ANKARA SCIENCE UNIVERSITY, RESEARCHER
Vol. 2, No. 1, July 2022
e-ISSN: 2717-9494
Research Article/Araştırma Makalesi
Doi: https://doi.org/10.55185/researcher.1107769

Reverberation Effect on Online Hazardous Sound Event Detection

Yüksel ARSLAN[1*]
[1]Ankara Science University, Faculty of Engineering and Architecture, Software Engineering Department, Ankara, Türkiye;
ORCID: 0000-0003-4791-5534
* Corresponding Author: yuksel.arslan@ankarabilim.edu.tr

Received: 24 April 2022; Accepted: 25 June 2022

Reference/Atıf: Y. Arslan, "Reverberation Effect on Online Hazardous Sound Event Detection", Researcher, vol. 02, no. 01, pp. 29-39, Jul. 2022, doi:10.55185/researcher.1107769

Abstract

This paper reports the results of the research on hazardous Sound Event Detection (SED). We used Deep Neural Networks (DNN) to detect car crashes and screams. These are the two of the hazardous sound events on which studies are done for detection. We have selected these sounds because detection of these sounds and early warning can save lives. The research made on hazardous sound events are generally on recorded data. In this paper we wanted to show that there is a difference between recorded data and online (playing) data. At the end if an audio surveillance algorithm would be used in real time, to test it with online data was also an important part of the development. In this research we have developed an online detection environment which consists of a database, automatic audio playing and receiving software, detection software and automatic evaluating software. Our tests show that the reverberation degrades performance significantly. Current research on SED usually only takes into account background noise which is inserted artificially during model development. The results we have found during these online tests are the same as the ones we encountered during far field speaker recognition.

Keywords: audio surveillance, hazardous sound event detection, deep neural networks, reverberation

1. Introduction

Sound event detection (SED) sometimes also called environmental sound recognition (ESR) can be used for many different kinds of purposes. The aim of the SED is to locate temporally and label the sound event classes present in an acoustic signal. For a SED task a set of target sound event classes should be determined. Applications of SED can be listed as follows: There are studies of SED in military, forensic and law enforcement domain. In [1], a gunshot detection system is proposed. In [2], the gunshot blast is used to identify the caliber of the gun. In [3] and [4] SED is used for robot navigation. SED can be used for home monitoring. It can be used to assist elderly people living in their home alone [5],[6]. In [7], it is used for home automation. In [8] and [9], SED is used for recognition of animal sounds. In the surveillance area, it is used for surveillance of road [10], public transport [11], elevator [12] and office corridor [13].

This paper has two contributions to the field: One is to show that the importance of online tests to assess the performance of developed models for SED. The second is the developed online testing environment. Our aim is to develop models that can be used in real environments for real-time audio surveillance. The other studies done before for the acoustic surveillance worked on the recorded data. The studies done previously in this area focus on background noise and signal to noise ratio (SNR). The event sounds to be detected are embedded into different background noises such as traffic, metro station, park etc. at different signal to noise ratio (SNR) levels. In this study during online tests, it is shown that the reverberation is another factor which will affect the performance as much as background noise. It is important to take the reverberation effect during training and testing the model.

In our case, for car crash and scream detection we have prepared event files and background files. The event files are embedded into background files at different SNR levels. Model is developed using these sound files. After model development offline tests are always done as a part of the model development process. If the model reaches required performance during offline tests, then online tests should be conducted. The same data used in offline case can be played back in laboratory environment to verify that the SED algorithms perform similarly as in offline case [14].

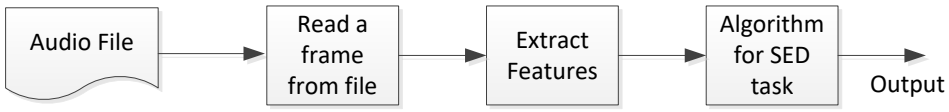

Figure 1: Offline SED [15]

In Figure 1 schematic diagram of offline SED is seen. Audio file consists of the sounds to be detected and the background sounds. In our case the detected sounds are car crashes and screams. The detected sounds are artificially embedded into background sounds. The background sounds are the sounds in which the detected sounds are required to be detected. The audio file is read frame by frame and then features are extracted from these frames. Features are fed into SED algorithm/model for detection. SED algorithm outputs 0 or 1. 0 for normal event and 1 for scream or car crash since we have a model for each detection.

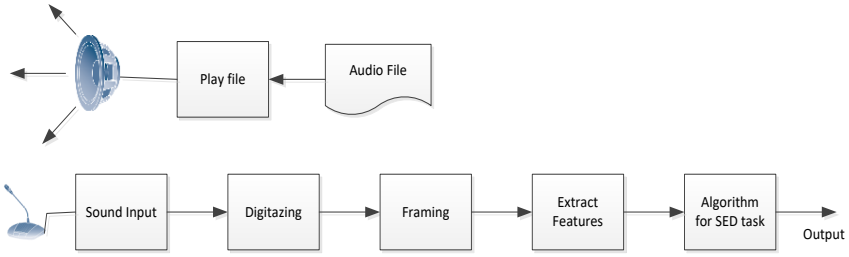

Figure 1: Online SED [15]

In Figure 2 online SED is seen. First the same audio file used in offline SED is read and played from a speaker. A microphone is used to capture these sounds and digitized by the sound card. Digitized input is framed and then same process as in offline case is applied. For online case the frequency range of microphone and speaker is considered. If our features depend on frequency that is the case for most of the time for SED, the frequency characteristics of hardware have big impact on the performance.

Our aim is to find high performance algorithms at the end for online hazardous sound event recognition. For this purpose, we used DNN. Previously other algorithms are used such as Gaussian mixture model (GMM), support vector machine (SVM), hidden markov model (HMM) and other versions of DNN such as convolutional neural networks (CNN) and recurrent neural networks (RNN) on offline data for SED. We followed the usual machine learning methods to find a model for hazardous SED. At last, we used online testing and saw that our quite high-performance model does not work as expected. We can conclude that for real-time applications it is necessary to consider reverberation during development and testing.

The other sections are as follows: In the second section we make a literature review of previous research on the same field. The third section explains the data, feature extraction, model construction for the DNN. In the fourth section we describe the evaluation method and test environment. In the conclusion section we briefly describe our contributions and future work.

2. Literature Review

There has been a lot of research on SED on offline data but has limited number of research on online SED. We have to make a distinction between online and offline research, because all the data is already available in offline case and some computations are possible on the whole data during recognition or in advance. On the other hand, in the online case our algorithms encounter reverberation at different levels depending on the environment and the speed and specs of the devices used must support the algorithms developed.

Recently, the research on SED has been shifted from HMM, GMM and SVM classifiers towards deep learning-based methods such as feed forward neural networks (FNN), CNN, RNN and convolutional

recurrent neural networks (CRNN). Detection and Classification of Acoustic Scenes and Events (DCASE) 2017 challenge Task-2 is to detect rare sound events namely glass break, baby cry and gunshot [16]. It is seen on the result page of the challenge that most of the participants have used DNN, CNN or CRNN.

Some papers from the challenge are as follows: A CRNN is used in [17]. In this paper it is declared that convolutional layers of a DNN is capable of learning high-level shift invariant features from time-frequency representations of acoustic samples, while recurrent layers can be used to learn longer term temporal context from the extracted features. These two approaches are combined to detect rare sound events. Among the proposed architectures of CRNN, the best one has an error rate of 0.1733 and F-score 91%.

In [18], it is used 1D convolutional NN as opposed to often used 2D. 128 Mel coefficients are extracted from 46 ms windows of audio signal with 50% overlap. This model achieves 0.1307 error rate and 93% F-score.

In [19], parallel CNN and RNN are used together with Mel coefficients extracted from 40 msec window of input audio signal. This model achieves 0.25 error rate and 86.4 % F-score.

Other research on SED are as follows: For environmental sound classification, deep CNN is used in [20]. This paper explains a classifier based on CNN to classify the environmental sounds such as air conditioner, siren, dog bark and gunshot. This paper also inspects the effect of data augmentation on the performance of CNN. It offers different data augmentation methods to overcome the scarcity of available labelled data. By augmented training data CNN performance increases.

For scream and gunshot detection in noisy environments [21] uses GMM. Two parallel GMM architectures are used for discriminating scream and gunshot from noise. For each classifier, features are selected from a set of 47 features by applying a 2-step selection process.

In [1], a 2-step process is offered to detect gunshot. In the first step an impulsive sound detection process is employed and then a recognition step comes. In recognition step a template matching algorithm with SVM is used.

So far, we mentioned about papers explaining algorithms mainly on recorded data to detect environmental sounds comprising also sounds from hazardous events such as gunshots and screams. From now on we will list some papers which are application oriented, running online or declared in the paper that it is an online algorithm.

In [22], an approach based on the bag of words is proposed for audio surveillance. Authors have prepared a dataset to test the algorithm in realistic complex scenarios. Gunshot, glass break and scream are detected in various background sounds with 6 different SNR values ranging from 5 dB to 30 dB. Recognition rate and false positive rate (FPR) are used as metric. Average recognition rate is 86.7% and FPR is 2.6%.

In [23], the same dataset used in [22] is used to show the performance of a hierarchical RNN. The accuracy of the hierarchical RNN outperforms the work in [22].

In [24], detection, localization and recognition of hazardous sound events are described. This work has a difference from the others such that the embedding of event sounds is not being done on computer. The different environmental noise and the hazardous event recordings are played in an anechoic room and recorded in controlled way. Then the algorithms are run on these recordings.

In [25], audio surveillance is used for car crash detection. The audio signals are divided into short time frames and then features such as spectral spread, volume, energy, zero crossing, energy in 4 sub-bands etc. are extracted. For M classes to be detected M + 1 SVM are used to detect these sound classes and the background sound. This paper also discusses the architectural deployment of such a system in real environments.

3. Model Preparation

SED consists of two stages. First stage is to represent the sound and the second stage is classification. To represent sound, fixed length frames of the sound are taken and some features are extracted from these frames. First sound signals are divided into fixed length frames and then these frames are divided into windows. Then MFCC features are extracted from windows of a frame. The feature extraction can be defined formally as follows: x is the vector of acoustic features obtained from one frame of sound signal. x is obtained from matrix M with $\mathbb{R}^{SxF}$, where $S \in N$ is the number of features per window and $F \in \mathbb{N}$ is the total number of windows per frame. Then x is the vectorization of the matrix M. M is vectorized by concatenating each column. (Figure **2**) The DNN (classifier) task is to find frame probability $\hat{y} = p(y \mid x,\theta)$ for target output $y \in \{0,1\}$(One class output), where $\hat{y}$ denotes the probability of target event in the frame and θ is the parameters of the classifier. Then by applying a threshold on the frame probability, the estimated event z found, if $z = 1$ the event is present in the frame or if $z = 0$ the event is not present [17]. The parameters θ are trained by supervised learning.

The parameters (θ) of the classifier are found through training of the classifier by giving the labelled features extracted from the frames of the training data. This is called supervised learning. In the following section we explained how we found the data and its details.

Car crash dataset (Table 1) is taken from the research described in [25]. This dataset contains 56 files, in four folds. The duration of each of these files are 3 min. We have used three folds for training and last fold for testing. The dataset we have used totally contains 204 car crashes. These are inserted in 56 background files at 15 dB SNR level. Audio files are sampled at 32 KHz. The detailed explanation of the dataset can be found in [25].

Table 1: Car Crash Dataset [15]

Training Dataset		**Test Dataset**
Type	**Number of Files**	**Number of Files**
Car crash	150 (events)	54 (events)
Traffic	42	14

Scream dataset is taken from the research explained in [22]. The scream dataset contains 66 files for training and 20 files for testing. Each file has a 3 min duration. The dataset contains sound files at 6 different SNR values, but we have taken only the files with SNR values of 15 dB. Table 2 shows the scream dataset properties. The 2084 scream even sounds are inserted 86 background files. The background files contain different sounds such as metro, park, traffic e.g.

Table 2: Scream Dataset [15]

Training Dataset		**Test Dataset**
Type	**Number of Files**	**Number of Files**
Scream	1881(events)	203 (events)
Various	66	20

Figure 3 shows the feature extraction process that is used in detection of scream, and car crash. Each sound signal is divided into frames according to the predetermined length. These frames are then divided into 40 msec windows with 50% overlapping. From each 40 msec window, 40 MFCCs are computed. Then concatenating all MFCCs of one frame, we obtain the feature vector of the frame. So, each event sound has different size feature vector. These vectors have the output label 1. The same procedure with the same framing and windowing sizes are applied to background sounds. The obtained vectors are labelled as 0.

The minimum length of event signals and the number of MFCCs which are contained in their feature vector is shown in Table 3.

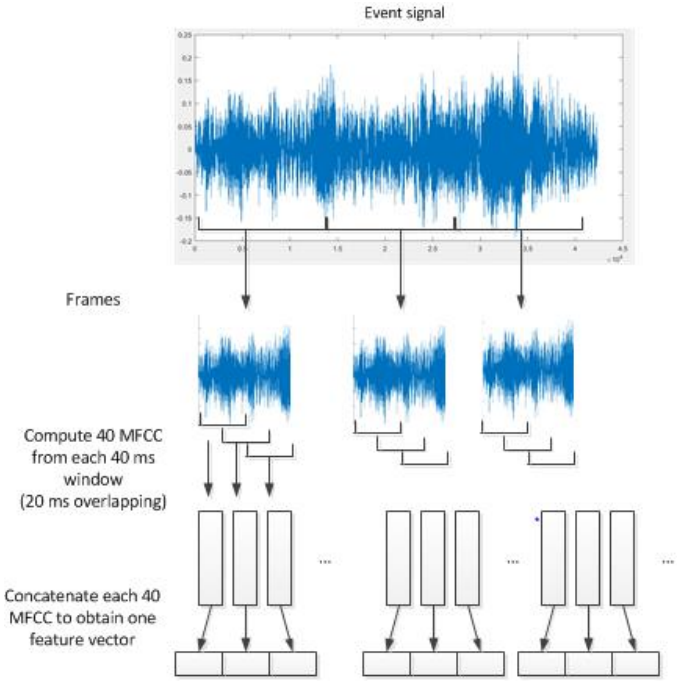

Figure 2: Feature Extraction [15]

Table 3: Event Sound Feature Properties [15]

Event Name	**Minimum Duration (ms)**	**Feature vector MFCC count**
Car crash	711	1360
Scream	490	920

We used DNN as classification algorithm to recognize the sound events. Two DNN models have been developed for each event type. DNN is a supervised and parameterized learning method. In supervised learning, we are given a set of input–target output pairs, and the aim is to learn a general model that maps the inputs to target outputs. Supervised learning methods aim to learn a model that can map the inputs to their target outputs for a given set of training examples. During model generation the model is also tested by using examples not used during learning. Table 4 summarizes the hyper parameters of the DNN used. We used rectified linear unit (RelU) as activation functions of hidden units and sigmoid function at the output unit. Figure 4 shows the DNN architecture used for car crash recognition.

Table 4: DNN Hyper-parameters [15]

Hyper - parameters	**Car crash**	**Scream**
# of layers	4	3
Learning rate	0.085	0.075
Number of iterations	1000	500
Hidden Unit Activation	RelU	RelU
Output activation function	Sigmoid	Sigmoid

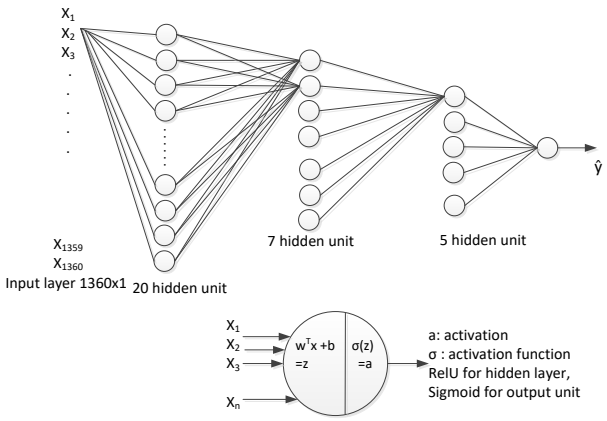

Figure 3: DNN Architecture for Car Crash Detection [15]

We have found hyper-parameters manually by making extensive tests. Grid search algorithms can be used to find best hyper-parameters automatically. Our aim is to find hyper-parameters for an acceptable performance and make offline and online testing with these parameters.

4. Testing

If we will use SED for audio surveillance, the developed algorithms or models will be used eventually in real life applications. The models we developed for road, home or elevator surveillance eventually will be installed in real life environments. After model development and testing with offline data it is necessary to see the efficiency of the models by switching more realistic scenarios.

4.1 Offline Tests

Offline testing provides fast testing on large datasets while online data speed is the recording length of the dataset. Although offline testing is not realistic, it provides us to test our system with large data sets in shorter time.

We read sound files containing the hazardous sound event embedded in a background noise in frames of minimum event length. From these frames we calculate MFCC features as in the training of the model. Then the obtained features are fed into DNN model. These tests are done for each event type separately. A sound file may contain scream and car crash events at the same time, but we detect just scream in scream model tests.

F-score, error rate and other metrics such as accuracy, true positive and false positive rates are used for model evaluation. In SED it is better to check more than one metric at the same time to evaluate the model performance. In SED surveillance applications the most important metrics are recognition rate, true positives, false positives and error rate. False positives and error rate are very important in real applications because the user can stop using application if he/she get many false warnings. The definitions for the metrics and formulas are as follows [27]:

True positives (TP): an event in the system output that has a temporal position overlapping with the temporal position of an event with the same label in the reference. A collar is usually allowed for the onset and offset, or a tolerance with respect to the reference event duration.

False positives (FP): an event in the system output that has no correspondence to an event with same label in the reference within the allowed tolerance.

False negative (FN): an event in the reference that has no correspondence to an event with same label in the system output within the allowed tolerance.

True negative (TN): truly not detected events.

Insertions (I): the number of events in system output that are not correct

Deletions (D): the number of reference events that were not correctly identified

Substitutions (S): events in the system output that have correct temporal position but incorrect class label.

Total events (N): total number of events need to be detected.

$$\text{Error rate (ER)} = \frac{S+I+D}{N} \quad (1)$$

$$\text{F-score} = \frac{2 .P. R}{P+R} \quad (2)$$

$$\text{P} = \frac{TP}{TP+FP} \quad (3)$$

$$\text{R} = \frac{TP}{TP+FN} \quad (4)$$

The metrics defined in [27] are for polyphonic sound event detection, in this paper we deal with monophonic sound event detection where each sound clip contains one type of event sounds. For our case we can write the error rate as follows:

$$\text{Error rate (ER)} = \frac{I+D}{N} = \frac{FP+FN}{N} \quad (5)$$

TP rate which is also called sensitivity is the percentage of events correctly detected. FP rate is percentage of non-event frames labelled as an event. Recognition rate is the percentage of TPs, TNs to the total frames such that TPs, TNs, FPs and FNs. The formulas are as follows:

$$\text{TPR (sensitivity)} = \frac{TP}{TP+FN} \quad (6)$$

$$\text{FPR} = \frac{FP}{FP+TN} \quad (7)$$

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (8)$$

These metrics are explained in more detail in [27]. As in [16] 500 msec tolerance was used for detection of events in this work. For offline detection, we assume the detected event on time if the detected time is 500 msec before the event start time or 500 msec after event end time. For online detection 1 sec tolerance was used. The performance of scream detection is seen in Table 5. Table 6 shows comparison results of scream detection with the results of the research in [22] and [23]. The proposed method in this work has a recognition rate 98.4% and outperforms [22] which is 87% and it is very close to the work done in [23].

Table 5: Performance of Scream Detection [15]

Accuracy (%)	**TPR (%)**	**Error rate**	**F-score (%)**
98.4	87.6	0.47	75.4

Table 6: Accuracy Comparison of Scream Detection with Proposed Method and Other Two Studies on The Same Dataset [15]

Proposed method (%)	**Foggia et al. (%)**	**Colangelo et al. (%)**
98.4	87	98.5

Car crash detection results are seen in Table 7.

Table 7: Car Crash Detection Performance [15]

Accuracy (%)	**TPR (%)**	**Error rate**	**F-score (%)**
98.4	77.7	0.35	81.5

The comparison of the car crash results with the work in [25] are seen in Table 8.

Table 8: Accuracy Comparison of Car Crash Detection with Proposed Method and The Work in [25] on The Same Dataset [15]

Proposed method (%)	**Foggia et al. [25] (%)**
98.4	84.5

4.2 Online Tests

For online tests an automatic testing environment was prepared as shown in Figure 5. The microphone and speaker frequency responses are important for the tests to be successful. The audio clips used in offline tests are sampled at 32 KHz for scream and car crash which means the sounds can have frequency components at most at 16 KHz. Therefore, the microphone and the speaker we used must support at least 16 KHz frequency as we used MFCC as feature and it depends on the audio clips frequencies. For this reason, a microphone and speakers which supports the frequency range from 80 Hz to 20 kHz are selected in online testing environment.

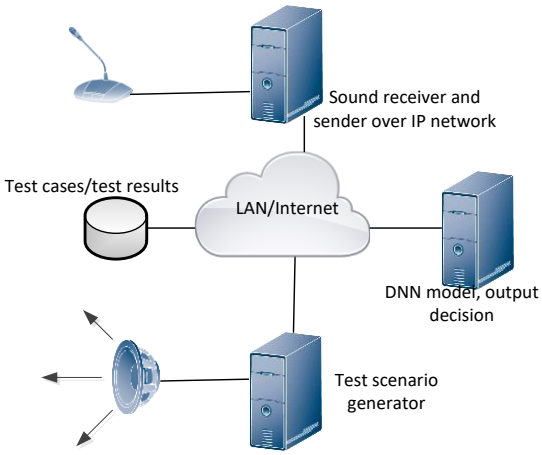

Figure 4: Online Test System [15]

We have prepared and installed three software on the computers shown in Figure 5 which we call online test system.

DNN model/output decision: This software runs the car crash and scream models. It has a user datagram protocol (UDP) receiver, and it takes the sound packets over network. We developed it as a standalone server that can run in distant place from the sound source, it can be a computer over cloud.

Test scenario generator/player: It reads the test sound files, writes the start time of playing of the sound files as offset and the start time of the events to the test cases table of the database. Then it plays each test sound file one by one.

Sound receiver and sender over IP network: The third software captures the played sound with a microphone and sends it over the local area network (LAN) or over Internet using UDP to the recognition software, namely it is DNN model/output decision software. If the recognition software detects car crash or scream it writes its decisions and the detection time to the test results table of the database. After inspecting test cases and test results the performance of the online system is found.

The online tests were done in three different room environments and in anechoic room. When the microphone is close to speakers such that the distance was less than 30 cm, the performance was the same as offline tests. In all three rooms we achieved almost the same results as offline case. When the distance is greater than 50 cm, the performance of online tests was degraded significantly. We repeat the same online tests in an anechoic room. We set the distance 2 m in this anechoic room, and we obtained the same results as offline tests.

Table 9 and Table 10 show the online and offline test result comparisons of scream and car crash detection respectively. Online results shown in tables below are the results obtained when the microphone to speaker distance is 30 cm.

Table 9: Comparison of Offline and Online Scream Detection [15]

	TPR	Error rate	F-score(%)
Offline	87.6	0.47	75.4
Online	80.2	0.50	72.3

Table 10: Comparison of Offline and Online Car Crash Detection [15]

	TPR	Error rate	F-score(%)
Offline	77.7	0.35	81.5
Online	75.4	0.40	75.7

4.3 Discussion

Online tests show that the performance of offline tests can only be achieved if the microphone is close to speaker. In our case, in approximately 30 cm we obtained the same results. If we increase the distance, performance will decrease gradually, it is half of the offline performance at about 50 cm. This performance decrease is due to reverberation. To prove this, we repeat the online tests in an anechoic room. In the anechoic room, we could place the microphones 2 m apart at most because the anechoic room that we could access was a small one. In this anechoic room, we measured the performance the same as offline performance.

The effect of of distance in speaker recognition is a known and studied problem. We encountered the same problem when we are making hazardous SED tests online. The problem of distant speaker recognition (DSR) can be explained as whenever speaker to microphone distance increases, recognition rates decrease and equal error rate (EER) increase [28]. In speaker recognition solutions such as reverberation compensation, feature warping or using multiple microphones can improve performance significantly [29].

In online hazardous SED we can use the followings to improve the performance:

- During training and model preparation we can prepare a sound database degraded with reverberation manually.
- We record pure event sounds and then degrade it with different level of reverberation and then use these sounds for training.
- We record pure event sounds and play and record the these sounds just at the same place where this SED application will be used. We then use these sounds for training and model preparation.
- We can develop the model as we did before but we we can apply de-reverberation methods before giving the sound to DNN model.

5. Conclusion

In this work we showed that hazardous SED models developed will show poor performance in real world applications. To use these models in a real-world scenario reverberation should be reconsidered. To show this after developing car crash and scream models we tested them online. The online performance of these models is less than offline performance when the microphone is apart from speaker. We proved that the reason is reverberation by repeating the tests in anechoic room.

We propose if the SED is used for an online application reverberation must be considered. As the future works the proposed de-reverberation techniques can be applied during online tests. The proposed four methods can be used to remove the reverberation effect and other methods can be proposed. Finally, the best one or combination can be used.

Conflicts of Interest

The author declares no conflict of interest.

References

[1] T. Ahmed, M. Uppal and A. Muhammad, “Improving Efficiency and Realibility of Gunshot Detection Systems”, IEEE, ICASSP 2013.

[2] P. Thumwarin, T. Matsuura and K. Yakoompai, "Audio forensics from gunshot for firearm identification", Proc. IEEE 4th Joint International Conference on Information and Communication Technology Electronic and Electrical Engineering Tailand, pp. 1-4, 2014.

[3] S. Chu, S. Narayanan, C.J. Kuo, M.J. Mataric,., “Where am I? Scene recognition for mobile robots using audio features”, in 2006 IEEE Int.Conf. on Multimedia and Expo. IEEE, 885–888, 2006.

[4] N. Yamakawa, T. Takahashi, T. Kitahara, T. Ogata, H.G. Okuno, “Environmental sound recognition for robot audition using Matching-Pursuit”, International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Springer Berlin Heidelberg, 1–10, 2011.

[5] J. Chen, A.H. Kam, J. Zhang, N. Liu, L. Shue, “Bathroom activity monitoring based on sound”, in Pervasive Computing, Springer Berlin Heidelberg, 47–61, 2005.

[6] M. Vacher, F. Portet, A. Fleury, N. Noury, “Challenges in the processing of audio channels for ambient assisted living”, in 2010 12th IEEE Int. Conf. on e-Health Networking Applications and Services (Healthcom), IEEE, 330–337, 2010.

[7] J.C. Wang, H.P. Lee, J.F. Wang, C.B. Lin, “Robust environmental sound recognition for home automation”., Automation Science and Engineering, IEEE Transactions on, 5 (1) (2008), 25–31.

[8] R. Bardeli, D. Wolff,, F. Kurth, M. Koch, K.H. Tauchert, K.H. Frommolt, “ Detecting bird sounds in a complex acoustic environment and application to bioacoustic monitoring”, Pattern Recognition. Letters, 31 (12) (2010), 1524–1534.

[9] F. Weninger, B. Schuller, “Audio recognition in the wild: static and dynamic classification on a real-world database of animal vocalizations”, in 2011 IEEE Int.Conf. on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2011, 337–340.

[10] P. Foggia, A. Saggese, N. Strisciuglio, M. Vento, and N. Petkov, “Car crashes detection by audio analysis in crowded roads”, In Advanced Video and Signal Based Surveillance (AVSS), 2015 12th IEEE International Conference on, pages 1-6, Aug 2015.

[11] J. Rouas, J. Louradour, and S. Ambellouis, "Audio Events Detection in Public Transport Vehicle," Proc. of the 9th International IEEE Conference on Intelligent Transportation Systems, 2006.

[12] R. Radhakrishnan and A. Divakaran, “Systematic acquisition of audio classes for elevator surveillance,” in Image and Video Communications and Processing 2005, vol. 5685 of Proceedings of SPIE, pp. 64–71, March 2005.

[13] P. K. Atrey, N. C. Maddage, and M. S. Kankanhalli, “Audio based event detection for multimedia surveillance,” in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP ’06), vol. 5, pp. 813–816, Toulouse, France, May 2006.

[14] T. Virtanen, M. Plumbley, D. Ellis, “Computational Analysis of Sound Scenes and Events”, book, Springer, 21 Sep. 2017.

[15] Arslan Y. Detection and recognition of sounds from hazardous events for surveillance applications. PhD, Yıldırım Beyazıt University, Ankara, Turkey, 2018

[16] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, “DCASE 2017 challenge setup: tasks, datasets and baseline system”, in Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017), November 2017.

[17] E.Cakır and T. Virtanen, “Convolutional Recurrent Neural Networks for Rare Sound Event Detection”, DCASE 2017, 27 Nov. 2017.

[18] H. Lim, J. Park, K. Lee, Y.Han, “Rare Sound Event Detection Using 1D Convolutional Recurrent Neural Networks ”, DCASE 2017, 27 Nov. 2017.

[19] A. Dang, T. H. Vu, J. C. Wang, “Deep Learning For DCASE 2017 Challenge”, DCASE 2017, 16 Nov. 2017.

[20] J. Salamon and J. P. Bello, “Deep convolutional neural networks and data augmentation for environmental sound classification”, IEEE Signal Processing Letters, Vol. 24, No.3, March 2017.

[21] L. Gerosa, G. Valenzise, F. Antonacci, M. Tagliasacchi and A. Sarti, "Scream and gunshot detection in noisy environments," in EURASIP, Poznan, Poland, September 2007.

[22] P. Foggia, N. Petkov, A. Saggese, N. Strisciuglio, M. Vento, "Reliable detection of audio events in highly noisy environments", Pattern Recognition Letters, vol. 65, pp. 22-28, 2015.

[23] F. Colangelo, F. Battisti, M. Carli, A. Neri, “Enhancing audio surveillance with hierarchical recurrent neural networks”, Advanced Video and Signal Based Surveillance (AVSS), 2017 14th IEEE International Conference on, Sept 2017.

[24] K. Lopatka J. Kotus, A. Czyzewski, “Detection, classification and localization of acoustic events in the presence of background noise for acoustic surveillance of hazardous situations”, Multimedia Tools and Applications, 75:1–33, 2016.

[25] P. Foggia, N. Petkov, A. Saggese, N. Strisciuglio, M. Vento, "Audio surveillance of roads: A system for detecting anomalous sounds", IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 1, pp. 279-288, Jan 2016.

[26] Y.Arslan and H. Canbolat, “A sound database development for environmental sound recognition”, Signal Processing and Communications Applications Conference (SIU), 25th, 2017.

[27] A. Mesaros, T. Heittola, T. Virtanen, “Metrics for Polyphonic Sound Event Detection” Applied Sciences, vol. 6, no. 6, p. 162, 2016.

[28] M. A. Nematollahi, S. A. R. Al-Haddad, "Distant speaker recognition: an overview", International Journal of Humanoid Robotics, pp. 1550032, 2015.

[29] Q. Jin, T. Schultz, A. Waibel, "Far-Field Speaker Recognition", *IEEE TASLP*, vol. 15, no. 7, pp. 2023-2032, Sept. 2007.