Robust scream sound detection via sound event partitioning

Baiying Lei and Man-Wai Mak

Abstract

This paper proposes a robust scream-sound detection scheme for acoustic surveillance applications. To enhance the discriminability between scream and non-scream sounds, a sound-event partitioning (SEP) method that facilitates the extraction of multiple acoustic vectors from a single sound event is developed. Regularized principal component analysis (PCA) and normalization are applied to the acoustic vectors, which are then classified by support vector machines (SVMs). Experimental results based on 1000 sound events show that the proposed scheme is effective even if there are severe mismatches between the training and testing conditions. The experimental results also show that a gain of 60% is achieved for equal error rate (EER) compared to a classical approach (based on mel-frequency cepstral coefficients (MFCC)). Extensive analyses on different processing stages of the proposed sound detection scheme also suggest that sound partitioning and feature normalization play important roles in boosting the detection performance.

Keywords—Scream sound detection; Regularized PCA-whitening; Feature normalization; Sound event partitioning

1 Introduction

Scream sounds are produced by a long loud piercing cry that expresses extreme fear or pain. The detection of scream sounds under real-world environments is of great significance because it is an enabling technology for acoustic surveillance and monitoring. In the literature, sound event classification and detection is a hot topic due to its wide applications [1-18]. For example, in [1], Guo et al. proposed an audio classification and retrieval system based on SVM using both perceptual feature (e.g., total power and pitch) and mel-frequency cepstral coefficients (MFCCs). In [7], probabilistic distance SVMs for sound event detection was investigated. In [16], a Radon transformed audio feature was utilized for automatic pornographic detection. In [15], speech and non-speech signals were classified by a joint regression and classification method using MFCC as features. In [12], a live scream detector for home surveillance and eldercare was developed based on MFCC and SVM classifiers.

To deal with the non-stationary nature of environmental sounds and music, matching pursuit had also been applied to obtain time-frequency representation of audio signals in the literature [4,6]. Specifically, in [4], for each analysis frame, acoustic fea-

tures were obtained by computing the mean and standard deviation of the frequency and scale parameters of a pre-defined number of Gabor time-frequency atoms in the matching pursuit. In [6], matching pursuit was applied to construct a time-frequency matrix of a sound event, which was followed by a dimension reduction step where non-negative matrix decomposition was applied to extract the frequency and temporal structures of the time-frequency matrix. Both studies found that these matching-pursuit-based features were complementary to MFCCs, although the former was more computationally demanding because of the iterative nature of the matching pursuit algorithm.

Another way of representing sound features is to convert an audio signal into a spectrogram and divide the spectrogram into a number of blocks. Image processing techniques are then applied to enhance the spectrogram, and sound features are extracted from the statistics (moments about the means) of individual blocks in the enhanced spectrogram [5]. Recently, this approach has been extended to sound signal representation using sub-band power distribution, where the distribution of log-spectral power over time in each sub-band is captured [9]. With this representation, reliable and high-power spectral components can be mapped to localized regions of the sub-band power distribution, which makes the images spectrogram more robust to noise because the localization of spectral power in the image facilitates the estimation of missing feature mask. Another extension of this approach is to extract the so-called "keypoints" from the spectrogram, where the keypoints aim to capture important geometrical information about the sound [11]. It was found that spectrogram-based acoustic features are robust to environmental noise.

In spite of the encouraging results obtained by the earlier studies, 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 (around one second), (3) function properly even if the operating conditions are different from the training conditions, and (4) conserve battery power. The last requirement is of particular significance for mobile surveillance because the detectors may need to operate continuously.

To address the above challenges, it is necessary to determine the acoustic features that can identify the unique scream signatures efficiently [12-14]. Research has demonstrated that time-frequency representation is very useful for the classification of sound and speech signals [6,7,4,19]. While MFCCs [20] 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 [21,10,22]. 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 [10] and classifiers [23] have attracted a lot of attention, primarily because of the good performance of the fusion systems as compared to systems that use individual features 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 fusion and score fusion.

Another issue is the preparation of training data for training 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 [24,25] 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 is obtained by concatenating the mean and standard deviation of the vectors within the partition. The mean portion of the acoustic vector is to represent the overall spectral characteristics of the partition, and the standard deviation portion is to capture the speech variation within the partition. This process allows us to obtain more training vectors for each sound event, thereby boosting the performance of the resulting classifier.

In addition to classifier training, feature pre-processing is also very important for sound-event detection because without discriminative features, the best classifier will still fail. For computation efficiency, support vector machines (SVMs) have been selected as the classifier in this work. We have performed extensive analyses as to which feature pre-processing methods is the best for the SVM classifier used in our detector. Our results suggest that principal component analysis (PCA) whitening [26,27] (which have been extensively used in image classification) followed by normalization achieves the best performance. To further improve performance, the eigenvalues of the PCA are also regularized.

The main contribution of this work includes the following: (1) a sound event partitioning technique is proposed to increase the number of acoustic vectors of training the SVM classifier, and (2) extensive analyses on the feature pre-processing techniques (such as PCA, whitening and L2 norm) for the SVM classifier are provided. The organization of the rest of this paper is as follows. The proposed method for scream sound is detailed in Section 2. Extensive experimental results are provided in Section 3. Finally, Section 4 provides concluding remarks.

2 Methodology

2.1 System overview

The block diagram of the proposed scream detection system is shown in Fig. 1. A spectral-subtraction based voice activity detection (VAD) proposed in [28] is applied to detect the sound regions. MFCCs [20] and GFCCs [21,10,22] 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 l_2-normalization. For each sound event, the acoustic vectors are divided into several partitions, and then the mean and standard deviation vectors of each partition are stacked to form the final vectors for SVM classification. Note that 32ms analysis frames with 50% frame overlapping are typical in speech recognition systems. Because scream sounds are somewhat similar to speech sounds (as they are both produced by human), we followed the convention used in speech recognition systems. However, unlike speech recognition systems, we did not use energy in the feature vectors because scream and non-scream sound could have very similar energy.

Unlike most existing work in sound detection [6,7,4,19], the main goal of the proposed system is to detect short sound events (some of them could be less than 1 second) in very noisy environments (with SNR less than −5dB) and severe mismatches between training and testing conditions. As shown in Fig. 2 (upper panel), the time-domain signal of scream sounds will become indistinguishable from the background noise at low SNR. Nevertheless, a texture-like pattern can still be observed in the spectrogram representation as shown in the lower panel of Fig. 2. Previous work [13,8] in sound detection has also reported such phenomenon. Therefore, our proposed system uses spectral-domain features. MFCCs [20] are the most common spectral features for speech and speaker recognition. Nevertheless, they are known to be not very robust under noisy conditions [21]. Recently, a new spectral features called GFCC [21,10,22] has been proposed for robust speaker recognition. Our proposed system combines these two features for sound detection by performing within sound-event partitioning, regularized PCA, and l_2-normalization on the acoustic vectors. The methods proposed are suitable for the implementation of acoustic surveillance systems and hazard detection systems running on mobile devices. The system was evaluated using sound events recorded from a mobile phone with real environmental noise acoustically added to the original sound signals.

Table 1 compares the proposed detector with other state-of-the-art detectors/recognizers in terms of evaluation data, classifiers and acoustic features. Table 2 shows the details of our evaluation data. The pros of our evaluation data is that it comprises sound events with a wide range of durations and that the sound events were recorded under real-acoustic environments by a mobile phone. Nevertheless, the numbers of scream and non-scream sounds are imbalance because of the difficulty in obtaining scream sounds.

2.2 Feature extraction and fusion

An appropriate feature is of vital importance to sound-event detection. Time-frequency representation is appropriate because sound events are non-stationary signals. It has been found that distinctive texture-like patterns represented in the form of MFCC spectrograms are effective for sound-event detection [3,2,8,13]. Fig. 3 shows the texture-like MFCC patterns for both scream and

non-scream sounds. Apparently, the two patterns are very distinctive. Note that the differences between scream and non-scream Fsounds are more visible on the first coefficient. This is because the intensity of the image plots in Fig. 3 represents the values of MFCC (a kind of cepstral coefficients) in which the first coefficient has the largest variance across different types of sounds. In fact, cepstral coefficients have the property that the variance decreases with increasing coefficient numbers, e.g., for 12-th order MFCC, the first coefficient has the largest variance whereas the 12-th coefficient has the lowest variance.

Recently, a new feature called GFCC [21,10] was found to be robust to noise in speaker recognition [21,10]. As illustrated in Fig. 4, the first coefficients of MFCC and GFCC are not totally correlated (with correlation value less than 1.0), which means that fusion techniques such as feature concatenation and score fusion can be explored to combine the two features. It has been found that feature fusion is a useful and effective way to boost the classification performance in speech segregation [10]. In spite of a great number of previous efforts [6,7,4,19] 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 combining 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:

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

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:

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

where γ is a fusion weight and $s(\mathbf{F})$ represents the score obtained by using the concatenated feature in Eq.(1).

2.3 Regularized PCA whitening and l_2 normalization

Regularized PCA whitening and l_2 normalization are performed on the feature vectors. Given a sound event, a sequence of *D*-dimensional acoustic vectors $\mathbf{Z} = [\mathbf{z}_1,...,\mathbf{z}_T]$ is obtained, where $\mathbf{z}_i \in \mathbb{R}^D$ is the *i*-th acoustic vector. Then, these acoustic vectors are transformed as follows:

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

where $\mathbf{P}$ is a $D \times D'$ projection matrix containing D' eigenvectors in its columns and $\lambda_1,...,\lambda_{D'}$ are the D' largest eigenvalues. The transformed vectors are then concatenated to form a matrix $\hat{\mathbf{Z}} = [\hat{\mathbf{z}}_1,...,\hat{\mathbf{z}}_T]$ for further processing. 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. Note that PCA is not only for dimension reduction, but also for whitening and regularization. We found that the latter plays more important role than the former. Without feature fusion, we found that using all of the features performs the best. That is why we set D'=36. With feature fusion, because feature dimension is doubled, we reduced it to the same value as without feature fusion to avoid the curse of dimensionality problem. The regularization of PCA is achieved by adding a small positive value to the eigenvalues as below:

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

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

As suggested in [27,29], 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 compensate for the effect of document-length variability on the term-frequency vectors in document retrieval [30]. 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 SVMs in which input space normalization has shown to be beneficial [31]. Also, it has been shown theoretically that SVM is justified only for input vectors of constant length [32].

2.4 Sound event partitioning

Scream sounds are in general shorter than non-scream sounds such as music [7,4]. However, there are many non-scream sounds (such as door slam, cough, and sneezing) that are much shorter than scream sounds. Fig. 5 shows the length distribution (in terms of number of frames) in 300 scream sounds and 700 non-scream sounds used in this study (see Section 3 for the details of the dataset). Evidently, the distribution suggests that most scream sounds are less than 500 frames, which is equal to 4 seconds at a framerate of 125Hz, but there are a few non-scream sounds (e.g., cheering) that last much longer than any of the scream sounds in the dataset. Fig. 5 also shows that the length of both scream and non-scream events varies significantly.

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.[1] However, the wide range of sound-event length as shown in Fig. 5 suggests that one mean vector and one standard deviation vector will not be sufficient for representing the acoustic characteristics of medium and long sound events. 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 [24,25] 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 $\hat{\mathbf{Z}}$ is obtained.

Step 2: Randomize the frame indexes in $\hat{\mathbf{Z}}$ to produce $\hat{\mathbf{Z}}^*$. This step follows the argument in [24,25] 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 sections and compute the mean and standard deviation for each section 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.

Fig. 6 illustrates this sound-event partitioning process and the procedure for computing the mean and standard deviation vectors for each partition. As shown in Fig. 6, the number of frames in each partition is about one-quarter of that of the total number of frames. For very short events, the mean vectors of the partitions could be very noisy, causing classification error. On the other hand, for very long events, the mean vectors corresponding to the partitions are more reliable and therefore it makes sense to use multiple mean vectors rather than a single mean vector (corresponding to the whole event) for classification. As a result, our method will work better for long events.

3 Experimental results

3.1 Experiment setup

The proposed scream detector was evaluated by using a variety of sound events collected from [33]. Specifically, a total of 240 scream and 760 non-scream sound files sampled at 16 kHz with 16-bit resolution were used. Table 2 summarizes the sound events used in the experiments. The scream sound includes the male and female screaming and children screaming under different situations. The non-scream sound includes baby cry, speech, laughter, applause, sneeze and non-speech noises. 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) using mono mode with sampling frequency 16 kHz, 16 bits per sample. Fig. 7 shows the setup for recording the station-noise contaminated sound events.

In our experiment, β in Eq. (5) was set to 0.00001. In SVM classifier, the RBF kernel is used and the RBF kernel width was set to 0.3, and the penalty factor C for SVM training was set to 1.0. 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) [19,34], receiver operating characteristic (ROC), detection error tradeoff curves [34] and equal error rates (EER). For each experimental condition, ten-fold cross validation was conducted to obtain the EER and minDCF.

3.2 Effect of noise level

To investigate the effect of background noise on sound detection, babble noise from NOISEX'92 [35] was added to the scream sound events at SNR levels of 10dB, 5dB, 0dB and −5dB, using the Matlab code available from [36]. Babble noise is selected due to its non-stationary characteristics and resemblance to human sounds. Note that ideally we should have played back the babble noise in the mouse simulator and acoustically added the babble noise to the sound events. However, for each sound file, we also needed to measure the SNR of the resulting noisy file. Given that we have 1000 sound files and 4 different SNR conditions, performing such procedure will be too time consumption and tedious.

Fig. 8 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

[1] It is important to note that individual frames do not contain sufficient information for differentiating scream and non-scream sounds. In fact, individual frames of scream and non-scream sound are highly overlapped in the feature space, which will cause problems if they are directly used for training SVM classifiers.

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 feature alone.

Fig. 8 suggests that even under adverse acoustic conditions, the proposed detection algorithm still achieves 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.

The EER (%) and minDCF quantify the performance of a detection system based on one decision threshold. While these performance measures are good for comparing system performance, they do not show the trade-off between false alarm (rate of miss-classifying non-scream sounds as scream sounds) and miss probability (rate of miss-classifying scream as non-scream). The detection error trade-off [34], which is a kind of receiver operating characteristics but with non-linear axes, is designed to compare the performance of detection systems operating at a wide range of decision thresholds. As demonstrated in Fig. 9, the performance of both feature fusion and score fusion is better than that without fusion across a wide range of decision thresholds. This suggests that the fusion methods are very robust. Fig.10 illustrates the ROC curves of the MFCC, GFCC, FusionFeature and FusionScore using true positive rate and false positive rate as x- and y-axes.

3.3 Mismatched noise tests

It is of great interest to perform the mismatched tests by training on clean data but testing on data at different noise levels [11], 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 [37]. We followed the procedure specified in the PRISM-SET [38] to generate the reverberated sound [39]. Tables 3 and 4 show the EER (%) and minDCF (in the parentheses) in the mismatched noise tests. The first row of Table 3 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.

3.4 Effect of sound-event partitioning

The effect of varying the number of partitions in a sound event under the clean condition is shown in Fig. 11. Evidently, systems with partitioning generally achieve better performance than those without partitioning (number of partitions=0), especially

when an optimal number of partitions were selected. As shown in Fig. 11, as the number of partitions increases, the performance does not necessarily improve and could become worse than the systems without partitioning. The plots, however, exhibit convex shapes, suggesting a balance between the number of partitions and detection performance needs to be made. Moreover, increasing the number of partitions will also increase the computational cost. In our dataset, the experimental results demonstrate that when the number of partitions is 2 or 4, the performance reaches optimum. Generally, the results suggest that SEP is beneficial to the overall performance. It is also observed that the proposed partitioning method is applicable to both MFCC and GFCC features.

3.5 Effect of feature and score fusions

In this work, feature fusion, score fusion, and feature fusion plus score fusion have been investigated. Fig. 12 shows the results of the three fusion techniques. For each configuration in the horizontal axis in Fig. 12, the fusion weights α and γ in Eq. (2) and Eq. (3) have been optimized through cross validation and the weights w_1 and w_2 in Eq. (1) were set to 1.0. Fig. 12, SEP means sound event partitioning technique was applied, and SEP+L2 means the proposed method, namely, both l_2-normalization and SEP were applied. As expected, combining score fusion and feature fusion, namely, FusionBoth in Fig. 12, achieves the best performance in all cases. Score fusion achieves very similar results to the FusionBoth, whereas feature fusion is slightly inferior. Also, it can be seen from Fig. 12 that SEP+L2 achieves the best performance among all the cases. The primary explanation is that both feature normalization and partitioning techniques are effective for the scream sound detection.

3.6 Algorithm comparison

Experiments have been carried out to evaluate the effectiveness of PCA, regularization, whitening, l_2-norm and SEP. Fig. 13 shows the results, where PCAR, PCAW and PCARW denote PCA regularization only, PCA whitening only, and joint PCA regularization and whitening. SEP denotes the proposed partitioning technique, and L2 means l_2-normalization. Note that all PCA will also have l_2-normalization because the latter is an important step to improve performance after PCA projection. It can be seen from Fig. 13 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 boundary to discriminate scream sounds from non-scream sounds but also provides lots more informative input vectors to the SVM during classification. Besides, partitioning will generate more samples in the training. 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 per-

formance 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.

To further validate the SEP technique and different feature pre-processing methods, the detection system was tested under different SNR. Fig. 14 shows that l_2-normalization are effective for boosting the performance of SEP. A comparison between L2 and PCAW+L2 reveals that PCA whitening is beneficial to the performance. Joint regularization and whitening (PCAW+L2) further improves the performance. In general, the results demonstrate that PCA can help to find a direction that improves performance.

4 Conclusions and future work

In this paper, feature normalization and sound event partitioning techniques have 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 (i.e., environment sound detection). Generally, more than 50% performance improvement is achieved for the proposed approach than the baseline approach. We have developed an Android app based on the methods described in this paper. The app can differentiate scream and non-scream sound events in real-time. A demonstration of the app can be found in http://www.eie.polyu.edu.hk/~mwmak/SoundDetector.html.

Acknowledgment

The work was supported partly by National Natural Science Foundation of China (No. 61402296), Motorola Solutions Foundation (ID: 7186445) and the Hong Kong Polytechnic University Grant No. G-YL78. The authors would like to thank Wing-Lung Leung for writing the sound recording systems and part of Android App.

References

1. Guo G, Li SZ (2003) Content-based audio classification and retrieval by support vector machines. Neural Networks, IEEE Transactions on 14 (1):209-215
2. Clavel C, Ehrette T, Richard G (2005) Events detection for an audio-based surveillance system. In: Proc.of IEEE International Conference on Multimedia and Expo, pp 1306-1309
3. Atrey PK, Maddage NC, Kankanhalli MS (2006) Audio based event detection for multimedia surveillance. In: Proc.of IEEE International Conference on Acoustics, Speech and Signal Processing, pp V813-V816

4. Chu S, Narayanan S, Kuo CCJ (2009) Environmental sound recognition with time-frequency audio features. IEEE Transactions on Audio, Speech, and Language Processing 17 (6):1142-1158
5. Dennis J, Tran HD, Li H (2011) Spectrogram image feature for sound event classification in mismatched conditions. IEEE Signal Processing Letters 18 (2):130-133
6. Ghoraani B, Krishnan S (2011) Time-frequency matrix feature extraction and classification of environmental audio signals. IEEE Transactions on Audio, Speech, and Language Processing 19 (7):2197-2209
7. Tran HD, Li H (2011) Sound event recognition with probabilistic distance SVMs. IEEE Transactions on Audio, Speech, and Language Processing 19 (6):1556-1568
8. Mak M-W, Kung S-Y (2012) Low-power SVM classifiers for sound event classification on mobile devices. In: Proc.of IEEE International Conference on Acoustics, Speech and Signal Processing, pp 1985-1988
9. Dennis J, Tran HD, Chng E-S (2013) Image feature representation of the subband power distribution for robust sound event classification. IEEE Transactions on Audio, Speech, and Language Processing 21 (2):367-377
10. Wang Y, Han K, Wang D (2013) Exploring monaural features for classification-based speech segregation. IEEE Transactions on Audio, Speech, and Language Processing 21 (2):270-279
11. Dennis J, Tran HD, Chng ES (2013) Overlapping sound event recognition using local spectrogram features and the generalised hough transform. Pattern Recognition Letters 34 (9):1085-1093
12. Huang W, Chiew T-K, Li H, Kok TS, Biswas J (2010) Scream detection for home applications. In: Proc.of 6th IEEE Conference on Industrial Electronics and Applications, pp 2115-2120
13. Ntalampiras S, Potamitis I, Fakotakis N (2009) On acoustic surveillance of hazardous situations. In: Proc.of IEEE International Conference on Acoustics, Speech and Signal Processing, pp 165-168
14. Valenzise G, Gerosa L, Tagliasacchi M, Antonacci F, Sarti A (2007) Scream and gunshot detection and localization for audio-surveillance systems. Paper presented at the Proc.of IEEE Conference on Advanced Video and Signal Based Surveillance,
15. Liao W-H, Lin Y-K (2009) Classification of non-speech human sounds: Feature selection and snoring sound analysis. In: Proc. of IEEE International Conference on on Systems, Man and Cybernetics, pp 2695-2700
16. Kim MJ, Kim H (2011) Automatic extraction of pornographic contents using Radon transform based audio features. In: Prof. of 9th International Workshop onContent-Based Multimedia Indexing, pp 205-210
17. Kotus J, Lopatka K, Czyzewski A (2014) Detection and localization of selected acoustic events in acoustic field for smart surveillance applications. Multimedia Tools and Applications 68 (1):5-21
18. Penet C, Demarty C-H, Gravier G, Gros P (2014) Variability modelling for audio events detection in movies. Multimedia Tools and Applications:1-31
19. Kinnunen T, Li H (2010) An overview of text-independent speaker recognition: From features to supervectors. Speech Communication 52 (1):12-40
20. Davis S, Mermelstein P (1980) Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE Transactions on Acoustics, Speech and Signal Processing 28 (4):357-366
21. Zhao X, Wang D (2013) Analyzing noise robustness of MFCC and GFCC features in speaker identification. In: Proc.of IEEE International Conference on Acoustics, Speech and Signal Processing, pp 7204-7208
22. Zhao X, Shao Y, Wang D (2012) CASA-based robust speaker identification. IEEE Transactions on Audio, Speech, and Language Processing 20 (5):1608-1616

23. Hautamaki V, Kinnunen T, Sedlak F, Lee KA, Ma B, Li H (2013) Sparse classifier fusion for speaker verification. IEEE Transactions on Audio, Speech, and Language Processing 21 (8):1622-1631
24. Mak M-W, Rao W (2011) Utterance partitioning with acoustic vector resampling for GMM–SVM speaker verification. Speech Communication 53 (1):119-130
25. Rao W, Mak M-W (2013) Boosting the performance of i-vector based speaker verification via utterance partitioning. IEEE Transactions on Audio, Speech, and Language Processing 21 (5):1012-1022
26. Sánchez J, Perronnin F, Mensink T, Verbeek J (2013) Image classification with the fisher vector: theory and practice. Int J Comput Vis 105 (3):222-245
27. Jégou H, Chum O (2012) Negative evidences and co-occurences in image retrieval: the benefit of PCA and whitening. In: Proc.of European Conference on Computer Vision, pp 774-787
28. Mak M-W, Yu H-B (2014) A study of voice activity detection techniques for NIST speaker recognition evaluations. Computer Speech & Language 28 (1):295-313
29. Simonyan K, Parkhi OM, Vedaldi A, Zisserman A (2013) Fisher Vector Faces in the Wild. In: Proc. of British Machine Vision Conference, pp 8.1-8.12
30. Manning CD, Raghavan P, Schütze H (2008) Introduction to information retrieval, vol 1. Cambridge University Press Cambridge
31. Ali S, Smith-Miles KA (2006) Improved support vector machine generalization using normalized input space. In: Proc. of 19th Australian Joint Conference on Artificial Intelligence. pp 362-371
32. Ralf H, Thore G (2002) A PAC-Bayesian margin bound for linear classifiers. IEEE Transactions on Information Theory 48 (12):3140-3150
33. Human Sound Effects. http://www.sound-ideas.com/.
34. Martin A, Doddington G, Kamm T, Ordowski M, Przybocki M (1997) The DET curve in assessment of detection task performance. In: Proc.of 5th European Conference on Speech Communication and Technology, pp 1895-1898
35. Varga A, Steeneken HJM (1993) 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 12 (3):247-251
36. addnoise, http://www.mathworks.com/matlabcentral/fileexchange/32136-add-noise/content/addnoise/addnoise.m.
37. rir, http://sgm-audio.com/research/rir/rir.html. http://sgm-audio.com/research/rir/rir.html.
38. PRISM-SET, https://code.google.com/p/prism-set/. https://code.google.com/p/prism-set/.
39. Ferrer L, Bratt H, Burget L, Cernocky H, Glembek O, Graciarena M, Lawson A, Lei Y, Matejka P, Plchot O (2011) Promoting robustness for speaker modeling in the community: the PRISM evaluation set. In: Proc.of NIST 2011 Workshop,

Table 1

Comparisons between existing sound detectors/recognizers and the proposed scream-sound detector in terms of evaluation data, classifiers and acoustic features.

Ref.	Sample	classifier	Feature
Atey [3]	talk, shout, knock and footsteps (walking and running), 2 hours training and 2 hours testing	GMM	ZCR, LPC, LPCC, LFCC
Natlamprias [13]	Explosion, gunshot, scream subway	HMM	MFCC
Huang [12]	26 training scream clips, 56 testing scram clips, 49 non-scream training clips (speech, cry, break, applause, knock, laugh), 271 non-scream testing clips	SVM	MFCC
Valenzise [14]	Gunshot, scream, and noise from microphone arrays	GMM	ZCR, MFCC, spectral distribution
Tran [7]	2.5hours audio clips, 2794 training, 2782 testing	SVM	Short time energy
The proposed	**240 scream sound, 760 non-scream sound**	**SVM**	**MFCC, GFCC, MFCC+ GFCC**

Table 2

Summary of the sound events.

Sound event	Number of Events	Duration min-max (s)
applause	86	2.3-146
baby	83	2.4-22
babycry	16	4-5.4
cheer	8	18.6-41.37
cheering	60	19-3541
cough	63	9-116
crowd	16	74-2557
door	4	8-45
groan	18	13-51
grunt	77	0.1-1.2
gunshot	8	0.2-0.9
kiss	9	0.2-0.80
laugh	64	0.21-4.4
laughter	30	0.6-1.24
nose	9	0.2-1.96
phonering	9	0.8-6.44
sniff	19	0.2-2.22
sniffle	11	0.2-0.8
snore	3	2.9-3.74
snort	25	0.2-1.76
speech	10	13-1.08
spit	18	1.6-2.6
throat	24	0.2-0.84
vocal	31	0.2-0.95
whistle	50	0.2-1.65
scream	240	0.2-6
Total	**1000**	**66442**

Table 3

Mismatched test (babble noise).

	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 4

Mismatched test (reverberation noise).

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

Table 5

Algorithm comparison results.

Version	**Algorithm**	**EER**	**minDCF**
V0	Baseline	5.00	0.0247
V1	V0+PCA	4.74	0.0226
V2	V1+Whitening	4.06	0.0202
V3	V2+SEP	3.99	0.0150
V4	V3+Fusion	1.62	0.0137

Figure captions

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

Fig. 2. Waveforms and spectrograms of (a) clean scream sound and (b) scream sound mixed with babble noise (SNR −5dB).

Fig. 3. MFCC patterns for scream and non-scream sounds.

Fig. 4. Relationship between GFCC and MFCC after feature normalization.

Fig. 5. Distributions of the number of frames in scream and non-scream sounds.

Fig. 6. Sound-event partitioning (SEP in Fig. 1) and the procedure for creating input vectors for the SVM classifier. The diagram illustrates the case with 4 partitions for each sound event. The vertical columns at the top are acoustic vectors $\hat{\mathbf{z}}$ of one sound event after PCA whitening and l_2 -normalization (Eq. (4)). For clarity of illustration, the randomization of frame indexes is not shown.

Fig. 7. Illustration of the setup for recording the sound event detection system.

Fig. 8. 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.

Fig. 9. DET performance of the sound detector based on different features and fusion methods under (a) clean condition and (b) −5dB SNR.

Fig.10. ROC curves of the sound detector based on different features and fusion methods under (a) clean condition and (b) −5dB SNR.

Fig. 11. Effect of sound-event partitioning on (a) EER and (b) minDCF (Note that number of partitions=0 means no partitioning was applied).

Fig. 12. Effect of fusion techniques, where L2 represent L2-norm (L2).

Fig. 13. The EER and minDCF achieved by detection systems with and without sound-event partitioning (SEP) . The baseline denotes the system without any feature pre-processing such as PCA, L2-norm (L2), whitening (W), and regularization (R). (a) Feature fusion. (b) Score fusion.

Fig. 14. The EER and minDCF achieved by detection systems with and without SEP . The baseline denotes the system without any feature pre-processing such as PCA, L2-norm (L2), whitening (W), and regularization (R). (a) Feature fusion. (b) Score fusion.

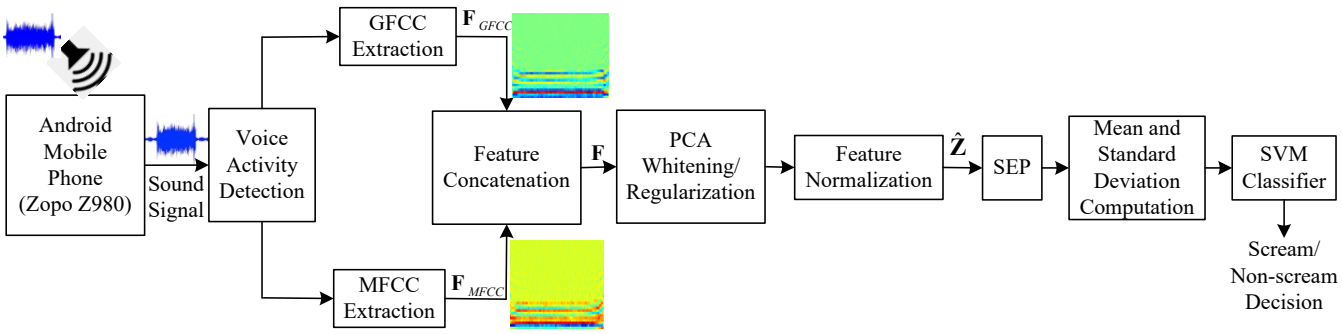


Fig. 1. Schematic diagram of the scream detection system under the feature fusion configuration (SEP stands for sound event partitioning).

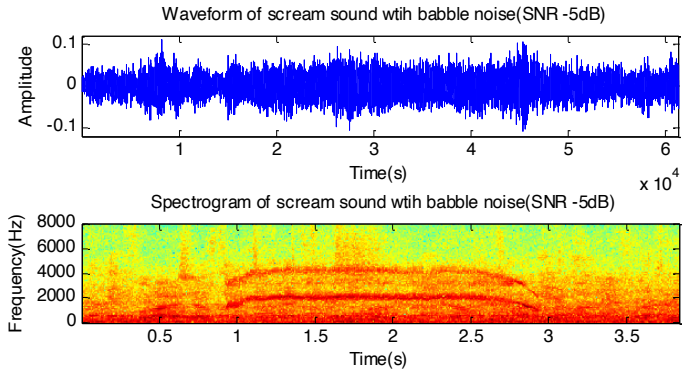
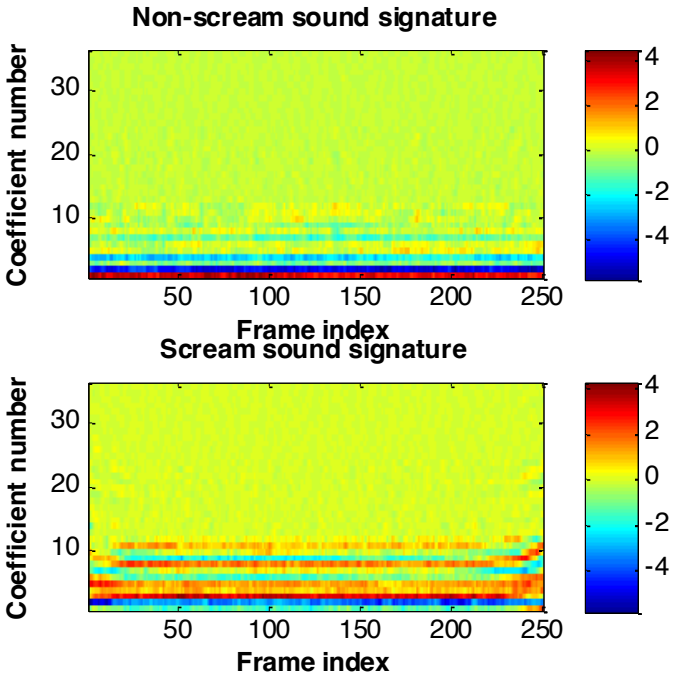


Fig. 2. Waveforms and spectrograms of scream sound mixed with babble noise (SNR −5dB).



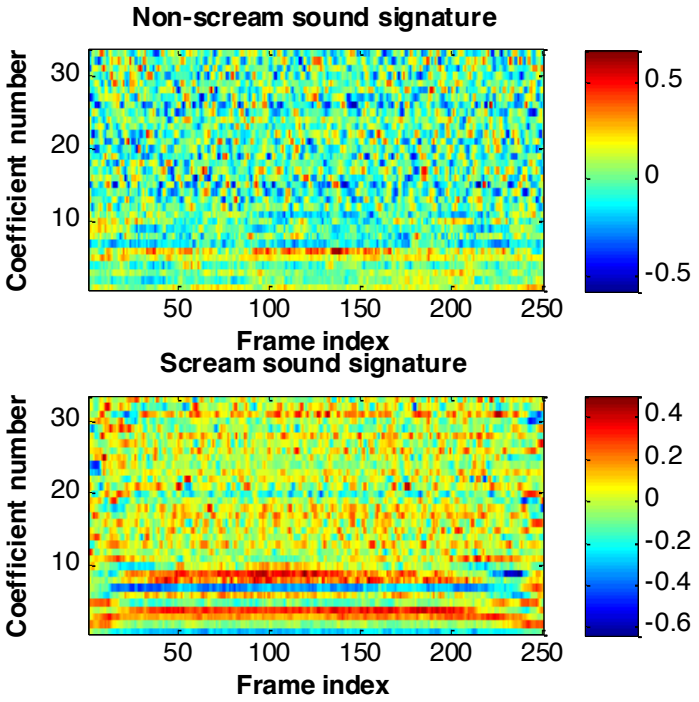


Fig. 3. Image plots showing the MFCC patterns for scream and non-scream sounds. The intensity (see colormap on the right) represents the MFCC values.

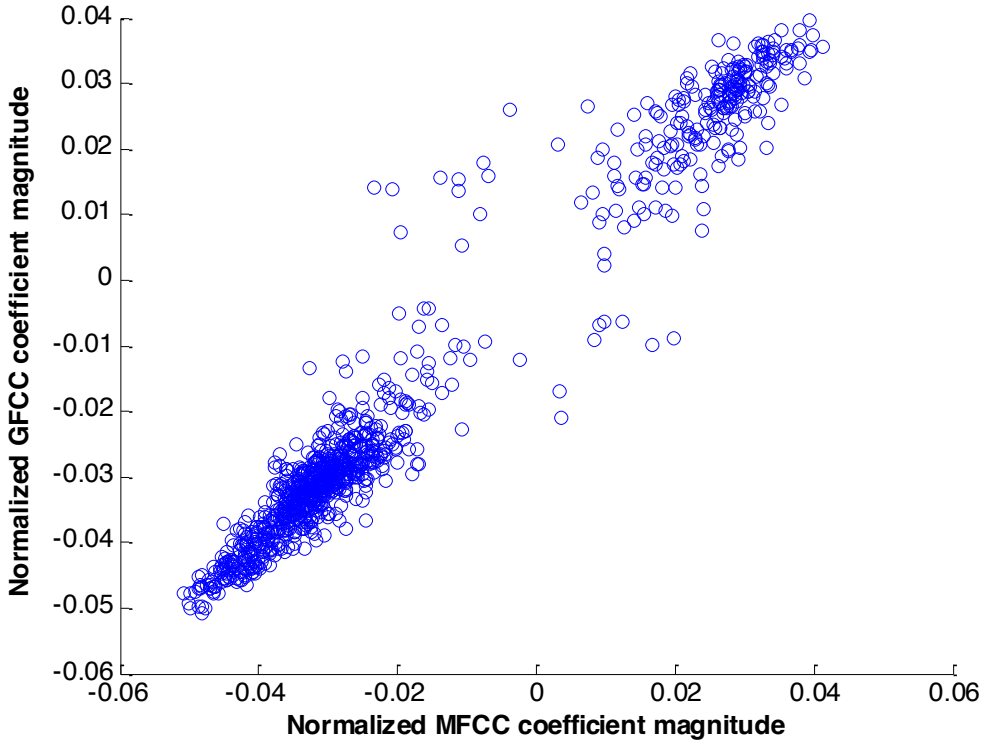


Fig. 4. Relationship between the first coefficients of GFCC and MFCC after feature normalization (Note that the correlation between GFCC and MFCC coefficients is 0.9792)

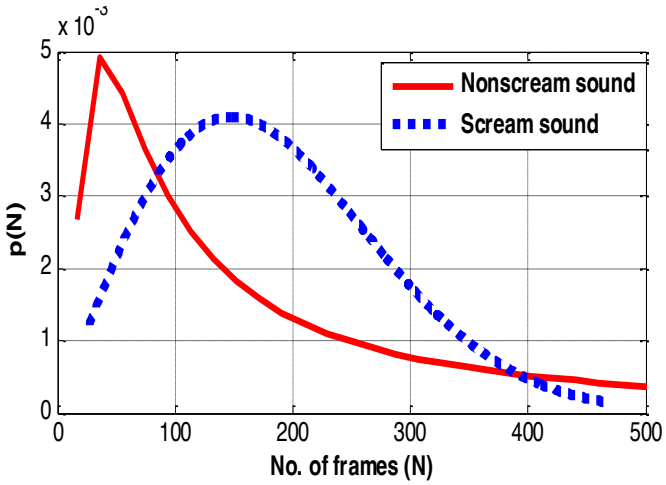


Fig. 5. Distributions of the number of frames in scream and non-scream sounds.

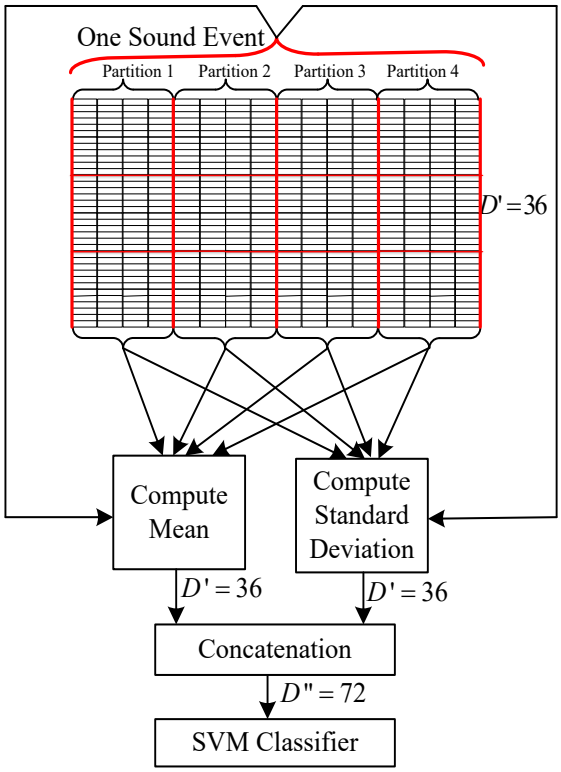
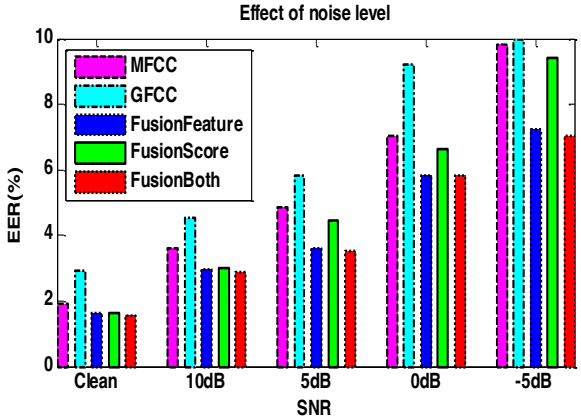
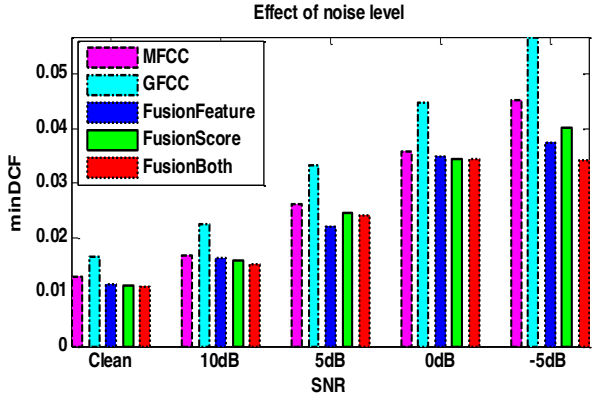


Fig. 6. Sound-event partitioning (SEP in Fig. 1) and the procedure for creating input vectors for the SVM classifier. The diagram illustrates the case with 4 partitions for each sound event. The vertical columns at the top are acoustic vectors $\hat{\mathbf{z}}$ of one sound event after PCA whitening and l_2 -normalization (Eq. (4)). For clarity of illustration, the randomization of frame indexes is not shown.

Fig. 7. The setup for recording the station-noise contaminated sound events. The mount simulator on the left are used for playing the sound events and the two speakers in the middle are for playing the station noise. The sound events and station noise are added acoustically and recorded by the mobile phone in the middle.

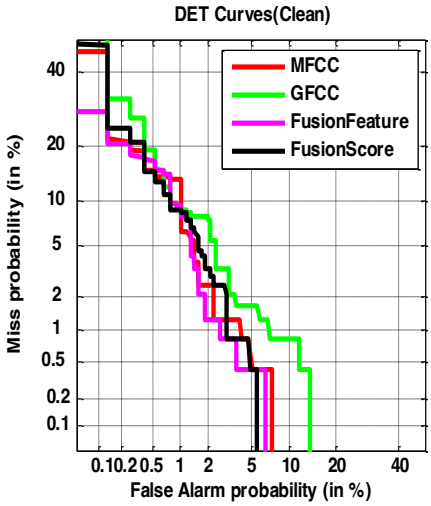


(a) EER.

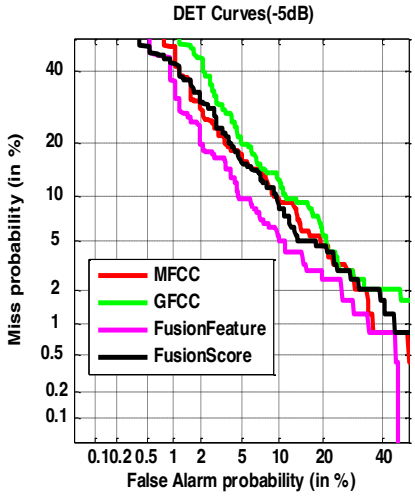


(b) minDCF.

Fig. 8. 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.

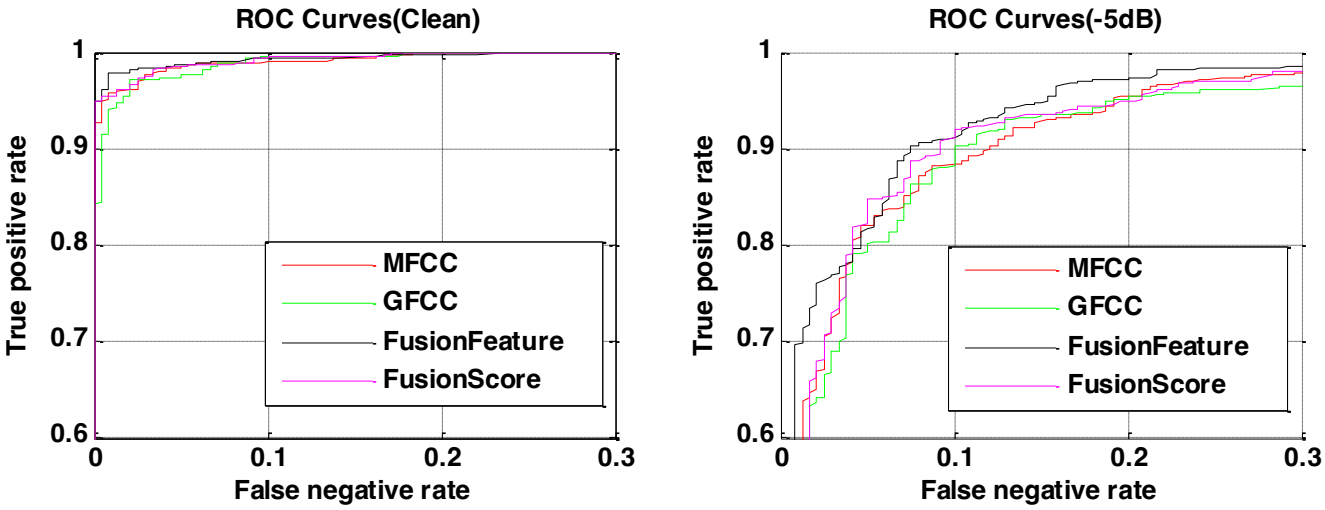


(a) Clean condition.



(b) −5dB SNR.

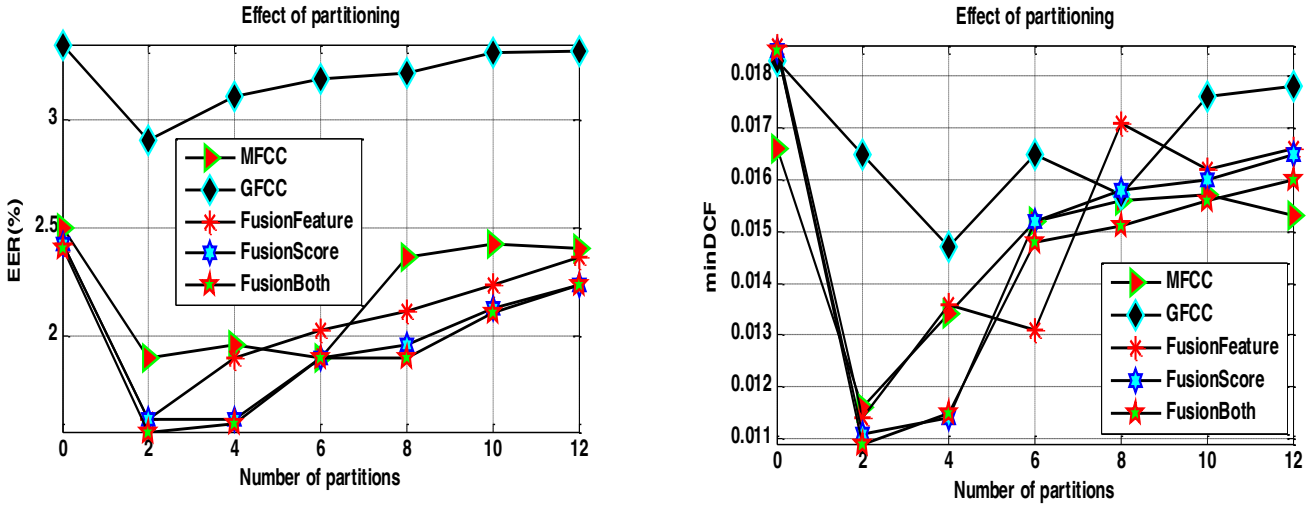
Fig. 9. DET performance of the sound detector based on different features and fusion methods under (a) clean condition and (b) −5dB SNR.



(a) Clean condition

(b) −5dB SNR.

Fig.10. ROC curves of the sound detector based on different features and fusion methods under (a) clean condition and (b) −5dB SNR.



(a) EER

(b) minDCF

Fig. 11. Effect of sound-event partitioning on (a) EER and (b) minDCF (Note that number of partitions=0 means no partitioning was applied).

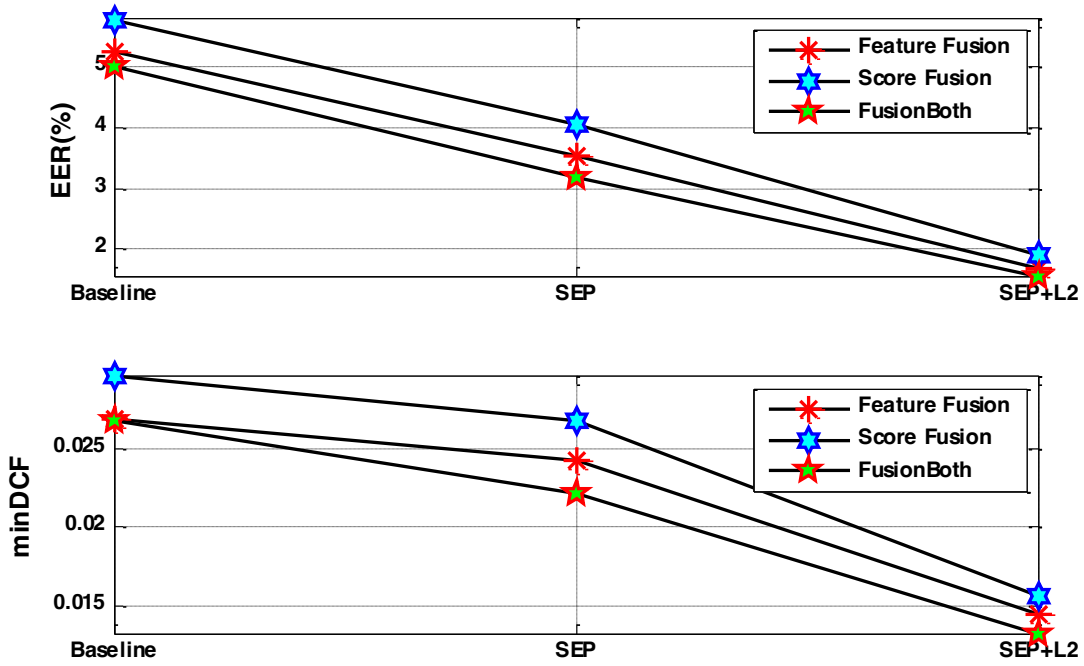
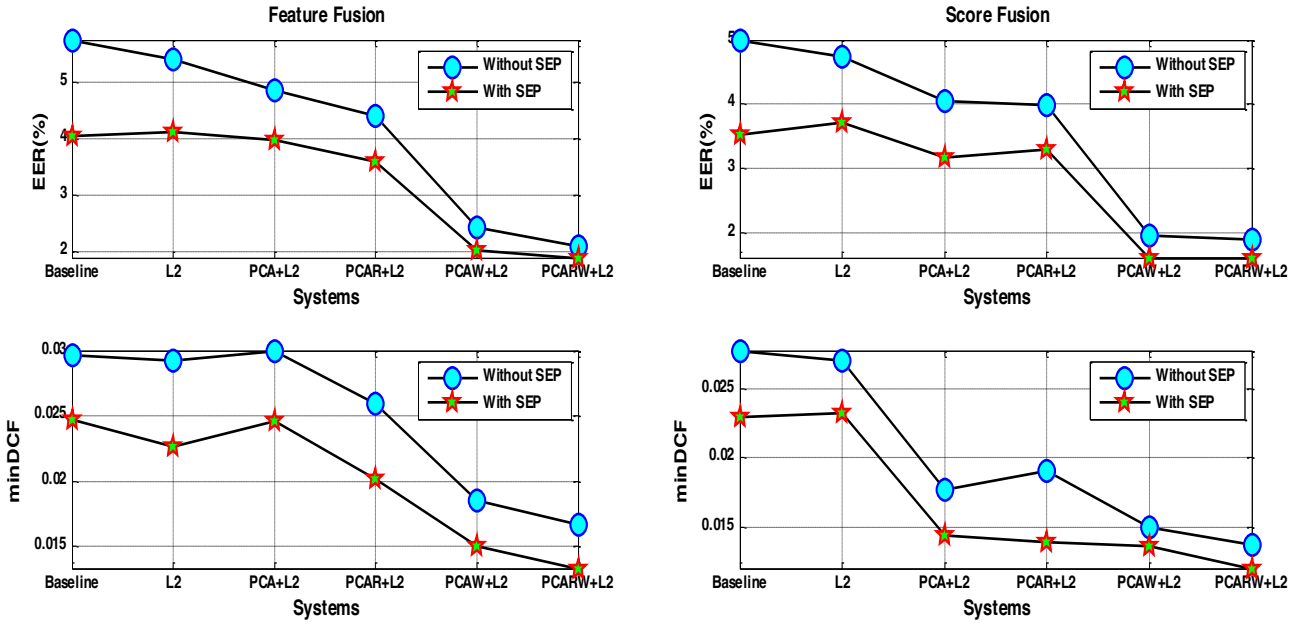
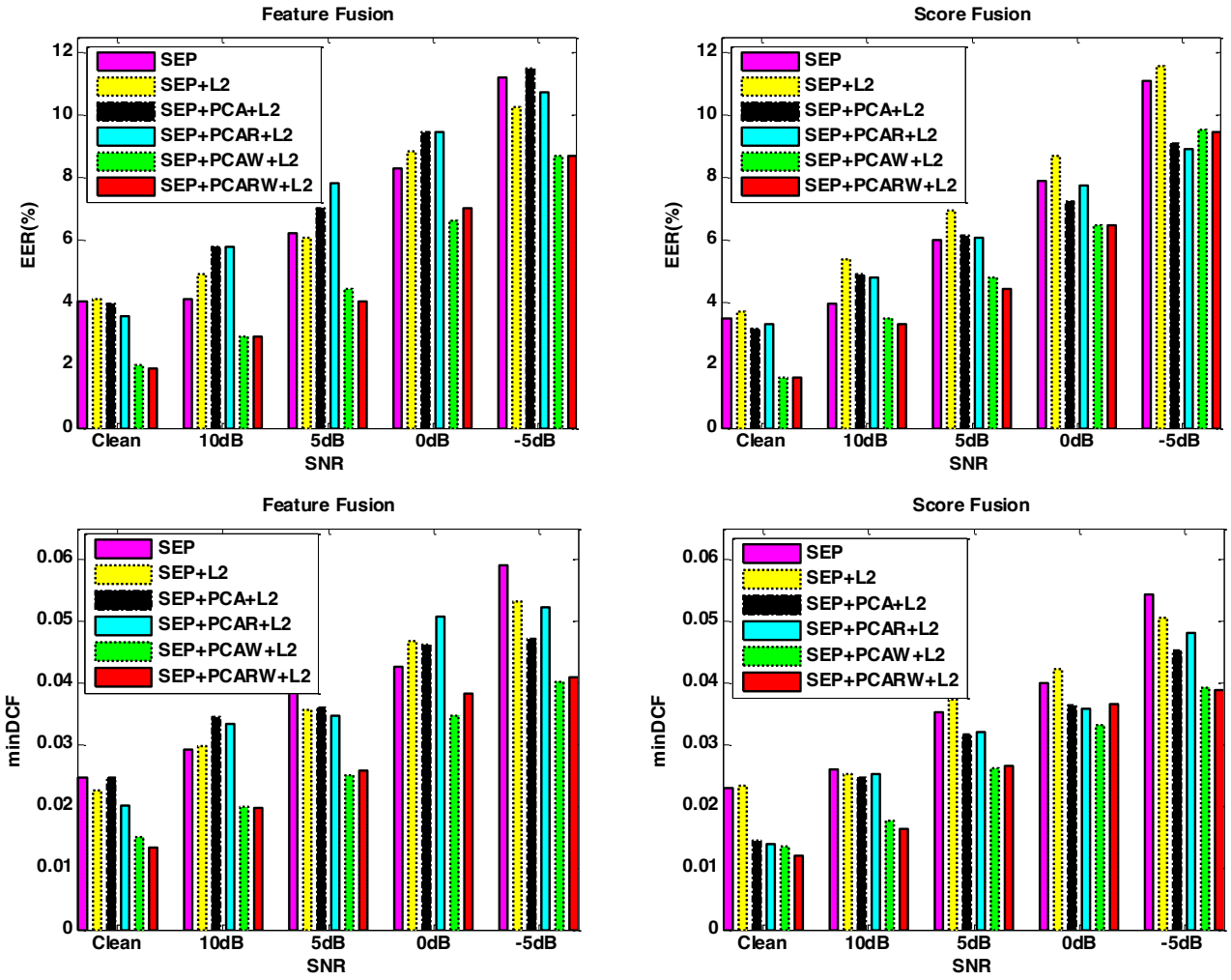


Fig. 12. Comparison of feature fusion, score fusion, and feature plus score fusion under the baseline system, systems with sound-event partitioning (SEP), and systems with SEP and feature normalization (L2).



(a) Feature fusion. (b) Score fusion.

Fig. 13. The EER and minDCF achieved by detection systems with and without sound-event partitioning (SEP) . The baseline denotes the system without any feature pre-processing such as PCA, L2-norm (L2), whitening (W), and regularization (R). (a) Feature fusion. (b) Score fusion.



(a) Feature fusion. (b) Score fusion.

Fig. 14. The performance achieved by detection systems with and without SEP and with different feature preprocessing schemes (PCA, whitening (W), L2-norm (L2), and regularization (R)) under different SNRs. (a) Feature fusion. (b) Score fusion.