recently, the fusions of acoustic features [11] and classifiers [13] have attracted a lot of attention, primarily because of the good performance of fusion systems as compared to systems that use individual features or classifiers alone. In this work, we combined MFCCs and GFCCs for sound-event detection in three different modes: feature fusion (concatenation), score fusion, and combination of feature and score fusion.

Another issue is the preparation of training data for constructing the classifier of a sound-event detector. Because the amount of scream sound data is much smaller than that of non-scream sounds, there is a severe imbalance between the two classes of data. Recently, a technique called utterance partitioning [14, 15] has been developed for speaker verification to address this issue. Here, we extend the technique to sound-event detection and refer to it as sound event partitioning (SEP). Specifically, given a sound event, a number of training vectors can be obtained by randomizing the frame indexes of the sound event followed by partitioning the acoustic vectors into a number of equal-length segments. For each segment (partition), an acoustic vector was then obtained by computing the mean and standard deviation of the vectors within the partition. This process allows us to obtain a lot more training vectors for each sound event, thereby boosting the performance of the resulting classifier.

For computation efficiency, support vector machines have been selected as the classifier in this work. We have performed extensive analysis as to which feature pre-processing methods is the best for the SVM classifier used in our detector. Our results suggest that PCA whitening [16, 17] followed by L2 normalization achieves the best performance. To further improve performance, the eigenvalues of the PCA are also regularized.

In the literature, sound event classification and detection is a hot topic due to its wide applications, and hence has attracted a lot of interest. For example, in [18], Guo et al. proposed an audio classification and retrieval system based on support vector machines (SVM) using both perceptual feature (e.g. total power and pitch) and MFCCs. In [6], probabilistic distance SVMs for sound event detection was investigated.

The main contribution of this work is as follows 1) acoustic partitioning of sound event is proposed and analyzed; 2) joint PCA whitening and feature normalization are employed for performance boosting; 3) feature and score fusion are investigated under very severe noisy condition. The organization of the rest of this paper is as follows. A detailed analysis of the proposed method for scream sound is discussed in Section II. Extensive experimental results are provided in Section III. Finally, Section IV provides concluding remarks.

II. METHODOLOGY

A. System Overview

The block diagram of the proposed scream detection system is shown in Fig. 1. Sound regions are first determined by an energy-based voice activity detector (VAD) [19]. MFCCs [5-8] and GFCCs [10-12] are extracted from the sound regions only. For each 32ms analysis frame, twelve cepstral coefficients (excluding energy) and their first and second time derivatives (Δ and ΔΔ) are concatenated to form a 36-dimensional feature vector (feature dimension D = 36). The MFCC and GFCC vectors are then concatenated to form 72-dimensional vectors, which are subject to regularized PCA whitening and L2-normalization. For each sound event, the acoustic vectors are divided into several partitions and the mean and standard deviation of each partition are stacked to form the final vectors for SVM classification.

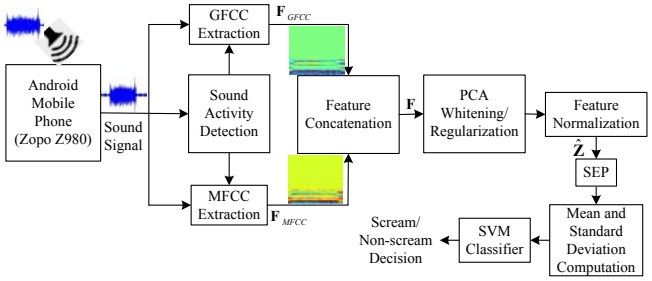


Fig. 1. Schematic diagram of the scream detection system under the feature fusion configuration (SEP stands for sound event partitioning).

B. Feature Extraction and Fusion

Recently, a new feature called GFCC [10, 11] was found to be robust to noise in speaker recognition. It has been found that feature fusion is a useful and effective way to boost the classification performance in speech segregation [11]. In spite of a great number of previous efforts [5-8] to explore discriminative features, there is no investigation on GFCC feature and fusion of MFCC and GFCC for sound-event detection. To the best of our knowledge, this work is the first to fuse MFCC and GFCC for scream sound detection.

There are two popular approaches to combine acoustic features: feature fusion and score fusion. Denote $\mathbf{F}_{MFCC}$ and $\mathbf{F}_{GFCC}$ as $D \times N$ matrices containing N frames of D-dimensional MFCC and GFCC vectors, respectively. Then, feature fusion can be written as:

$$\mathbf{F} = w_1 \mathbf{F}_{MFCC} \oplus w_2 \mathbf{F}_{GFCC}, \tag{1}$$

where w_1 and w_2 are weights for MFCC and GFCC features, respectively, and $\oplus$ is a concatenation operator.

Score fusion, on the other hand, can be implemented by linearly combining the scores obtained from MFCC- and GFCC-based classifiers. Specifically, denote s_{MFCC} and s_{GFCC} as the scores from MFCC- and GFCC-based classifiers, then the fusion score is given by:

$$s = \alpha \times s_{MFCC} + (1-\alpha) \times s_{GFCC}, \tag{2}$$

where α is a fusion weight.

To further exploit the complementarity between MFCCs and GFCCs, the scores obtained from the classifier that uses the feature-fusion vectors as input can be further combined with the scores obtained from score fusion. Mathematically, this hierarchical fusion can be written as:

$$s_f = \gamma \times s(\mathbf{F}) + (1-\gamma) \times s, \tag{3}$$

where γ is a fusion weight.

C. Regularized PCA-Whitening and l_p-Normalization

Regularized principal component analysis (PCA), whitening and l_p normalization are performed on the feature vectors. Whitening and l_p normalization are implemented by the following equation:

$$\hat{\mathbf{Z}} = \frac{diag(\lambda_1^{-\frac{1}{2}}, ..., \lambda_{D'}^{-\frac{1}{2}}))\mathbf{P}^{\mathrm{T}}\mathbf{Z}}{\left\| diag(\lambda_1^{-\frac{1}{2}}, ..., \lambda_{D'}^{-\frac{1}{2}}))\mathbf{P}^{\mathrm{T}}\mathbf{Z} \right\|_p}, \tag{4}$$

where $\mathbf{Z}$ is a D-dimensional acoustic vector, $\mathbf{P}$ is a $D \times D'$ projection matrix containing D' eigenvectors in its columns and $\lambda_1, ..., \lambda_{D'}$ are the D' largest eigenvalues. The PCA will perform whitening as well as dimension reduction when feature fusion is applied. Specifically, PCA reduces the dimension of the *MFCC+GFCC* vectors from 72 to 36. However, because the dimension of MFCC and GFCC vectors is not high ($D = 36$ only), PCA will only perform whitening on either the MFCC or GFCC vectors when no feature fusion is applied. As a result, $D' = 36$, for both feature fusion or without feature fusion. The regularization of PCA is achieved by adding a small positive value to the eigenvalues in Eq. 4:

$$\lambda_d \leftarrow \lambda_d + \beta \times \lambda_{\max}, 1 \le d \le D' \tag{5}$$

where $\lambda_{\max}$ is the largest eigenvalues and β is a regularization parameter.

As suggested in [17, 20], the above whitening and normalization process can minimize the effect of missing words and the co-occurrence of visual words in visual features. In particular, the normalization process is to suppress the double-count effect caused by co-occurred words. l_2-norm is also a common approach to compensating for the effect of document-length variability on the term-frequency vectors in document retrieval [21]. Here, we argue that the same process is also beneficial for our sound-event detector. The main reason is that our detector is based on support vector machines (SVMs) in which input space normalization

has shown to be beneficial [22]. Also, it has been shown theoretically that SVM is justified only for input vectors of constant length [23].

D. Sound Event Partitioning

Scream sounds are in general shorter than non-scream sounds such as music [6, 7]. However, there are many non-scream sounds (such as door slam, cough, and sneezing) that are much shorter than scream sounds. An analysis of the sound events used in this study suggests that most scream sounds are less than 4 seconds, but there are a few non-scream sounds (e.g., cheering) that last much longer than any of the scream sounds.

Because our proposed sound detection algorithm is designed to run on mobile devices, computation complexity and power consumption are important concerns. To minimize power consumption, we opt for an SVM classifier and use the mean and standard deviation of the acoustic feature vectors (MFCC and GFCC) across the whole sound event as the input to the classifier. However, the wide range of sound-event length suggests that using the mean and standard deviation (i.e. one input vector) for each sound event will lead to sub-optimal performance. This is because for medium and long sound events, there must be some spectral variations within the events but the mean and standard deviation fail to capture these sub-event variations. To address this deficiency, we extend our recently proposed utterance partitioning technique [14, 15] to sound-event detection. The partitioning procedure is as follows:

Step 1: For each sound event, a sequence of MFCC and GFCC vectors are computed. After feature concatenation, whitening and normalization, a feature matrix $\hat{\mathbf{Z}}$ shown in Fig. 1 is obtained.

Step 2: Randomize the frame indexes in $\hat{\mathbf{Z}}$ to produce $\hat{\mathbf{Z}}^*$ (This step follows the argument in [14, 15] that the mean and standard deviation will not be affected by rearranging the indexes).

Step 3: Partition the feature matrix $\hat{\mathbf{Z}}^*$ into M equal-length partitions and compute the mean and standard deviation for each partition to produce M vectors.

Step 4: Repeat Steps 2 and 3 R times to produce RM input vectors. Together with the mean and standard deviation of the full-length matrix $\hat{\mathbf{Z}}$, this procedure will give $RM + 1$ vectors for each sound event.

III. EXPERIMENTS AND RESULTS

A. Experiment Setup

The proposed scream detector was evaluated by using a variety of sound events. Specifically, a total of 240 scream and 760 non-scream sound files sampled at 16 kHz with 16-bit resolution were used. Table 1 summarizes the sound events used in the experiments. Metro station noise was acoustically added to these sound files. This was achieved by playing back the original sound files through a B&K Mouth Simulator Type 4227 and at the same time metro station noise was played back through another loudspeaker. The mixed signals were recorded by an Android phone (Zopo Z980).[1]

In our experiment, β in Eq. 5 was set to 0.00001, the RBF kernel parameter was set to 0.3 and penalty factor C was set to 1. The performance of the sound detector under various configurations, environmental noise levels, and parameter settings was compared based on the minimum detection cost functions (minDCF) and equal error rates (EER). For each experimental condition, ten-fold cross validation was conducted to obtain the EER and minDCF.

Table 1
Summary of the sound events in the experiments

Event Type	No. of Events	Duration (s)
Scream sound	240	0.2–6
Non-scream sound	760	0.1–139
Total	1000	66442

B. Effect of Noise Level

To investigate the effect of background noise on sound detection, babble noise from NOISEX'92 [24] was added to the scream sound events at SNR levels of 10dB, 5dB, 0dB and −5dB. Babble noise is selected due to its non-stationary characteristics and resemblance to human sounds. Fig. 2 shows the EER and minDCF achieved by the detector using different features and fusion methods under different SNRs. The fusion weights α and γ in Eq. 2 and Eq. 3 have not been optimized in this experiment, and both were set to 0.5 across all SNRs. Results demonstrate that both the score and feature fusions achieve very good performance, suggesting that using both MFCC and GFCC is better than using the individual features alone.

Fig. 2 also suggests that even under adverse acoustic conditions, the proposed detection algorithm is still able to obtain very promising results. For instance, even at SNR = −5dB, the EER (7.24%) and minDCF (0.0378) achieved by the detector are still acceptable for real-life application.

Table 2
Performance of Mismatched tests (babble noise). Equal error rate (EER in %) and minimum detection cost (in parentheses).

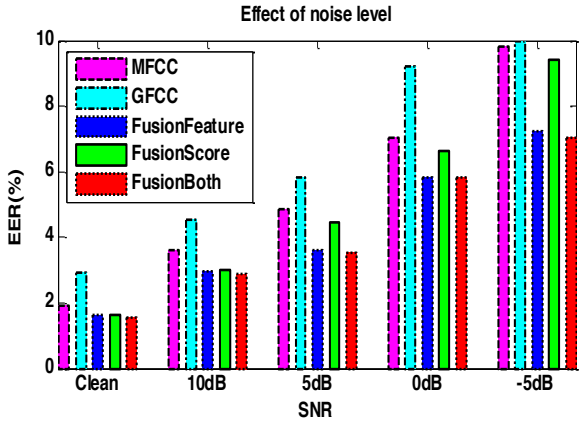
	Test clean	Test 10dB	Test 5dB	Test 0dB	Test −5dB
Train clean	2.5 (0.023)	12.11 (0.07)	16.22 (0.085)	18.39 (0.096)	20.75 (0.096)
Train 10dB	10.34 (0.057)	4.87 (0.029)	7.91 (0.043)	13.65 (0.072)	16.77 (0.083)
Train 5dB	11.96 (0.07)	7.5 (0.044)	6.96 (0.038)	9.87 (0.052)	15.34 (0.0751)
Train 0dB	14.06 (0.088)	12.24 (0.066)	9.12 (0.053)	8.31(0.046)	13.65 (0.0646)
Train −5dB	13.18 (0.089)	13.72 (0.055)	12.91 (0.07)	10 (0.062)	11.22 (0.0543)

Table 3
Mismatched tests (reverberation noise). Equal error rate (EER in %) and minimum detection cost (in parentheses).

	Test RT 0.3	Test RT 0.5	Test RT 0.7
Train RT0.3	2.03 (0.0167)	2.37 (0.0227)	3.18 (0.0249)
Train RT0.5	2.5 (0.0209)	2.37 (0.0163)	2.84 (0.0183)

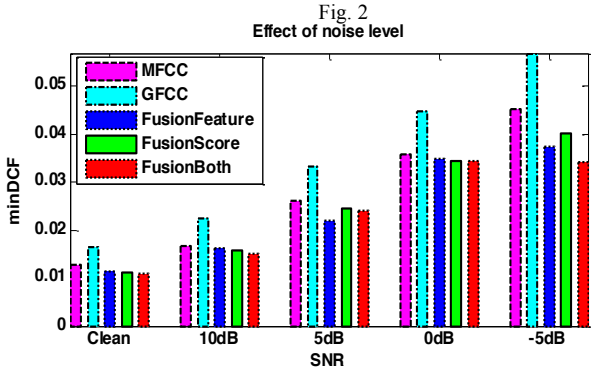
[1] The system setup and demonstration can be found in http://www.eie.polyu.edu.hk/~mwmak/SoundDetector.html

Train RT0.7	3.72 (0.0183)	3.18 (0.021)	2.77 (0.019)



(a) EER.

Fig. 2



(b) minDCF.

Fig. 2. Effect of babble noise on scream detection performance using different features and different fusion methods. For the x-axis labels, *Clean* means that sound files contaminated with metro station noise were used, whereas for the rest, babble noise was added to these sound files at the specified SNR. In the legend, *FusionFeature*, *FusionScore*, and *FusionBoth* means Eq. 1, Eq. 2, and Eq. 3 were used for the fusion, respectively. (a) EER and (b) minDCF.

C. Mismatched Noise Tests

It is of great interest to perform the mismatched tests by training on clean data but testing with data at different noise levels [25], which could further evaluate the robustness of the detection system. Moreover, robustness of the detection system to reverberation effect was also investigated by convolving the clean sound files with various room impulse responses at reverberation time of 0.3, 0.5 and 0.7 using the RIR tool [26]. Tables 2 and 3 show the EER (%) and minDCF (in the parentheses) in the mismatched noise tests. The first row of Table 2 suggests that the performance of the detector degrades rapidly when it is trained on clean sounds but tested on noisy sounds. The performances of mismatched train-test conditions (off-diagonal entries) are also significantly poorer than that of the matched conditions. However, the discrepancy between the performance of matched and mismatched conditions reduces when the SNR reduces. This suggests that for robustness consideration, the sound detector should better be trained on noisy sound files instead of clean sound files. The mismatched noise test demonstrates the robustness against various noises under different mismatched test conditions.

D. Algorithm Comparison

Experiments have been carried out to evaluate the effectiveness of PCA, regularization, whitening, l_2-norm and sound-event partitioning. Fig. 3 shows the results, where PCAR, PCAW and PCARW denote PCA with regularization, PCA with whitening, and joint PCA regularization and whitening. SEP denotes the proposed sound-event partitioning, and L2 means l_2-normalization. Note that l_2-normalization was applied to all types of PCA it is an important step to improve performance after PCA projection.

It can be seen from Fig. 3 that SEP is very effective for the systems with and without PCA and normalization. The SEP technique is able to create more informative input vectors to the SVM classifier for each sound event, which not only helps the SVM training algorithm to find better decision boundaries to discriminate scream sounds from non-scream sounds but also provides more informative input vectors to the SVM during classification. Besides, partitioning will generate more samples for training the SVM. Therefore, the performance of the systems with SEP consistently outperforms those without SEP. For systems that involve PCA, it is observed that PCA whitening is of vital importance to improve detection performance, whereas regularization could slightly improve detection performance. Moreover, it is clear from the comparisons that l_2-normalization could improve the baseline performance but not significantly without PCA projection. Among the system involving PCA, l_2-normalization is the most important step to improve performance; without l_2-normalization, the performance degrades significantly.

IV. CONCLUSIONS

In this paper, a feature normalization and sound event partitioning technique has been proposed and analyzed in a scream-sound detection system. It was found that joint PCA regularization and whitening improves the detection performance greatly. It is also found that SEP and feature normalization is very important for performance boosting. Extensive experimental results demonstrate the robustness of the proposed detection scheme to both additive and reverberation noises. The SEP and feature normalization methods could be generalized and applied to other sound detection applications.

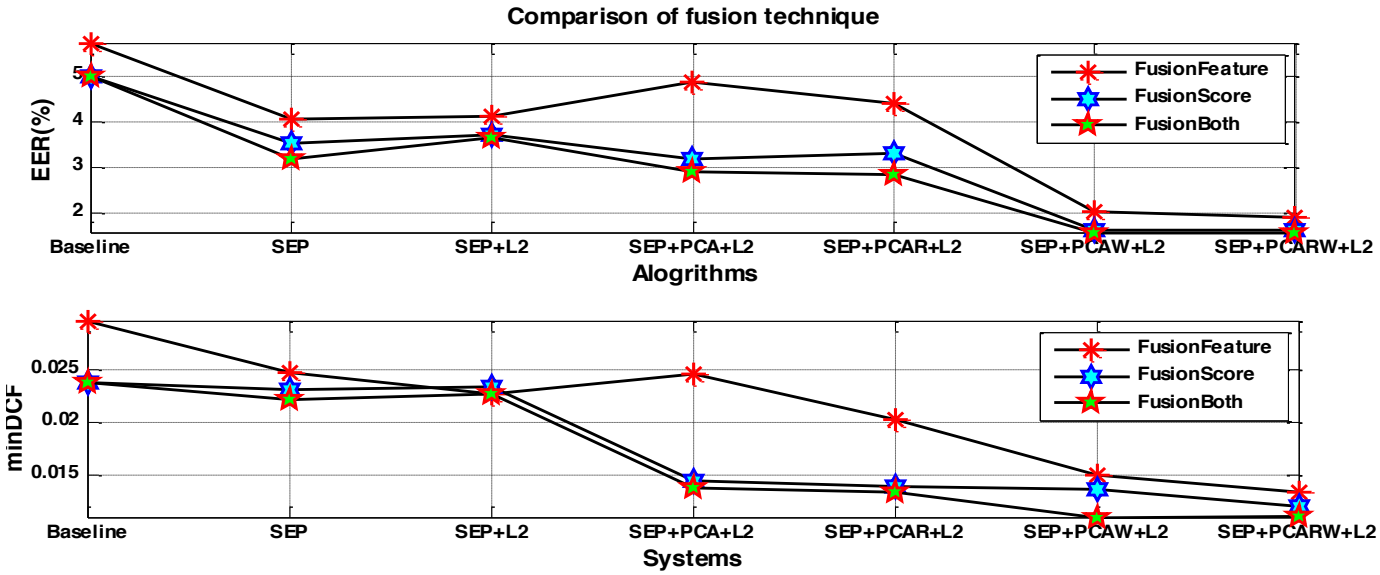


Fig. 3. Effect of fusion techniques. PCAR, PCAW and PCARW denote PCA regularization, PCA whitening, and joint PCA regularization and whitening.

ACKNOWLEDGMENT

This work was supported by Motorola Solutions Foundation (ID: 7186445) and The Hong Kong University Grant No. GYL78. Part of this work was done when Baiying Lei was affiliated with The Hong Kong Polytechnic University. The authors would like to thank Wing-Lung Leung for developing the sound recorder.

REFERENCES

[1] P. K. Atrey, N. C. Maddage, and M. S. Kankanhalli, "Audio based event detection for multimedia surveillance," in *Proc.of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2006, pp. V813-V816.

[2] C. Clavel, T. Ehrette, and G. Richard, "Events detection for an audio-based surveillance system," in *Proc.of IEEE International Conference on Multimedia and Expo*, 2005, pp. 1306-1309.

[3] M.-W. Mak and S.-Y. Kung, "Low-power SVM classifiers for sound event classification on mobile devices," in *Proc.of IEEE International Conference on Acoustics, Speech and Signal Processing* 2012, pp. 1985-1988.

[4] S. Ntalampiras, I. Potamitis, and N. Fakotakis, "On acoustic surveillance of hazardous situations," in *Proc.of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 165-168.

[5] B. Ghoraani and S. Krishnan, "Time-frequency matrix feature extraction and classification of environmental audio signals," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 19, pp. 2197-2209, 2011.

[6] H. D. Tran and H. Li, "Sound event recognition with probabilistic distance SVMs," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 19, pp. 1556-1568, 2011.

[7] S. Chu, S. Narayanan, and C. C. J. Kuo, "Environmental sound recognition with time-frequency audio features," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 17, pp. 1142-1158, 2009.

[8] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: From features to supervectors," *Speech Communication,* vol. 52, pp. 12-40, 2010.

[9] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech and Signal Processing,* vol. 28, pp. 357-366, 1980.

[10] X. Zhao and D. Wang, "Analyzing noise robustness of MFCC and GFCC features in speaker identification," in *Proc.of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 7204-7208.

[11] Y. Wang, K. Han, and D. Wang, "Exploring monaural features for classification-based speech segregation," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 21, pp. 270-279, 2013.

[12] X. Zhao, Y. Shao, and D. Wang, "CASA-based tobust dpeaker identification," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 20, pp. 1608-1616, 2012.

[13] V. Hautamaki, T. Kinnunen, F. Sedlak, L. Kong Aik, M. Bin, and L. Haizhou, "Sparse classifier fusion for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 21, pp. 1622-1631, 2013.

[14] M.-W. Mak and W. Rao, "Utterance partitioning with acoustic vector resampling for GMM–SVM speaker verification," *Speech Communication,* vol. 53, pp. 119-130, 2011.

[15] W. Rao and M.-W. Mak, "Boosting the performance of i-vector based speaker verification via utterance partitioning," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 21, pp. 1012-1022, 2013.
[16] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the fisher vector: theory and practice," *International Journal of Computer Vision,* vol. 105, pp. 222-245, 2013/12/01 2013.
[17] H. Jégou and O. Chum, "Negative evidences and cooccurences in image retrieval: the benefit of PCA and whitening," in *Proc.of European Conference on Computer Vision*, 2012, pp. 774-787.
[18] G. Guo and S. Z. Li, "Content-based audio classification and retrieval by support vector machines," *Neural Networks, IEEE Transactions on,* vol. 14, pp. 209-215, 2003.
[19] M.-W. Mak and H.-B. Yu, "A study of voice activity detection techniques for NIST speaker recognition evaluations," *Computer Speech & Language,* vol. 28, pp. 295-313, 2014.
[20] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Fisher Vector Faces in the Wild," in *Proc. of British Machine Vision Conference*, 2013.
[21] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval* vol. 1: Cambridge University Press Cambridge, 2008.
[22] S. Ali and K. A. Smith-Miles, "Improved support vector machine generalization using normalized input space," in *AI 2006: Advances in Artificial Intelligence*, ed: Springer, 2006, pp. 362-371.
[23] H. Ralf and G. Thore, "A PAC-Bayesian margin bound for linear classifiers," *IEEE Transactions on Information Theory,* vol. 48, pp. 3140-3150, 2002.
[24] A. Varga and H. J. M. Steeneken, "Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech Communication,* vol. 12, pp. 247-251, 1993.
[25] J. Dennis, H. D. Tran, and E. S. Chng, "Overlapping sound event recognition using local spectrogram features and the generalised hough transform," *Pattern Recognition Letters,* vol. 34, pp. 1085-1093, 2013.
[26] *rir*. Available: http://sgm-audio.com/research/rir/rir.html