Towards joint sound scene and polyphonic sound event recognition

Helen L. Bear[1,2]*, Inês Nolasco*[1]*, and Emmanouil Benetos*[1,3]

[1]School of Electronic Engineering and Computer Science, Queen Mary University of London, UK
[2]Institute for Informatics, Technical University Munich, Germany [3]The Alan Turing Institute, UK
{h.bear,emmanouil.benetos}@qmul.ac.uk, i.nolasco@se17.qmul.ac.uk

Abstract

Acoustic Scene Classification (ASC) and Sound Event Detection (SED) are two separate tasks in the field of computational sound scene analysis. In this work, we present a new dataset with both sound scene and sound event labels and use this to demonstrate a novel method for jointly classifying sound scenes and recognizing sound events. We show that by taking a joint approach, learning is more efficient and whilst improvements are still needed for sound event detection, SED results are robust in a dataset where the sample distribution is skewed towards sound scenes.

Index Terms: Acoustic scene classification, sound event detection, computational sound scene analysis, CRNN.

1. Introduction

Computational sound scene analysis refers to the field of study investigating computational models and methods for making sense of soundscapes in urban, domestic and nature environments [1]. Core problems in the field include identifying the acoustic environment of an audio stream, this is acoustic scene classification (ASC) or sound scene recognition [2], and on detecting the sound events or sound objects within a scene, namely sound event detection (SED) [3]. ASC and SED are commonly considered as two separate tasks in understanding sound scenes, as can be demonstrated by the evolution of the field through the IEEE AASP Challenges in Detection and Classification of Acoustic Scenes and Events (DCASE) [4, 5, 6].

Polyphonic sound scenes are those containing multiple overlapping sound events, as opposed to *monophonic* sound scenes that do not contain event overlaps [4]. These can be background noise or foreground events where more than one sound source can be generating sounds at a single point in time. Polyphonic sound mixtures are challenging for recognising different sound sources and events, and this is the focus of this work. It has been suggested that sound event information can help acoustic scene classification, meaning with event classifications a-priori, the accuracy of scene classification increases [2]. Vice versa, with an accurate scene prediction the confidence of likely events in that scene increases [7]. In the latter case, SED can be described as *scene-dependent* or *scene-independent* [8] but given the variability of events in a scene, this description does not work in reverse for ASC, i.e. that a scene is dependent or not on any single event.

Prior work includes training separate models for ASC and SED, where models require tuning for each task, for example [9]. Typically for ASC, researchers use Convolutional Neural Network (CNN) models (for examples we refer the reader to review the submissions online for the DCASE 2018 ASC task [10]), since temporal dependencies are not considered important for ASC [11]. This differs to SED where researchers are most recently using Convolutional Recurrent Neural Networks (CRNNs) (for examples see Task 4 for [10]) where local time information improves detection accuracy (e.g. [8]).

However, we are yet to predict both of these tasks simultaneously with a single model which is the main contribution of this paper. When recognising environmental sounds, humans use prior knowledge of likely events in the scene and their prior experiences of the scenes to classify environments, as demonstrated by one such listening test in [12]. This, plus more evidence that context information such as a scene descriptor increases SED accuracy by machines [7], motivates this work to build a single recognition model by learning both scene and event data concurrently. To the best of our knowledge this is the first attempt to create a system for joint ASC and SED. With respect to prior work in sound scene analysis, this is a novel proof-of-concept which has the potential to optimise future ASC and SED systems whereby robust ASC and SED inputs are coupled before training a single model to predict both scenes and events jointly. In the SED literature, detection and recognition refer to the same problem. This is because SED systems are not evaluated just on spotting events (with onsets and offsets), but on both spotting the events and assigning a label to them. For clarity, in the rest of this paper, we use the term *classification* for matching class labels, *detection* for spotting events and *recognition* as both classification and detection.

For ASC and SED, deep learning models based on CNNs/CRNNs are achieving good recognition accuracy [5, 6, 10]. A summary from the recent DCASE 2018 challenge submissions [10] shows that for SED particularly, CRNN models are robust for sound event recognition. A CRNN is a neural network where the architecture includes both convolutional and recurrent layers [13]. SED recognition scores are, as a rule of thumb, lower than ASC. However, it is an unfair comparison as ASC is usually treated as a single-label classification task, whereas SED is evaluated as a recognition task involving both detection and classification. Whilst SED can benefit from the recurrent layers in an CRNN, real-world applications of SED involve overlapping sound events which requires multi-label classification. Robust sound event features are the log of mel spectrogram energies [14] as this compact representation more closely approximates human perception, when compared with for example a magnitude STFT spectrogram. Conversely for ASC, to predict one single label for an entire recording, the modeling of temporal dependencies is of lesser importance. Therefore, good features for ASC are temporally smoothed time-frequency representations (e.g. the approach in [15]).

Whilst ASC and SED as separate tasks benefit from different models and inputs, real-world audio streams include both scene and sound event data. As humans listening to a real-world

This work was funded under EPSRC grant EP/R01891X/1. EB is supported by a UK RAEng Research Fellowship (RF/128) and a Turing Fellowship. This work was supported by an NVIDIA GPU grant.

Table 1: *Proposed groupings of foreground sound event labels for each acoustic scene class.*

Scene	Sound Events
bus	clearthroat, cough, keys, laughter, phone, speech.
busystreet	bus-passby, doorclose, footsteps, key_lock, knock, laughter, motorbike, speech, running, wind.
office	chairs_moving, doorslam, drawer, keys, knock, laughter, switch, phone.
openairmarket	bag_rustle, bus-passby, cooking, footsteps, footsteps_on_grass, light_rain, money, speech, wind.
park	bus_passby, birdsong, footsteps_on_grass, gate, laughter, light_rain, phone, pushbike, speech, wind.
quietstreet	birdsong, footsteps, key_lock, light_rain, pushbike, wind.
restaurant	chairs_moving, cooking, doorclose, footsteps, laughter, speech.
supermarket	bag_rustle, checkout beeps, footsteps, money, switch, trolley.
tube	announcement, bag_rustle, footsteps, phone, slidingDoor_close, speech, train.
tubestation	announcement, footsteps, running, slidingDoor_close, speech, train.

acoustic scene, one uses knowledge of a scene to help limit our choices of likely events, or if a distinct event is prevalent, known to only be likely in limited scenes, our expectation of predicting a specific scene increases. Benefits of a single model include: only one model to design and train; there is no integration or linear pipelining of separate models for each task; and by using data which contains only likely events in each scene, synthesized from real-world samples, it should generalize to real-life data by learning the variation of events from multiple different scenes, and thus we can predict both ASC and SED concurrently in the evaluation stage. The rest of this paper is as follows: the proposed joint SED and ASC method, including data curation, is described in Section 2. Results are presented in Section 3 and conclusions are drawn in Section 4.

2. Method

2.1. Data preparation

For the proposed model, a new dataset is needed as to the best of our knowledge all publicly available prior datasets are for a single task, meaning annotations for both sound scenes and events are not available. Recordings are designed such that the foreground sound event classes are only those likely to be in the background sound scene in the real world. For example, on a quiet street one might hear birdsong, but it is unlikely to hear the beeps from a supermarket checkout. Background scenes are taken from the DCASE 2013 ASC challenge private test dataset [4]. For each of the ten scene classes, there are 20 recordings of 30 seconds long. These recordings may contain unannotated foreground sound events as they are real-world recordings. So to mitigate the risk of erroneous false positives in evaluation, the background loudness in the final recordings is reduced.

The first event samples are also taken from a DCASE challenge, this time from DCASE 2016 Task 2 ("Sound event detection in synthetic audio", focusing on office sounds) [5]. Additional sounds are sourced from `FreeSound.org`. Only recordings of isolated sound events are used, not event sequences. Classes with greater intra-class variation (such as the *footsteps* class in various shoe styles on different surface at different speeds has more samples than say, the *wind* class which varies mostly by its strength) have a greater number of sources to redress variation imbalance.

All are recorded as `.wav` files, no re-encoded files from lossy compression formats are collected. Any sampling rates other than 44100Hz are upsampled or downsampled to match the scene sampling rate of 44100Hz. All files are max-normalised using python's `soundfile` library. Any leading silences are stripped. All source recordings are real-world ones.

Table 2: *Number of isolated event recordings per event class. '*' classes are from DCASE 2016 Task 2; others are from `freesound.org`. #R is the total number of recordings and #T is the number of held-out test files.*

Sound Event	#R	#T	Sound Event	#R	#T
announcement	29	3	bag_rustle	37	4
birdsong	39	4	bus_passby	31	3
chairs_moving	35	3	checkout_beeps	35	3
clear_throat	20	2	cooking	35	3
cough*	20	2	doorclose	27	3
doorslam*	20	2	drawer*	20	2
footsteps	30	3	footsteps_on_grass	36	4
gate	30	3	keys*	20	2
key_lock	33	3	lake	38	4
knock*	20	2	laughter*	20	2
light_rain	28	3	money	34	3
motorbike	29	3	phone*	20	2
pushbike	30	3	running	20	2
sliding_door_close	29	3	speech*	20	2
switch*	20	2	trolley	28	3
train	25	3	wind	36	4

The total number of input event sources is tripled; for all foreground sound events a copy is produced with $+10dB$ relative to the source and a second duplicate with $-10dB$. This creates more variation in the synthetic scenes. The outcome of this sourcing is 824MB/5792.55sec (foreground) and 504MB/3000sec (background) data. For each of the 32 events there are at least 20 recordings from different sources. Certain DCASE 2016 event classes (alarm, keyboard, mouse, pen_drop, and printer) were removed as they are specific to the office scene. The new events selected were based on likely events in each background scene as listed in Table 1, grouped by the paired background scenes.

As much as possible, event labels are in at least two different scenes to reduce the model learning 'if event x occurs then it is always scene y'. This moves this work away from traditional closed set problems in ASC and SED towards real-world scenarios which are open set, thus an event can be occurring in multiple scenes [16]. The only exceptions to this are possible lake sounds in a park and trolley sounds in a supermarket. These are permitted due to the high likelihood of those sounds being associated with the scenes but have very low likelihood of presence in the other scenes.

To transform our collected sounds into scenes, `Scaper` [17] is used. `Scaper` is a python-based tool for synthesising

sound scenes with accompanying annotation files. For each of our ten background scene classes, there are ten unique locations for each class which each having one sample recording, with a total of 100 background recordings. Each background is used in 10 new sound scenes for a new set of 1000 sound scenes.

During synthesis, foreground sound events are added. Parameters allow the event pitch to be altered by a random value between -3 to $+3$ semitones before it is added to the mixture and the event duration can be stretched with multipliers randomly selected between 0.8 and 1.15. The event-to-background signal-to-noise ratio (SNR) is randomly assigned in the range -15 to 15 and all events for one scene are normally distributed throughout the 30 second scene. The number of events in a scene ranges from one to the total number of events in that background class plus one multiplied by three (e.g. scene class *bus* has six possible event types so its event range is one to 21, likewise, scene class office has eight possible events, so ranges from one to 27 events in a 30 second recording). Thus scenes with more options are more likely to be busier scenes. Each scene is permitted duplicate event classes, and duplicate events from the same source for maximum variability. Event polyphony level in all scenes is three, i.e. the maximum number of concurrent sound events within a 30 second recording is three and event choice is random from the scenes list of event options. Next, all resulting sound scenes are augmented with pitch shifting using the MuDA Python library [18]. In music signal analysis applications, this shift is often one or two semitones (e.g. [19]). Listening tests on the new sound scenes found that pitch could shift by six semitones up and down and the resulting scene remained realistic to human listeners. In order to create more variation in the dataset to support the recognition model to generalize well, each of the 1000 sound scenes are duplicated with two pitch shifts to treble the dataset, so the final sound scene and event dataset consists of 3000, 30 second `.wav` files with accompanying `JAMS` and annotation text files, a total size of 6.4GB. All synthesised sound scenes are mono and available publicly[1].

2.2. Feature extraction

Features are extracted for every sound scene created as the log of mel-spectrogram energies using 128 mel bands, hop length of 512, and an STFT size of 2048. Five folds of data are structured as per the background scene divisions from DCASE 2013 to ensure scene sources are not duplicated between train and test folds. After dividing the data into training and test folds, for each data sample we subtract the training fold mean and divide by the standard deviation for the fold. Next, a copy of each sound scene feature is smoothed over time as per [15], as this has been shown to be effective in ASC. Thus, the feature set contains one log-mel-specctrogram (useful for SED) and one temporally smoothed log-mel-spectrogram (useful for ASC). The final step stacks each 2D event feature with the 2D scene feature for each original recording into a 3D input such that the final training data are all two-channel inputs.

In summary we have 1292 frames per 30second recording, 2100 training, 300 validation, and 600 test samples per fold.

2.3. Sound scene and sound event recognition

Recognition is undertaken by first dividing the 3000 feature files into 600 for test, 2100 for training, and of the training files 300 are held out for validation using stratified five-fold cross validation.Meaning, 20% of data is for each test fold, and a quarter of the training data is held out for CRNN validation during training. There is no source/device/scene location/event overlap between train and test folds, consistent with the divisions in the original 2013 DCASE ASC task. The CRNN model architecture is three convolutional layers with max pooling, batch normalization, and dropout layers, followed by an LSTM layer, a fully connected layer, a further batch normalization and finally a Sigmoid activation layer in place of Softmax for multi-label classification with a binary-cross entropy loss function. Model parameters are detailed in Table 3. For the ASC baseline the class predictions are binarised with a global threshold of 0.9 before majority voting (as per prior ASC work such as [20]). For the SED baseline we use the sed_eval toolkit [21] to measure the segment-based error rates (ER) and F1 score. The baseline models are ASC and SED models trained separately as single tasks against which we can compare the the proposed joint network. All are implemented in Tensorflow with Keras. For the joint recognition task we combine the measuring requirements with a minimum global threshold, tuned for each ASC/SED task.

Table 3: *CRNN structure and parameters. Adam optimiser is used with* $LR = 0.001$, $beta_1 = 0.9$, $beta_2 = 0.999$, $epsilon = None$, $decay = 0.0$, $amsgrad = False$.

Layer	params
Convolutional	filters=64, kernel=(3,3)
MaxPooling2D	pool_size=(3,3), strides=2, padding='same'
BatchNormalization	
Dropout	prob_drop_conv=0.25
Convolutional	filters=128, kernel=(3,3)
MaxPooling2D	pool_size=(3,3), strides=2, padding='same'
Dropout	prob_drop_conv=0.25
Convolutional	filters=256, kernal=(2,2)
MaxPooling2D	pool_size=(2,2), strides=2, padding='same'
BatchNormalization	
Dropout	prob_drop_conv=0.25
Reshape	shape=(256,-1)
LSTM	filters=256, input_shape=(1292, 128,channels)
Dense	filters=256, activation='relu'
Dropout	prob_drop_hidden=0.5
BatchNormalization	
Dense	nb_classes, activation='sigmoid'

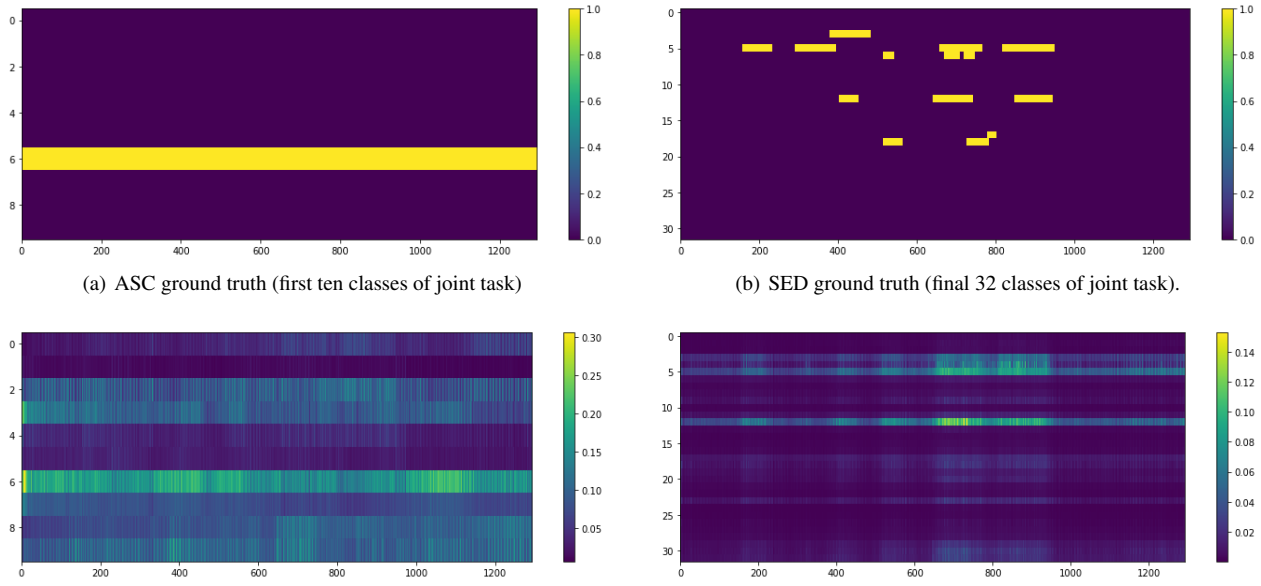
Each feature vector input is the extracted feature matrices from each recording. All feature matrices are of equal length (one column for each time point) by 128 feature parameters. The corresponding label for each column is a N-hot vector of 43 binary values. The first ten elements represent scene classes and the remaining 32 are the sound event classes.

3. Results

Conventional metrics for sound event detection (SED) are the segment-based Precision, Recall, the harmonic mean of these, the F1 score, and the Error Rate [21]. Predictions are in the form of a binary vector; for each frame of the test samples, we produce a one hot vector. Measuring ASC is straightforward using classification accuracy (Acc) on scenes with majority voting on all the frames grouped by test recording sample as per many prior works [22].

Care is taken in selecting metrics for SED due to the class imbalance between events as selections are random per scene creation in `Scaper`. We use two metrics from the SED_eval toolkit [21], segment-based Error Rate (ER) (default segment size of one second is used) and the F1 score, this is com-

[1] `http://doi.org/10.5281/zenodo.2565309`



(a) ASC ground truth (first ten classes of joint task)

(b) SED ground truth (final 32 classes of joint task).

(c) ASC predictions.

(d) SED predictions.

Figure 1: *An example test sample from source to prediction.*

mon practice for comparison with other systems such as in the DCASE 2017 task 3. All code for feature extraction, recognition, and evaluation tasks is online[2].

Table 4: *Results of separate ASC & SED models, compared with the proposed joint task model. MV = Majority Voting.*

	ASC	SED	
	MV Acc	F1	ER
Separate models	$0.99\sigma 0.01$	$28.86\%\sigma 2.67$	$0.86\sigma 0.01$
Joint model	$0.98\sigma 0.03$	$13.73\%\sigma 8.29$	$1.00\sigma 0.05$

All results for ASC and SED are presented in Table 4 and in Fig 1 there are four plots for an example test sample. The respective ground truths are shown in Figs 1a and b for the ten ASC and 32 SED classes respectively and Figs 1c and d show the predictions for this test sample with the joint-task model. The ASC prediction is distinctive but in this complex, polyphonic test sample, we see that the SED predictions are strong for timings (shown by lighter coloured frames), but where there are more than three classes in the ground truth, these are not so well detected. Also short event predictions are darker in Fig 1d.

First results in Table 4 show robust ASC scores which we attribute to the fact that our dataset is built of real-world recordings with synthesised additions of specific events. With the joint model there was no significant variation in mean ASC performance. But the joint models (for all folds) trained in approximately half the epochs of the separate ASC task, which supports the hypothesis that scene-specific event classes help ASC.

There is a decrease in SED F1 and Error Rate when comparing the joint model with the separate SED model; we attribute this decrease to the skew in training samples per class once scenes and events are trained jointly. For example, our model is trained with per frame labels thus there are $30 \times 1292 = 38760$ training samples per scene, yet some event classes have as few as 140 training samples due to their short duration and sporadic appearance in each scene. We also attribute the greater standard deviation to this skew for the joint model SED F1, shown in Table 4.

Reviewing class based measures, the events which perform poorly are either very short (e.g. door slam) or could be part of the background noise (e.g. motorbike). It is possible to artificially inflate the SED scores by increasing the segment-size (to improve detection accuracy by increasing recall, whilst risking extra insertion errors) or alter the prediction threshold for binarisation, but neither of these approaches gain more meaningful results thus we present the realistic SED results. These results demonstrate the efficacy of the proposed joint model by comparison to separate ASC and SED tasks. To optimise the joint approach further one seeks a number of training epochs which does not overfit ASC but improves SED further than presented here.

4. Conclusions

This paper presents a novel approach to jointly perform ASC and SED with a single model. In doing so, the work has produced a new publicly available dataset for sound scene and sound event recognition, with related scene and event classes, as well as onset and offset annotations for sound events.

The new novel joint recognition model harnesses learning from the best of inputs used in ASC and SED but developing a model that will generalise well to two related but distinct tasks is a difficult undertaking and here we have shown promising results. These results show that a joint model can learn more efficiently. It does create future work for SED. For example: discovery of the right number of training epochs to improve SED without overfitting for ASC, altering the feature inputs to highlight distinct sound events and, revising the model architecture, therefore there is scope to build on and improve this work. Model performance has been quantified using metrics as benchmarks for the researchers interested in ASC and SED.

[2] `https://github.com/drylbear/jointASCandSED`

5. References

[1] T. Virtanen, M. D. Plumbley, and D. Ellis, Eds., *Computational analysis of sound scenes and events*. Springer, 2018.

[2] 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.

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

[4] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events," *IEEE Transactions on Multimedia*, vol. 17, no. 10, pp. 1733–1746, 2015.

[5] A. Mesaros, T. Heittola, E. Benetos, P. Foster, M. Lagrange, T. Virtanen, and M. D. Plumbley, "Detection and classification of acoustic scenes and events: Outcome of the dcase 2016 challenge," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 26, no. 2, pp. 379–393, 2018.

[6] T. Virtanen, A. Mesaros, T. Heittola, A. Diment, E. Vincent, E. Benetos, and B. M. Elizalde, *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*. Tampere University of Technology. Laboratory of Signal Processing, 2017.

[7] T. Heittola, A. Mesaros, A. Eronen, and T. Virtanen, "Context-dependent sound event detection," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2013, no. 1, p. 1, 2013.

[8] E. Cakir, G. Parascandolo, T. Heittola, H. Huttunen, T. Virtanen, E. Cakir, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 25, no. 6, pp. 1291–1303, 2017.

[9] H. Phan, L. Hertel, M. Maass, P. Koch, R. Mazur, and A. Mertins, "Improved audio scene classification based on label-tree embeddings and convolutional neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1278–1290, 2017.

[10] Detection and Classification of Acoustic Scenes and Events workshop hosted by University of Surrey, "All challenge submissions and results," 2018. [Online]. Available: http://dcase.community/challenge2018/

[11] V. Peltonen, J. Tuomi, A. Klapuri, J. Huopaniemi, and T. Sorsa, "Computational auditory scene recognition," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 2. IEEE, 2002, pp. II–1941.

[12] S. Chu, S. Narayanan, and C.-C. J. Kuo, "Environmental sound recognition with time–frequency audio features," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1142–1158, 2009.

[13] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1.

[14] R. Lu, Z. Duan, and C. Zhang, "Multi-scale recurrent neural network for sound event detection," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 131–135.

[15] V. Bisot, R. Serizel, S. Essid, and G. Richard, "Feature learning with matrix factorization applied to acoustic scene classification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1216–1229, 2017.

[16] H. Bear and E. Benetos, "An extensible cluster-graph taxonomy for open set sound scene analysis," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018)*, November 2018, pp. 183–187.

[17] J. Salamon, D. MacConnell, M. Cartwright, P. Li, and J. P. Bello, "Scaper: A library for soundscape synthesis and augmentation," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2017, pp. 344–348.

[18] B. McFee, E. Humphrey, and J. Bello, "A software framework for musical data augmentation," in *16th International Society for Music Information Retrieval Conference*, ser. ISMIR, 2015.

[19] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on speech and audio processing*, vol. 10, no. 5, pp. 293–302, 2002.

[20] T. Nguyen and F. Pernkopf, "Acoustic scene classification using a convolutional neural network ensemble and nearest neighbor filters," in *Workshop on Detection and Classification of Acoustic Scenes and Events*, 2018.

[21] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, p. 162, 2016.

[22] A. Mesaros, T. Heittola, and T. Virtanen, "Acoustic scene classification: An overview of dcase 2017 challenge entries," in *2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC)*, Sep. 2018, pp. 411–415.