CROSS-TASK LEARNING FOR AUDIO TAGGING, SOUND EVENT DETECTION AND SPATIAL LOCALIZATION: DCASE 2019 BASELINE SYSTEMS

Technical Report

Qiuqiang Kong, Yin Cao, Turab Iqbal, Wenwu Wang, Mark D. Plumbley

Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey, UK
{q.kong, yin.cao, t.iqbal, w.wang, m.plumbley}@surrey.ac.uk

ABSTRACT

The Detection and Classification of Acoustic Scenes and Events (DCASE) 2019 challenge focuses on audio tagging, sound event detection and spatial localisation. DCASE 2019 consists of five tasks: 1) acoustic scene classification, 2) audio tagging with noisy labels and minimal supervision, 3) sound event localisation and detection, 4) sound event detection in domestic environments, and 5) urban sound tagging. In this paper, we propose generic cross-task baseline systems based on convolutional neural networks (CNNs). The motivation is to investigate the performance of a variety of models across several audio recognition tasks without exploiting the specific characteristics of the tasks. We looked at CNNs with 5, 9, and 13 layers, and found that the optimal architecture is task-dependent. For the systems we considered, we found that the 9-layer CNN with average pooling after convolutional layers is a good model for a majority of the DCASE 2019 tasks.

Index Terms— Cross-task, convolutional neural networks, audio tagging, sound event detection, sound event localisation.

1. INTRODUCTION

Sound carries a large amount of information that can be utilised in a number of applications, such as information retrieval [1], abnormal event detection [2] and autonomous cars [3]. The Detection and Classification of Acoustic Scenes and Events (DCASE) is a series of challenges aimed at developing sound classification and detection systems. The first DCASE challenge was organized in conjunction with WASPAA 2013 [4], and is now an annual challenge and workshop [5, 6, 7]. The most recent DCASE challenge is the DCASE 2019 [7]. In this paper, we propose a number of cross-task baseline systems for the DCASE 2019 challenge. These systems are designed to be applicable to a wide variety of tasks.

Previous work in DCASE challenges mainly focus on particular tasks [8, 6]. That is, the recognition systems are built for individual tasks. However, a system trained on one task may not reflect its generalisation ability on other tasks. The motivation of this paper is to investigate how generic systems perform across several task. We call such systems *cross-task* systems. As convolutional neural networks (CNNs) have achieved state-of-the-art performance in many DCASE challenge tasks [8, 5, 6], we investigate different CNN architectures as possible cross-task systems.

The DCASE 2019 challenge consists of five tasks. In Task 1, *Acoustic scene classification (ASC)* [6], the aim is to classify audio recordings, recorded in a public area, into one of several predefined acoustic scene classes. This task includes three subtasks: a matching device ASC subtask, a mismatching device ASC subtask and an open set ASC subtask. In Task 2, *Audio tagging with noisy labels and minimal supervision* [9], the aim is to predict the tags of audio recordings while utilising a small number of manually-verified labels and a much larger number of noisy labels. In Task 3, *Sound event localization and detection (SELD)* [10], the aim is to predict the presence or absence of sound events, their onset and offset times, and their spatial locations in azimuth and elevation. In Task 4, *Sound event detection in domestic environments* [11], the aim is to predict the presence or absence and the onset and offset times of sound events. Task 4 provides weakly-labelled data, unlabelled data and simulated strongly-labelled data for training. In Task 5, *Urban sound tagging* [12], the aim is to predict urban sound tags of audio recordings recorded in New York. Multiple annotations are provided for each recording, and they do not always agree. Dataset of Task 1, 2 and 5 are weakly labelled. That is, only tags of the audio recording are provided, but not the onset and offset times of the tags. Dataset of Task 3 and 4 are strongly labelled, where the onset and offset of sound events are labelled.

Recently, neural-network based methods such as fully-connected neural networks [13], convolutional neural networks (CNNs) [14, 15, 16] and recurrent neural networks (RNNs) [17] have been used for sound classification and detection, and have achieved state-of-the-art performance in recent DCASE tasks [5, 6]. The CNN based methods usually take log mel spectrogram of audio recordings as input and output the presence probability of sound events in either frame-level or clip-level. However, many of these neural network based methods are designed for particular tasks. It is not clear whether a system designed for one task can generalise well to other tasks.

The motivation of this paper is to investigate the performance of generic systems across several tasks, instead of focusing on particular tasks. In this paper, we investigate cross-task baseline systems based on CNNs starting from our deep neural network (DNN) and CNN based baseline systems for the DCASE 2016 [13] and 2018 [15] challenges. In addition to previous work [13, 15], we investigate deeper CNN architectures with up to 13 layers, including 5-layer, 9-layer and 13-layer CNNs. We also investigate different pooling strategies after the convolutional layers, including average pooling and max pooling. We compare the performance of different CNN architectures across all five DCASE 2019 challenge tasks. We release the source code of our cross-task systems as baselines for future research.

This paper is organised as follows. Section 2 introduces the CNN architectures. Section 3 presents experimental results. Section 4 concludes the paper.

2. CONVOLUTIONAL NEURAL NETWORKS

CNNs have been widely used in computer vision and have achieved state-of-the-art performance in several tasks such as image classification [18]. A conventional CNN consists of several convolutional layers, where each convolutional layer consists of filters to convolve with the output from the previous convolutional layer. The filters can capture local patterns of the input feature maps, such as edges in lower layers and profiles of objects in higher layers [18]. CNNs have been applied to many audio classification and sound event detection tasks using inputs such as log mel spectrogram [8, 6]. However, the choice of architecture is usually task-dependent. There is a lack of research investigating the performance of cross-task CNN systems.

We investigate three kinds of CNNs with different depths: a 5-layer CNN, a 9-layer CNN and a 13-layer CNN. The 5-layer CNN is similar to AlexNet [19], which consists of 4 convolutional layers with a kernel size of 5×5. The later VGG network [18] was proposed to decompose the 5×5 kernel to a convolutional block consisting of two cascaded convolutional layers with 3×3 kernels: this inspired the 9-layer and 13-layer CNNs used in this paper with 4 and 6 convolutional blocks. For all the architectures, batch normalisation [20] is applied after each convolutional operation to speed up and stabilise the training. The ReLU function [18] is used as a non-linearity after batch normalisation. Average pooling or max pooling with a size of 2×2 is applied after each convolutional block to reduce the feature map size. As the 9-layer CNN perform better than the 5-layer and 9-layer CNNs in experiment, we compare the 9-layer CNN with average pooling named CNN9-avg with the 9-layer CNN with max pooling named CNN9-max. To help the systems robust to frequency shift of sound events, we average out the frequency information in the feature maps of the last convolutional layer. For audio tagging tasks with weakly labelled data, the information over time frames are maxed out, which is designed to select the predominant information over time steps for clip-level classification. Finally, a fully connected layer is applied to predict the presence of sound events either at the clip-level or frame-level. Table 1 summaries the CNN architectures used in this paper. For example, 5×5 @ 64 indicates a convolutional layer with a kernel size of 5×5 and an output feature maps number of 64. The parameters number (PN) of models are shown at the bottom of each column in Table 1. For classification tasks, softmax nonlinearity is applied and cross entropy (CE) loss l_{CE} is used for training the network:

Table 1: CNN architectures.

CNN5	CNN9	CNN13
Log-mel spectrogram		
$\begin{pmatrix} 5 \times 5 \text{ @ } 128 \\ \text{BN, ReLU} \end{pmatrix}$	$\begin{pmatrix} 3 \times 3 \text{ @ } 64 \\ \text{BN, ReLU} \end{pmatrix} \times 2$	$\begin{pmatrix} 3 \times 3 \text{ @ } 64 \\ \text{BN, ReLU} \end{pmatrix} \times 2$
2×2 Pooling		
$\begin{pmatrix} 5 \times 5 \text{ @ } 128 \\ \text{BN, ReLU} \end{pmatrix}$	$\begin{pmatrix} 3 \times 3 \text{ @ } 128 \\ \text{BN, ReLU} \end{pmatrix} \times 2$	$\begin{pmatrix} 3 \times 3 \text{ @ } 128 \\ \text{BN, ReLU} \end{pmatrix} \times 2$
2×2 Pooling		
$\begin{pmatrix} 5 \times 5 \text{ @ } 256 \\ \text{BN, ReLU} \end{pmatrix}$	$\begin{pmatrix} 3 \times 3 \text{ @ } 256 \\ \text{BN, ReLU} \end{pmatrix} \times 2$	$\begin{pmatrix} 3 \times 3 \text{ @ } 256 \\ \text{BN, ReLU} \end{pmatrix} \times 2$
2×2 Pooling		
$\begin{pmatrix} 5 \times 5 \text{ @ } 512 \\ \text{BN, ReLU} \end{pmatrix}$	$\begin{pmatrix} 3 \times 3 \text{ @ } 512 \\ \text{BN, ReLU} \end{pmatrix} \times 2$	$\begin{pmatrix} 3 \times 3 \text{ @ } 512 \\ \text{BN, ReLU} \end{pmatrix} \times 2$
PN: $4,304,320$	PN: $4,686,144$	2×2 Pooling
		$\begin{pmatrix} 3 \times 3 \text{ @ } 1024 \\ \text{BN, ReLU} \end{pmatrix} \times 2$
		2×2 Pooling
		$\begin{pmatrix} 3 \times 3 \text{ @ } 2048 \\ \text{BN, ReLU} \end{pmatrix} \times 2$
		PN: $75,477,312$

$$l_{\text{CE}}(\mathbf{p}, \mathbf{y}) = \sum_{k=1}^{K} y_k \ln p_k \tag{1}$$

where $\mathbf{y} = (y_1, ..., y_K) \in \{0, 1\}^K$ is the clip-level or frame-level target and K is the number of sound classes. The prediction $\mathbf{p} = (p_1, ..., p_K) \in [0, 1]^K$ is the predicted probability of sound classes. For audio tagging and polyphonic sound event detection (SED) tasks, sigmoid nonlinearity is applied. Binary cross entropy (BCE) loss l_{BCE} is used for training the network together with the sigmoid output:

$$l_{\text{BCE}}(\mathbf{p}, \mathbf{y}) = \sum_{k=1}^{K} \left[y_k \ln p_k + (1 - y_k) \ln (1 - p_k) \right]. \tag{2}$$

For the localisation task (Task 3), the target is to predict both the occurrence of sound events and their corresponding azimuth and elevation. We denote the azimuth and elevation target of the k-th sound event as $y_k^{(\text{azi})}$ and $y_k^{(\text{ele})}$. Similarly, the azimuth and elevation of output is denoted as $o_k^{(\text{azi})}$ and $o_k^{(\text{ele})}$. The detection and localisation network is jointly trained using the following loss function:

$$\text{loss} = l_{\text{BCE}}(\mathbf{p}, \mathbf{k}) + \lambda \sum_{k=1}^{K} y_k \left(\left| o_k^{(\text{azi})} - y_k^{(\text{azi})} \right| + \left| o_k^{(\text{ele})} - y_k^{(\text{ele})} \right| \right) \tag{3}$$

where λ is a weight to balance the SED loss and the localisation loss. We only calculate the MAE between prediction and target in the time steps where target y_k is true for each sound class.

3. EXPERIMENTS

We apply the described CNNs to the DCASE 2019 tasks. Table 2 shows the statistics of the DCASE 2019 tasks. Audio recordings from all tasks are resampled to 32 kHz which contains most energy. Task 1, 2, 4, 5 are trained with clip-level weak labels. Task 3 and the subset of Task 4 with synthetic data are trained with frame-level strong labels. A short-time Fourier transform (STFT) with a Hanning window size of 1024 samples and a hop size of 500 samples is used to extract the spectrogram, so that there are 64 frames in one second. The number 64 is chosen because it is divisible by the power of two which is convenient for the pooling operations in CNNs. Mel filter banks with 64 bins and cut-off frequencies of 50 Hz to 14 kHz are applied on the spectrogram. Then, a logarithm operation is applied to obtain the log-mel spectrogram. The baseline systems for the five tasks in the DCASE 2019 challenge are implemented with Python and PyTorch. The source code is released on github[1][2][3][4][5].

3.1. Task 1, Acoustic scene classification

Acoustic scene classification [6] is a task to classify a test recording into one of the provided predefined classes that characterises the

[1] https://github.com/qiuqiangkong/dcase2019_task1

[2] https://github.com/qiuqiangkong/dcase2019_task2

[3] https://github.com/qiuqiangkong/dcase2019_task3

[4] https://github.com/qiuqiangkong/dcase2019_task4

[5] https://github.com/qiuqiangkong/dcase2019_task5

Table 2: Statistics of DCASE 2019 challenge tasks.

	Task type	Duration	Classes num.	Channels	Sample rate	Label type	Loss
Task 1	Classification	~47.2 h	10	1 or 2	44.1 kHz or 48 kHz	Weak	CE (clip)
Task 2	Tagging	~90.5 h	80	1	44.1 kHz	Weak	BCE (clip)
Task 3	SED + DOA	~6.7 h	11	4	48 kHz	Strong	BCE + MSE (frame)
Task 4	SED	~53.3 h	10	1 or 2	44.1 kHz	Weak, strong	BCE (clip, frame)
Task 5	Tagging	~7.8 h	29	1	44.1 kHz	Weak	BCE (clip)

Table 3: Development accuracy of Task 1.

	SUBTASK A	SUBTASK B				SUBTASK C		
Model	Dev. A	Dev. B	Dev. C	Avg. (B,C)	Dev. A	Dev. A	Unknown	Avg.
Baseline [6]	0.625	0.396	0.431	0.414	0.619	0.542	0.431	0.487
CNN5-avg	0.698	0.456	0.522	0.489	**0.691**	0.581	0.481	**0.531**
CNN9-avg	**0.703**	**0.533**	0.548	0.541	0.686	0.596	0.432	0.514
CNN9-max	0.686	0.526	**0.567**	**0.547**	0.673	0.569	**0.487**	0.528
CNN13-avg	0.614	0.448	0.498	0.473	0.658	**0.616**	0.249	0.433

environment in which it was recorded. There are 10 sound classes recorded in 12 European countries such as “airport” and “metro station”. This task includes three subtasks: 1) ASC with matched devices, where the testing data are recorded from the same device as the training data, 2) ASC with mismatched recording devices, where the testing data are recorded using different devices as the training data, and 3) open set ASC, where part of the testing data is not encountered in the training data. For each subtask, the evaluation criterion is classification accuracy, which is obtained by averaging the class-wise accuracy of all sound classes.

Table 3 shows the performance of different CNNs. In Subtask A, the CNN9-avg achieves an accuracy of 0.703, outperforming the 5-layer and 13-layer CNN. In Subtask B, the CNN9-avg and CNN9-max achieve accuracy of 0.541 and 0.547 in the mismatching devices, outperforming the 5-layer and 13-layer CNN. This indicates that the 9-layer CNNs have good performance in classifying mismatching devices. In Subtask C, the 5-layer CNN achieves an average accuracy of 0.531, outperforming the 9-layer and 13-layer CNN.

3.2. Task 2, Audio tagging with noisy labels and minimal supervision

Audio tagging with noisy labels and minimal supervision [9] is a task to predict multiple labels of audio clips. The dataset consists of: 1) a small set of curated data from the Freesound Datasets (FSD) [21], which are manually verified and have variable duration from 0.3 s to 30 s; 2) a large set of noisy data from the Yahoo Flickr Creative Commons 100M dataset (YFCC) [22], which are not manually verified and may contain incorrect labels. Following [9], label-weighted label-ranking average precision (lwLRAP) is used for evaluating the performance of the designed systems.

As there is no official validation set provided, we split the development data to four folds for validation. The systems are trained on fold 2, 3 and 4 and validated on fold 1. Long audio recordings are split into 5-second segments. Short audio recordings are padded to 5-second segments with repeated pattern. Each segment inherit the tag of the audio recording. In Table 3.2, the capital letter A, B indicates training on curated FSD and noisy YFCC subset, respectively. A + B indicates training on both FSD and YFCC dataset. We evaluate the lwLRAP on the curated and noisy subset. Table 3.2 shows that the CNN9-avg trained with A + B achieves lwLRAP of 0.714 and 0.635 on curated and noisy data, outperforming the 5-layer, 13-layer CNN architectures.

Table 4: Development lwLRAP of Task 2.

	A		B		A + B	
Model	Curated	Noisy	Curated	Noisy	Curated	Noisy
CNN5-avg	0.806	0.246	0.393	0.613	0.669	0.616
CNN9-avg	**0.822**	**0.261**	**0.420**	**0.629**	**0.714**	**0.635**
CNN9-max	0.815	0.256	0.401	0.620	0.707	0.629
CNN13-avg	0.768	0.215	0.387	0.592	0.686	0.601

3.3. Task 3, Sound event localization and detection

Sound event localisation and detection (SELD) [10] is a task to detect the onset and offset time of sound events and their directions-of-arrival (DOAs) in azimuth and elevation angles. The development set provides 400 1-minute audio recordings. There are two formats of recording including 4-channel first-order-ambisonic (FOA) and four-channel directional microphone recordings from a tetrahedral array. The impulse responses are collected from five indoor locations at 504 unique combinations of azimuth-elevation-distance. The isolated sound events are from DCASE 2016 Task 2. The F-score and error rate are calculated in one-second segments for evaluating SED. DOA error and frame recall are used for evaluating localisation.

We extract the log-mel spectrogram of the 4-channel FOA as input to the CNNs. Different from the baseline system [10] using additional phase information as input, we only use the magnitude of log-mel spectrogram as input. In training, audio recordings are split into overlapped 10-second segments. The loss function in equation (3) is used for training. The evaluation follows [10]. Table 5 shows the performance of the CNN systems. The 5-layer and 9-layer CNNs achieve similar results in SED and localisation, outperforming the

Table 5: Development performance of Task 3.

	Error rate	F1 score	DOA error	Frame recall	SELD score
Baseline [10]	0.34	79.9%	**28.5°**	**85.4%**	-
CNN5-avg	0.33	**80.7%**	54.4°	77.0%	0.263
CNN9-avg	**0.32**	80.5%	44.0°	77.1%	**0.248**
CNN9-max	0.34	79.4%	45.6°	76.3%	0.260
CNN13-avg	0.42	72.8%	42.8°	71.4%	0.303

Table 6: Development performance of Task 4.

	Event-based F1	segment-based F1	Audio tagging mAP
Baseline [11]	23.5%	54.7%	-
CNN5-avg	18.0%	59.9%	0.765
CNN9-avg	20.0%	58.6%	0.778
CNN9-max	**24.1%**	**63.0%**	**0.791**
CNN13-avg	17.0%	58.3%	0.740

13-layer CNN. A DOA error of 42.8° using the 13-layer CNN.

3.4. Task 4, Sound event detection in domestic environments

Sound event detection in domestic environments [11] is a task to detect the onset and offset time steps of sound events in domestic environments. The datasets are from AudioSet [16], FSD [21] and SINS dataset [23]. The aim of this task is to investigate whether real but weakly annotated data or synthetic data is sufficient for designing SED systems. There are 1578 real audio recordings with weak labels, 2045 synthetic recordings with strong labels, and 14412 unlabelled in-domain recordings in the dataset. Audio recordings are 10 seconds in duration and consist of polyphonic sound events from 10 sound classes.

Table 6 shows the performance of CNN systems using the real weakly-labelled subset for training. CNN9-max achieves an event-based F1 of 24.1%, a segment-based F1 of 63.0% and an audio tagging mAP of 0.791, outperforming the other CNN systems. Different from other tasks, 2×2 max pooling outperforms 2×2 average pooling in this task. In addition, we observe that using only the real weakly labelled data for training achieves an event-based F1 of 24.1%, outperforming the result of using synthetic weakly and strongly labelled data for training of 13.0% and 13.4%, respectively.

3.5. Task 5, Urban sound tagging

Urban sound tagging [12] is a task to predict the presence or absence of different type of sounds recorded in New York City. This task has established a set of coarse-grained and fine-grained classes. The coarse-grained and fine-grained tags consist of 7 and 23 sound classes under a hierarchical taxonomy. There are 2351 10-second audio recordings for development. Each audio recording is annotated by one or more annotators. The label of a single audio recording can be different depending on the annotator. We apply an OR function to aggregate the labels from different annotators. That is, an audio recording for a sound class is positive if at least one annotator labels it as positive. Different from the baseline system [12] trained on AudioSet features [16], we train the CNNs on the log mel spectrogram from scratch. Table 7 shows that CNN9-avg achieves fine-grained and coarse-grained micro AUPRCs of 0.672 and 0.782, outperforming the 5-layer, 13-layer CNNs.

In the final submission, we also provided a 13 layers CNN model pretrained on AudioSet as a second submission system.

Table 7: Development performance of Task 5.

	FINE-GRAINED			COARSE-GRAINED		
	Micro AUPRC	Micro F1	Macro AUPRC	Micro AUPRC	Micro F1	Macro AUPRC
Baseline	0.672	0.502	0.427	0.742	0.507	0.530
CNN5-avg	0.659	**0.395**	**0.451**	0.765	**0.552**	0.570
CNN9-avg	**0.672**	0.371	0.433	**0.782**	0.519	**0.628**
CNN9-max	0.668	0.337	0.457	0.765	0.493	0.555
CNN13-avg	0.637	0.368	0.428	0.742	0.491	0.333

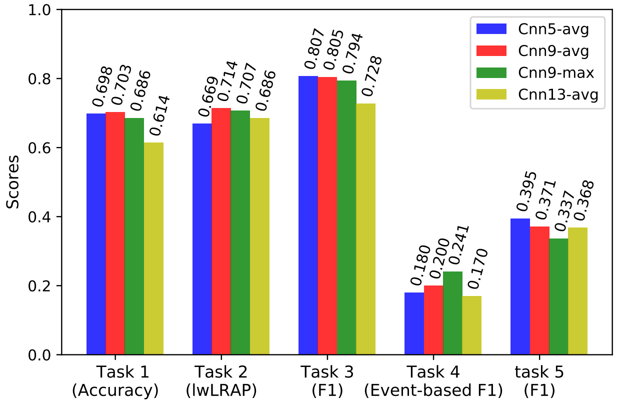

Figure 1: Performance of cross-task CNN systems on the DCASE 2019 challenge Task 1 - 5.

4. CONCLUSION

This paper presents cross-task convolutional neural network baseline systems for the DCASE 2019 tasks. We investigated the performance of 5-layer, 9-layer and 13-layer CNN systems with average or max pooling on five tasks. Figure 1 shows the performance of cross-task CNN systems on the DCASE 2019 challenge Task 1 - 5. Although different tasks apply different evaluation metrics, it can be seen that each individual task may favor particular CNN systems. Average pooling achieves better performance than max pooling in 4 out of 5 tasks. The 9-layer CNNs achieve better performance than the 5-layer and 13-layer CNNs in 3 out of 5 tasks. Overall, the 9-layer CNN with average pooling was shown to perform good for a majority of tasks. We release the cross-task baseline code. In future, we will continue to explore the cross-task structures with more neural network architectures such as recurrent neural networks.

5. ACKNOWLEDGEMENT

This research was supported by EPSRC grant EP/N014111/1 “Making Sense of Sounds” and a Research Scholarship from the China Scholarship Council (CSC) No. 201406150082.

6. REFERENCES

[1] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet, “Semantic annotation and retrieval of music and sound effects,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 467–476, 2008.

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

[3] Y. Xu, Q. Kong, W. Wang, and M. D. Plumbley, “Large-scale weakly supervised audio classification using gated convolutional neural network,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 121–125.

[4] D. Giannoulis, E. Benetos, D. Stowell, M. Rossignol, M. Lagrange, and M. D. Plumbley, “Detection and classification of acoustic scenes and events: An IEEE AASP challenge,” in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2013.

[5] 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 *Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2017.

[6] A. Mesaros, T. Heittola, and T. Virtanen, “A multi-device dataset for urban acoustic scene classification,” 2018.

[7] “DCASE 2019,” http://dcase.community/challenge2019/.

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

[9] E. Fonseca, M. Plakal, F. Font, and D. P. W. Ellis, “Audio tagging with noisy labels and minimal supervision,” http://dcase.community/challenge2019/task-audio-tagging.

[10] S. Adavanne, A. Politis, J. Nikunen, and T. Virtanen, “Sound event localization and detection of overlapping sources using convolutional recurrent neural networks,” *IEEE Journal of Selected Topics in Signal Processing*, 2018.

[11] R. Serizel, N. Turpault, A. P. Shah, J. Salamon, and H. Eghbal-zadeh, “Sound event detection in domestic environments,” http://dcase.community/challenge2019/task-sound-event-detection-in-domestic-environments.

[12] J. P. Bello, C. Silva, O. Nov, R. L. DuBois, A. Arora, J. Salamon, C. Mydlarz, and H. Doraiswamy, “SONYC: A system for the monitoring, analysis and mitigation of urban noise pollution,” in *Communications of the ACM*, 2018.

[13] Q. Kong, I. Sobieraj, W. Wang, and M. D. Plumbley, “Deep neural network baseline for DCASE challenge 2016,” *Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2016.

[14] K. Choi, G. Fazekas, and M. Sandler, “Automatic tagging using deep convolutional neural networks,” in *International Society of Music Information Retrieval (ISMIR)*, 2016.

[15] Q. Kong, T. Iqbal, Y. Xu, W. Wang, and M. D. Plumbley, “DCASE 2018 Challenge baseline with convolutional neural networks,” in *Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2018.

[16] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, “Audio set: An ontology and human-labeled dataset for audio events,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 776–780.

[17] E. Cakır, 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*, vol. 25, no. 6, pp. 1291–1303, 2017.

[18] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *International Conference on Learning Representations (ICLR)*, 2015.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems (NIPS)*, 2012, pp. 1097–1105.

[20] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *International Conference on Machine Learning (ICML)*, 2015.

[21] E. Fonseca, J. Pons Puig, X. Favory, F. Font Corbera, D. Bogdanov, A. Ferraro, S. Oramas, A. Porter, and X. Serra, “Freesound datasets: a platform for the creation of open audio datasets,” in *International Society for Music Information Retrieval (ISMIR), year=2017*.

[22] E. Fonseca, M. Plakal, D. P. Ellis, F. Font, X. Favory, and X. Serra, “Learning sound event classifiers from web audio with noisy labels,” *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

[23] G. Dekkers, S. Lauwereins, B. Thoen, M. W. Adhana, H. Brouckxon, T. van Waterschoot, B. Vanrumste, M. Verhelst, and P. Karsmakers, “The SINS database for detection of daily activities in a home environment using an acoustic sensor network,” in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017), Munich, Germany*, 2017, pp. 32–36.