Empirical Study of Drone Sound Detection in Real-Life Environment with Deep Neural Networks

Sungho Jeon[1], Jong-Woo Shin, Young-Jun Lee, Woong-Hee Kim, YoungHyoun Kwon, and Hae-Yong Yang
The Affiliated Institute of ETRI
Daejeon, South Korea
Email: {sdeva, jwshin, lhyjlee, whkim, wishwill, formant}@nsr.re.kr

***Abstract*—This work aims to investigate the use of deep neural network to detect commercial hobby drones in real-life environments by analyzing their sound data. The purpose of work is to contribute to a system for detecting drones used for malicious purposes, such as for terrorism. Specifically, we present a method capable of detecting the presence of commercial hobby drones as a binary classification problem based on sound event detection. We recorded the sound produced by a few popular commercial hobby drones, and then augmented this data with diverse environmental sound data to remedy the scarcity of drone sound data in diverse environments. We investigated the effectiveness of state-of-the-art event sound classification methods, i.e., a Gaussian Mixture Model (GMM), Convolutional Neural Network (CNN), and Recurrent Neural Network (RNN), for drone sound detection. Our empirical results, which were obtained with a testing dataset collected on an urban street, confirmed the effectiveness of these models for operating in a real environment. In summary, our RNN models showed the best detection performance with an F-Score of 0.8009 with 240 ms of input audio with a short processing time, indicating their applicability to real-time detection systems.**

I. Introduction

Motivation. Popularization of commercial hobby drones brings unexpected threats to the environment in which we live, such as terror to people or important facilities. A common four-propeller drone is suitable to enjoy as a hobby and for broadcasting, however, at the same time, it surprisingly makes existing defense systems appear to be outdated legacy systems. Some accidents already proved that these drones can easily penetrate the highest level of security systems, such as landing in front of the prime minister of Germany, on the rooftop of the official residence of the prime minister of Japan, and at the White House in the United States. Thus, the ability to detect the appearance of a drone is a matter of the highest priority to prevent any threats.

Existing work. Even though few studies have been concerned with the problem of drone sound detection, previous work was conducted in isolated or calm places rather than in a real-life environment without the polyphonic sound environment typical of outside areas, such as on the rooftop of a building in a calm place or isolated environment [1], [2], [3]. However, considering our target problem, which is to detect drones used for malicious purposes, the system inevitably needs to be utilized in a real-life environment, and this requires us to consider polyphonic sound data. Other work differs by using an impressive approach based on radar information or the RF frequency [4], [5], but we need to consider a combined detection system with a multiple approach to complement the drawback of each method.

Event Sound Classification (ESC) in a real environment has been highlighted for diverse purposes. Many researchers have focused on finding useful features and classifiers based on the machine-learning approach. The most popular combination of feature and classification is Mel-frequency Cepstrum Coefficients (MFCC) [6] with the Gaussian Mixture Model (GMM) [7], [8]. More recently, the impressive success achieved with Deep Neural Networks (DNNs) has motivated researchers to introduce these networks to environmental sound recognition. Two popular DNN models, the Convolutional Neural Network (CNN) [9], [10] and Recurrent Neural Network (RNN) [11], have also been highlighted for audio-related tasks. Even though these previous studies cover the ESC problem, considering the importance and urgency of our problem in terms of terrorism, it is worth exploring how ESC work can be applied and to assess its effectiveness for drone sound detection. Here it should be noted that rather than intended to propose novel features or models for drone sound detection, our work aims to investigate the practical effectiveness of popular classification models for our problem in real environments used in previous ESC studies.

Contribution. Our contributions are summarized as follows:

- To the best of our knowledge, we are the first to investigate drone sound detection in highly noisy real environments with the aim of constructing a detection method for practical usage with real-time systems based on three popular ESC models: GMM, CNN, and RNN.
- We show that the shortage of training data for a drone sound classification model can be remedied with our audio augmentation that synthesizes raw drone sound with diverse background sounds.
- We investigate the effectiveness of these models for a testing dataset collected from real-life environments in terms of the F-Score and by taking consideration of the processing time for application to real-time systems.

II. Method

A. Data Augmentation

Especially in real environments, unseen event sound has a detrimental effect in terms of deterioration of the detection rate.

[1]Contact email: sdeva14@gmail.com

ISBN 978-0-9928626-7-1 © EURASIP 2017

The most challenging difficulty for this work is the absence of public drone sound data for training. Although supplying for commercial hobby drone is available, collecting drone sound in diverse environments is only possible to a limited extent, because flying a drone in most public or residential areas is restricted. We therefore remedied the shortage of training data, by augmenting the drone sound with diverse real-life environmental sounds from a public dataset [12], [13] and our collection. The drone sounds were collected in a quiet place outside. The purpose of this augmentation is to produce drone sound combined with realistic noise data, while preserving the characteristics of drone sound. Data augmentation involved amplifying the power of the drone sound such that it exceeded that of the background sound data by 5% in terms of max peak to emphasize the characteristics of the drone sound. Our augmented audio clip consists of concatenated raw background sound and overlapped background sound with repeated drone sounds equal to the length of the background.

B. Feature: MFCC and Mel-spectrogram

In many previous ESC studies, MFCC is known to possess outstanding features for classifiers. MFCC also has useful features to capture periodicity from the fundamental frequencies caused by the rotor blades of a drone. Our recorded drone sound indicated a noticeable harmonic shape below a frequency of 1500 Hz. In addition, we also observed a noticeable influence area on the spectrogram between 5000 Hz and 7000 Hz (Figure 1) as was previously pointed out [1], [2]. However, these characteristics were not exhibited by all the drone models used in our experiments. Furthermore, low-frequency data is carried farther than high-frequency data in terms of their energy. Therefore, we only focus on low-frequency data below 1500 Hz.

The other important consideration for feature engineering is the length of instantaneous input data to the model. The minimum length of audio data converted to an MFCC vector and that shows the best performance with our GMM configuration is 40 ms with 50% of overlapping. The other models, CNN and RNN, deliver the best performance when they process data of at least 240 ms in length, converted to mel-spectrogram with mel-bin as 40.

C. Classifier1: Gaussian Mixture Model

The GMM detector we construct consists two GMMs trained by positive and negative respectively. For a given length of audio data, it is clipped as fixed windows, which is described as the sample $X = x_1, x_2, ..., x_l$, where l is the frame length. Then we compare the log-likelihood (L) of both models with a decision threshold to decide drone appearance, $Label_{predicted} = L_1 - L_2 > \theta_{decision}$. In our experiment, GMM, with the number of Gaussian as 13, the number of MFCC as 20, and the number of mel-bin as 40, shows the best detection performance. Higher values for these parameters, as proposed in previous work, lead to the overfitting problem that shows higher detection performance in training, but produces dissatisfactory results on the testing dataset. The type of covariance shape affected by the detection performance is nearly 0.1 in our training, although we apply a diagonal shape instead of a full shape to alleviate the overfitting effect.

TABLE I: Our CNN architecture

Input size	Description
(3, 3, 1, 32)	(3, 3) reception field with kernel size 1 to 32
(3, 3, 32, 32)	(3, 3) reception field with identical kernel size
-	max pooling with (1, 2, 2, 1)
Drop-out	Drop-out with 0.5 probability
(3, 3, 32, 256)	(3, 3) reception field with kernel size 32 to 256
(3, 3, 256, 256)	(3, 3) reception field with identical kernel size
-	max pooling with (1, 2, 2, 1)
Drop-out	Drop-out with 0.5 probability
(3*10*256, 1024)	Full-connected layer
Drop-out	Drop-out with 0.5 probability
(1024, 2)	binary output class label

D. Classifier2: Convolutional Neural Network

CNN for audio-related tasks showed outstanding results with spectral features instead of focusing on feature engineering [9], [10]. The main idea of a CNN is the use of a convolutional layer that performs localized filtering for local connectivity. This local connectivity is known to be effective to capture invariance useful patterns and highly correlated values with time-frequency representation of sound signal data. Our observation that drone sound has noticeable invariance characteristics below 1500 Hz with harmonics (Figure 1).

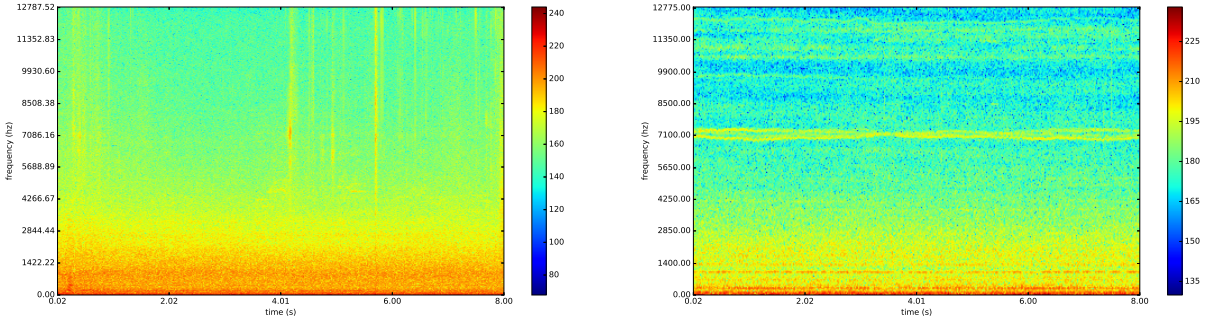
Our proposed simple architecture consists of nine stages contrary to previous approaches proposed in audio-related tasks (Table I), because rather than improving the performance, a more complex model easily leads to the overfitting problem. During training, we periodically checked the accuracy and loss with the testing dataset, then stopped training if the accuracy did not improve for three epochs of training. Eventually, we selected the model that showed the best accuracy. We shuffled the training dataset every epoch with a learning rate of 0.001 and a batch size of 128.

E. Classifier3: Recurrent Neural Network

The other popular DNN model, RNN, is designed to make use of past information to feedforward the network. They perform the same task repeatedly with memory, which represents the context of the information accumulated up to that moment. This memory component has the role of preventing the vanishing gradient problem that decays the influence of past data. Based on this idea, the long short-term memory (LSTM) design is commonly used for standard RNN through replacing simple neurons to LSTM memory blocks, which consist of several gates, such as a $tanh$ input gate, a forget gate to decide whether to remain, and an output gate to control which value is used to compute the output activation. Finally, the output of the LSTM memory block is computed as a multiplication of these gates.

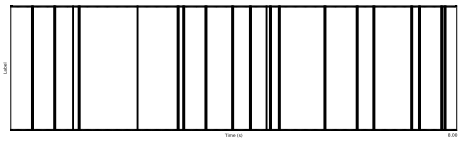
In this work, bi-directional LSTM-RNN with three layers and 300 LSTM blocks shows the best detection performance. Likewise, when training the CNN model, an early stopping strategy is used in RNN model training by periodically checking the accuracy and loss from the testing dataset. We stop training if it is not improved over 3 epochs of training, after which we retain the model that shows the best accuracy. We shuffle the training dataset every epoch and use a learning rate of 0.0005 and batch size of 64.

ISBN 978-0-9928626-7-1 © EURASIP 2017

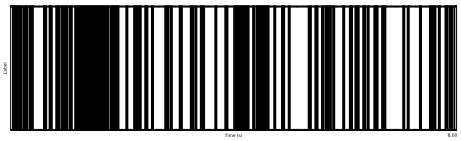


(a) Spectrogram of negative data (Freq: 0∼12k)

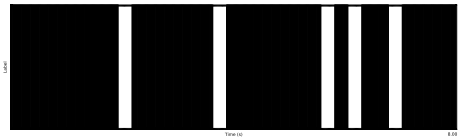
(b) Spectrogram of positive data (Freq: 0∼12k)



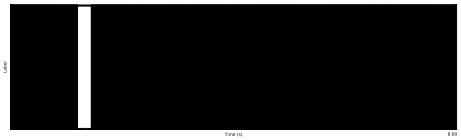
(c) Predicted detection label w.r.t time from GMM



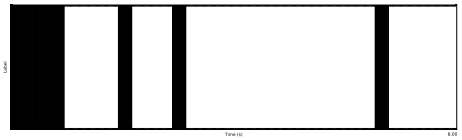
(d) Predicted detection label w.r.t time from GMM



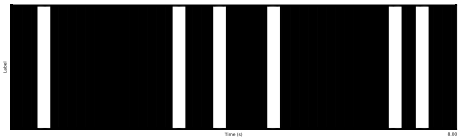
(e) Predicted detection label w.r.t time from CNN



(f) Predicted detection label w.r.t time from CNN



(g) Predicted detection label w.r.t time from RNN



(h) Predicted detection label w.r.t time from RNN

Fig. 1: Example: Spectrogram of negative and positive data and corresponding detection label from urban street (black area in Figure (c)∼(h) indicates predicted period during which drone exists)

III. Experiment

We evaluated our methodology by answering the following questions: (Q1) comparing the detection performance of three models, GMM, CNN, and RNN. (Q2) determining the detection performance for unseen types of data such as detecting different drone models or in different environments. (Q3) considering the required computing cost for application to real-time detection systems. All reported performance values are averaged across 10 evaluation results. We implemented the model in Python 2.7 with Scikit-learn 0.18, Librosa 0.4.3, and Tensorflow 0.12 on the following system: 4-core 2.6-GHz CPU, SSD, and GTX 1070.

A. Data description

Our training dataset is augmented by raw background sound and drone sound. The background sound consists of data from our own recording and from a public dataset [12], [13] and the drone sound was collected manually with two popular commercial drones – Phantom3 and Phantom4 from DJI. Our background sound data contains sounds from ordinary real-life situations with common noise such as chatting, car passing, and airplane noise with a total time of 677 seconds. Our drone sound data was recorded in a quiet outdoor place at a distance of 30m, 70m, and 150m for two types of behavior, hovering and approaching, with a total time of 64 seconds. Exact labeling of the drone sound was achieved by starting to record after the drone is activated, and stopping before deactivation. As a result of augmentation, the total audio time used for training is 9556 seconds.

TABLE II: Data Description

Data Type	Total time (s)	Description
Raw: background	677	Background audio used for augmentation
Raw: drone	64	Drone audio used for augmentation
Training: augmented	9556.68	augmented data for training
Testing: detection	151.06	measured in urban street for testing
Testing: unseen	1557	measured in outside with unseen type data

Note that we separate the training and testing datasets to enable us to strictly measure the performance, instead of the k-fold cross validation technique, which is commonly used to remedy a data shortage. Although an augmented dataset is useful for training, it has limited scope for completely

ISBN 978-0-9928626-7-1 © EURASIP 2017

mimicking a real dataset. We observe that the real dataset is not completely reproduced by augmentation, due to the complexity of audio characteristics and influence from the environment. Our testing dataset was collected on a real urban street, half of the data relating to a normal situation and the other half in proximity of a building construction site for 151 seconds with an equal amount of positive and negative data. Additionally, we built another testing dataset to measure detection performance for unseen types of data in training, such as unseen types of drones and background. This dataset includes other drone types with only positive label dataset, DJI Inspire and 3DR Solo, and other types of background such as near a highway or a very noisy road.

B. Testing: detection performance

We evaluated the detection performance of the drone using the proposed three models with the actual predicted period (Figure 1), precision, recall, F-Score, and accuracy (Figure 2). In our experiment, RNN achieves the best performance on the training datasets in terms of F-Score (RNN $>$ CNN $>$ GMM: $0.8009 > 0.6415 > 0.5232$). Our RNN also shows the most balanced detection performance between precision and recall (0.7953, 0.8066). It is evident that our data augmentation is meaningful to remedy the shortage of the drone training dataset through this high-detection performance. Our CNN model is reported as the second best model in terms of F-Score. We note that it remains difficult to decide whether our CNN model outperforms GMM. Our CNN and GMM show a distinctly different tendency according in terms of precision (CNN, GMM: $0.5346 < 0.9031$) and recall (CNN, GMM: $0.8019 > 0.3683$). Our CNN shows the tendency to predict data as positive rather than negative. On the contrary, GMM treats most of the data as negative, thus it shows lower recall but higher precision. However, considering our detection label result (Figure 1), GMM shows more accurate detection performance than statistics, but discontinuity in the positive prediction degrades the detection performance. This unstable consistency of positive prediction can be remedied by smoothing techniques. Therefore, in view of the operator, GMM can be regarded a more appropriate detection model to operate in practice. We also report the accuracy of these models, but do not consider it as important as the other measures.

Despite our diverse attempts we were unable to find CNN model architecture for previously proposed models. This could be attributed to the variation in audio data of the audio part unrelated to drone sound. In a real environment, we observe that the noticeable area affected by drone sound is small compared with the entire spectrogram image. Because of the fundamental mechanism of CNN, it is easily influenced by the other different areas of the spectrogram consisting of diverse environmental sound rather than focusing on drone sound only.

C. Testing: unseen types of data

The drawback of the machine-learning approach is the possibility of significant deterioration of detection performance when processing unlearned data. In this experiment, we aim to report degradation of detection performance for unseen types of data and improved understanding of the tendency of the proposed model. Our RNN still achieves the best performance in terms of F-Score (0.6984) with balanced precision and recall (0.5477, 0.9635). Interestingly, our report shows that the CNN model failed to classify the data, instead treating all data as positive. This misclassification could be caused by unseen highly noisy background sound that could not be distinguished from drone sound by the CNN model. According to this result, the CNN model is vulnerable to unseen noisy background data. Our GMM exhibits more accurate detection performance than CNN, but has a significantly decreased measure such that it would not be appropriate to operate in practice (0.3910 of F-Score).

Our experiment with unseen data provides additional insight on the tendency of the proposed models for GMM to predict data as negative and the other models based on deep neural networks to predict data as positive. In our experiment, even introducing additional training data does not improve the GMM model significantly; however, RNN can improve their precision performance through the diverse background training dataset. Above all, this experiment confirmed that it is essential to collect diverse types of data for the target environment.

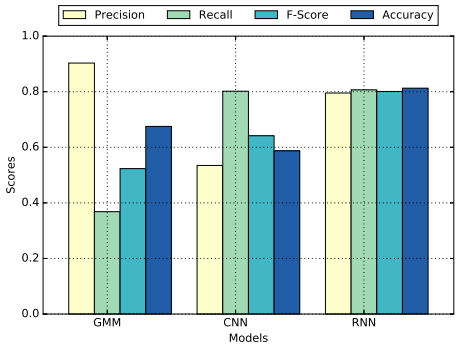


Fig. 2: Detection performance for testing dataset. The RNN achieves the best performance on the training datasets with F-Score of 0.8009

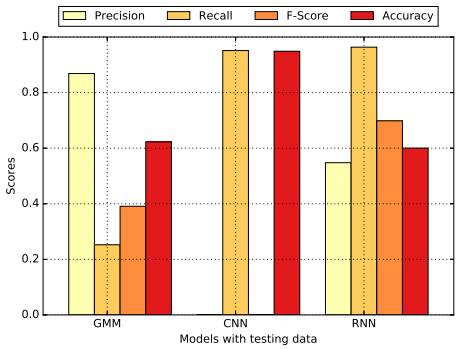


Fig. 3: Performance comparison for unseen type data. The RNN still achieves the best performance with F-Score of 0.6984. (this dataset only has positive label)

D. Required cost versus Detection performance

In practice, the other factor that should be considered to operate the detection system is the cost required by the

ISBN 978-0-9928626-7-1 © EURASIP 2017

detection model. The two main factors determining the required cost for operating this system in practice are the processing time and the amount of input data for prediction. Many previous studies overlooked the practical constraint for improved detection performance, but it indeed may dominate the form of the system in a real environment, such as the difference in processing time between the GMM and DNN models [14]. Statistically, we only measured the execution time for each process except for the additional time required to execute the Python program. According to the results, all three proposed models seem appropriate for application to real-time systems. The most time-consuming stage is feature engineering to create the MFCC vector, but it only takes 145 ms for a 1-minute audio clip. The execution time for data loading varies according to the target platform and the classification time does not adversely affect real-time system operation. For a fair comparison, we report the processing time for CNN and RNN without GPU usage.

However, we should note that the reported processing time is the minimal partial execution time only for each stage. In practice, the execution time of the detection algorithm varies according to the platform and is influenced to a greater extent by other costs related to program execution. Especially, if we plan to operate with a different programming language, then importing the Python program into a system programmed in another language would seriously deteriorate the execution time. Second, we note that models based on a deep neural network require a larger amount of data than GMM for optimal performance (240 ms $>$ 40 ms). This indicates that our actual initial detected time would increase as a function of the amount of input data. If we avoid importing the program or operating a low-performance embedded platform, the amount of input data can affect the initial detection time substantially.

TABLE III: Execution time for 1-minute audio clip

Process	Execution time (s)
Data read from wav format	0.0500
Feature engineering	0.1451
Prediction: GMM	0.0088
Prediction: CNN	0.0473
Prediction: RNN	0.0116

IV. Conclusion

This paper presents our binary classification model that uses audio data to detect the existence of a drone. We configured the parameters for GMM and the network for CNN and RNN for our model. Then, we evaluated their detection performance in terms of the F-Score and the required cost for application to real-time systems in practice. In our experiment, the RNN model showed the best F-Score (0.8009) with 240 ms of audio input data. Our experiment also confirmed that the use of data augmentation to synthesize raw drone sound with diverse background sounds can alleviate the shortage of drone training data.

The other main concern of our work was the influence on the detection performance of increasing drone distance. Because of practical constraints, we could not evaluate distances exceeding 150 m. In our experience, audio data recorded at distances further than 150 m do not display noticeable characteristics on the spectrogram with a naíve recording with a single microphone. This was attributed to the drone sound exhibiting weakened characteristics in the spectrogram because it is covered by background data. The usage of multiple microphones with Beamforming, a signal processing technique used for filtering to achieve directional signal transmission, is expected to increase the maximum detection distance of our model. The other interesting future work would be utilization of the Generative Adversarial Network (GAN) to remedy the shortage of drone sound training data.

References

[1] J. Mezei, V. Fiaska, and A. Molnár, "Drone sound detection," in *Computational Intelligence and Informatics (CINTI), 2015 16th IEEE International Symposium on*. IEEE, 2015, pp. 333–338.

[2] J. Busset, F. Perrodin, P. Wellig, B. Ott, K. Heutschi, T. Rühl, and T. Nussbaumer, "Detection and tracking of drones using advanced acoustic cameras," in *SPIE Security+ Defence*. International Society for Optics and Photonics, 2015, pp. 96 470F–96 470F.

[3] J. Mezei and A. Molnár, "Drone sound detection by correlation," in *Applied Computational Intelligence and Informatics (SACI), 2016 IEEE 11th International Symposium on*. IEEE, 2016, pp. 509–518.

[4] G. J. Mendis, T. Randeny, J. Wei, and A. Madanayake, "Deep learning based doppler radar for micro uas detection and classification," in *Military Communications Conference, MILCOM 2016-2016 IEEE*. IEEE, 2016, pp. 924–929.

[5] P. Nguyen, M. Ravindranatha, A. Nguyen, R. Han, and T. Vu, "Investigating cost-effective rf-based detection of drones," in *Proceedings of the 2nd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*. ACM, 2016, pp. 17–22.

[6] D. Barchiesi, D. Giannoulis, D. Stowell, and M. D. Plumbley, "Acoustic scene classification: Classifying environments from the sounds they produce," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 16–34, 2015.

[7] J. Pohjalainen, T. Raitio, and P. Alku, "Detection of shouted speech in the presence of ambient noise." in *INTERSPEECH*, 2011, pp. 2621–2624.

[8] A. Mesaros, T. Heittola, and T. Virtanen, "Tut database for acoustic scene classification and sound event detection," in *Signal Processing Conference (EUSIPCO), 2016 24th European*. IEEE, 2016, pp. 1128–1132.

[9] H. Zhang, I. McLoughlin, and Y. Song, "Robust sound event recognition using convolutional neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 559–563.

[10] E. Cakir, E. C. Ozan, and T. Virtanen, "Filterbank learning for deep neural network based polyphonic sound event detection," in *Neural Networks (IJCNN), 2016 International Joint Conference on*. IEEE, 2016, pp. 3399–3406.

[11] G. Parascandolo, H. Huttunen, and T. Virtanen, "Recurrent neural networks for polyphonic sound event detection in real life recordings," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 6440–6444.

[12] J. K. M.W.W. Grootel, T.C. Andringa, "DARES-G1: Database of Annotated Real-world Everyday Sounds," in *Proceedings of the NAG/DAGA Meeting 2009*, 2009.

[13] A. Rakotomamonjy and G. Gasso, "Histogram of gradients of time-frequency representations for audio scene classification," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 23, no. 1, pp. 142–153, 2015.

[14] S. Sigtia, A. M. Stark, S. Krstulović, and M. D. Plumbley, "Automatic environmental sound recognition: Performance versus computational cost," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 11, pp. 2096–2107, 2016.

ISBN 978-0-9928626-7-1 © EURASIP 2017