# SOUND EVENT CLASSIFICATION AND DETECTION WITH WEAKLY LABELED DATA

*Sharath Adavanne*[*]*, Haytham M. Fayek, and Vladimir Tourbabin*

Facebook Reality Labs, Redmond, WA, USA

## ABSTRACT

The Sound Event Classification (SEC) task involves recognizing the set of active sound events in an audio recording. The Sound Event Detection (SED) task involves, in addition to SEC, detecting the temporal onset and offset of every sound event in an audio recording. Generally, SEC and SED are treated as supervised classification tasks that require labeled datasets. SEC only requires weak labels, i.e., annotation of active sound events, without the temporal information, whereas SED requires strong labels, i.e., annotation of the onset and offset times of every sound event, which makes annotation for SED more tedious than for SEC. In this paper, we propose two methods for joint SEC and SED using weakly labeled data: a Fully Convolutional Network (FCN) and a novel method that combines a Convolutional Neural Network with an attention layer (CNNatt). Unlike most prior work, the proposed methods do not assume that the weak labels are active during the entire recording and can scale to large datasets. We report state-of-the-art SEC results obtained with the largest weakly labeled dataset — Audioset.

***Index Terms*—** Convolutional neural network, sound classification, sound event detection, weakly supervised learning

## 1. INTRODUCTION

Sound Event Classification (SEC) is the task of recognizing the set of active sound events in a given audio recording. Additionally, detecting the temporal activity of each sound event, i.e., onset and offset times, is referred to as Sound Event Detection (SED). SEC and SED can be helpful in query based multimedia retrieval [1], acoustic scene analysis [2, 3], and bio-diversity monitoring [4–6]. The SED task requires datasets that are strongly labeled [7–9], i.e., annotation of active sound events and their respective onset and offset times. On the other hand, the SEC task requires weakly labeled datasets, that only provide annotation of the set of active sound events for every recording [5, 10, 11]. In terms of complexity, it is more tedious to annotate strongly labeled datasets than weakly labeled datasets.

The SEC task has traditionally been approached with Convolutional Neural Network (CNN) architectures [5, 10, 12]. Whereas for the SED task, which requires temporal localization of sound events, the joint architecture of CNNs with recurrent neural networks, referred to as Convolutional Recurrent Neural Network (CRNN) [8, 13], has shown consistently good results across SED datasets. Recently, it was shown in [9] that on large SED datasets, the performance of CNN architectures is comparable to CRNN architectures when the detection is happening at one-second resolution. In this paper, we aim to perform SED at a similar resolution using a large dataset. Given that the training time of CNN architectures is relatively faster than comparable CRNN architectures, we focus on CNN architectures.

Recently, methods have been proposed to jointly learn SEC and SED from just the weakly labeled data [14–18], in order to overcome the complexity of annotating strongly labeled datasets. Prior work in [14] used multiple established CNN architectures from the computer vision domain and applied them to this task, but these methods assumed that the weak labels were active throughout the recording during training, and is hereafter referred to as Strong Label Assumption Training (SLAT). This assumption leads to poor SEC performance, as shown in [17]. As an alternative to SLAT, the authors in [16] proposed a Fully Convolutional Network (FCN) based method that enabled learning from the weakly labeled dataset without assuming the presence of weak labels active throughout the recording; such a training approach is hereafter referred to as Weak Label Assumption Training (WLAT). Similar FCN based WLAT methods were also proposed in [17, 19], but all of these methods have only been evaluated on small datasets, and their performance on large datasets is unknown. In this paper, we study the performance of FCN on the largest publicly available dataset — Audioset [20].

In addition to the FCN-based approach, an alternative WLAT method is proposed that combines CNN and an attention layer (CNNatt). The attention layer enables the CNNatt to automatically learn to attend to relevant time segments of the audio during inference. Thus, in the current task, given a weak label, an attention layer can identify the relevant time segments in the audio where the weak label is active, and consequently provide strong labels.

To summarize, we study the performance of FCN for the task of joint SEC and SED from a large weakly labeled dataset, and further propose a novel CNNatt for the same task. The contributions of this paper are as follows. We present, for the first time since the benchmark work in [14], a study using the complete Audioset. Unlike [14], which used SLAT, the two methods in this paper, FCN and CNNatt, use WLAT to jointly perform SEC and SED. Finally, since Audioset provides just the weak labels, we only present the quantitative results for the SEC performance and compare them with the recently published baselines [14, 21, 22]. The SED performance is evaluated subjectively by visualizing the outputs and manual listening inspection.

## 2. METHOD

The input for the two methods — FCN and CNNatt — is a single channel audio recording. A feature extraction block produces $F$-band log mel-band energies for each of the $T$ frames of input audio. The feature sequence of dimension $T \times F$ for each recording is then mapped to the $C$ classes (SEC) as a multi-class multi-label classification task. Additionally, as an intermediate output, both studied methods generate frame-wise results for the $C$ classes (SED) of dimension $T_N \times C$. The time-dimensionality of the SED output $T_N$ is smaller than the input $T$ as a result of multiple max pooling operations. During training, only the audio recording and its respective weak label(s) — one-hot encoded — are used. During inference, given an input audio, both methods produce two outputs

[*] This work was performed during an internship at Facebook.

https://doi.org/10.33682/fx8n-cm43

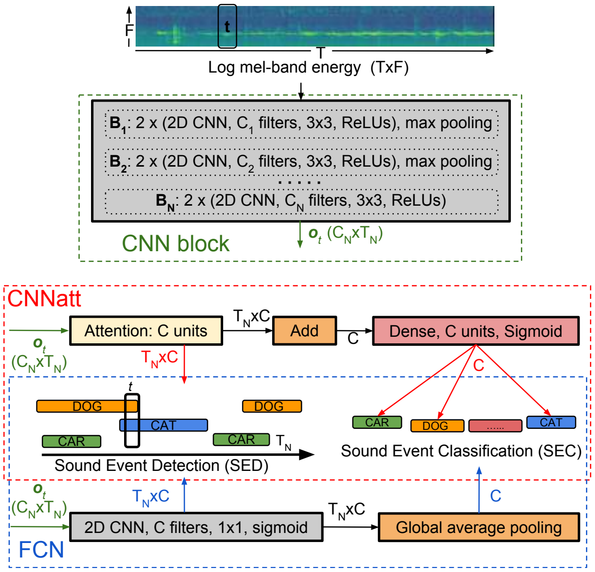

F
t
T
Log mel-band energy (TxF)
$B_1$: 2 x (2D CNN, $C_1$ filters, 3x3, ReLUs), max pooling
$B_2$: 2 x (2D CNN, $C_2$ filters, 3x3, ReLUs), max pooling
$B_N$: 2 x (2D CNN, $C_N$ filters, 3x3, ReLUs)
CNN block
$\mathbf{o}_t$ ($C_N$x$T_N$)
CNNatt
Attention: C units
$T_N$xC
Add
C
Dense, C units, Sigmoid
$T_N$xC
C
DOG
CAT
CAR
DOG
CAR
$T_N$
Sound Event Detection (SED)
CAR
DOG
......
CAT
Sound Event Classification (SEC)
$\mathbf{o}_t$ ($C_N$x$T_N$)
$T_N$xC
2D CNN, C filters, 1x1, sigmoid
$T_N$xC
Global average pooling
C
FCN

Figure 1: The Fully Convolutional Network (FCN) and Convolutional Neural Network with attention layer (CNNatt) methods for joint learning of SEC and SED from weakly labeled dataset.

in sequence: first, the strong labels (SED), followed by the weak labels (SEC). These outputs are the respective class probabilities in the continuous range of [0, 1]. A value closer to one signifies that the sound class is active, and closer to zero signifies that it is absent. The details of the feature extraction and the two methods studied are presented below.

### 2.1. Feature Extraction

The log mel-band energy features are extracted for each frame of 1024 samples with 50% overlap using a 1024-point fast Fourier transform. A total of 40 bands are extracted in the frequency range of 0-8000 Hz from an audio recording sampled at 16 kHz. For a 10 s audio input, the feature extraction step produces a sequence of $T = 320$ frames and $F = 40$ features.

### 2.2. Neural Network

#### *2.2.1. Fully Convolutional Network (FCN)*

Figure 1 presents the overall structure of the FCN. The input is a $T \times F$ dimension sequence of the extracted features. The initial layers of the network consist of 2D convolutional layers that learn local shift-invariant features. Each of the convolutional layers has filters with a $3 \times 3$ receptive field, with the output dimension kept the same as the input dimension using zero padding. Batch normalization [23] is performed on this output, followed by a Rectified Linear Unit (ReLU) activation and a dropout layer [24]. The audio features dimensionality is reduced by performing max pooling after every second convolutional layer, such that the temporal- and feature-dimensionality in the final convolutional layer $N$ with $C_N$ filters is reduced to $T_N$ and $F_N$, respectively. In the reduced dimensionality, each frame in $T_N$ represents one second of input audio and $F_N = 1$. These multi-layered convolutional layers are, hereafter, referred to together as the CNN block, and its output $\mathbf{o}_t$ is an embedding of dimension $C_N \times T_N$, as seen in Figure 1.

The embedding from the CNN block is fed to a single 2D convolutional layer with $C$ filters (equal to the number of classes in the dataset), a receptive field of dimension $1 \times 1$ and sigmoid activation to support multi-class multi-label classification. Given the CNN block embedding of dimension $C_N \times T_N$, the newly added layer produces SED results of dimension $T_N \times C$. Further, the SEC results are obtained from the SED results by performing a global average pooling across $T_N$. The FCN was tuned as described in Section 3.4.

#### *2.2.2. Convolutional Neural Network with attention layer (CNNatt)*

The overall structure of the CNNatt is shown in Figure 1. Given a feature sequence of $T \times F$ dimension, a CNN block similar to the FCN generates output $\mathbf{o}_t$ of dimension $C_N \times T_N$. This is fed to an attention layer identical to that described in [21, 22]. The attention layer performs the following operation on it:

$$\mathbf{a}_t = cls(\mathbf{o}_t) \odot (atn(\mathbf{o}_t) / \sum_{t=0}^{T} atn(\mathbf{o}_t)), \tag{1}$$

where $\odot$ signifies element-wise multiplication. The $atn()$ function guides the network to be attentive to certain time frames, while the $cls()$ function performs the classification for each input time frame $t$. The $atn()$ and $cls()$ functions are implemented as 2D convolutional layers with $C$ filters each and a receptive field of dimension $1 \times 1$. The $atn()$ function employs a softmax activation, while the $cls()$ function is implemented with a sigmoid activation. The output of the attention layer $\mathbf{a}_t$ produces the frame-wise SED results of dimension $T_N \times C$. Further, the SEC results are obtained from $\mathbf{a}_t$ by adding the activations across $T_N$ and feeding them to a fully connected dense layer with $C$ units and sigmoid activation. The CNNatt was tuned as described in Section 3.4.

The proposed implementation of the two methods enables them to operate on input audio of variable length, but with a minimum length criterion arising from the multiple max pooling operations employed. Both methods were trained for 100 epochs using binary cross entropy loss calculated between the predicted SEC output and the weak labels in the reference annotation of the dataset. As the optimizer, we employ Adam [25], a first-order adaptive variant of stochastic gradient descent, with the parameters introduced in [25], $\alpha = 0.001$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. Early stopping was used during training to avoid overfitting. The training was stopped if the mean Average Precision (mAP) score (see Section 3.2) did not improve for 25 epochs. The methods were implemented using PyTorch and trained in data parallel mode over eight GPUs.

## 3. EVALUATION

### 3.1. Dataset

We used the complete dataset, Audioset [20], in this paper. The dataset provides a pre-defined development and evaluation split. At the time of this study, only about 94% of the YouTube videos of Audioset were active. The audio recordings for these videos were pre-processed to have a sampling rate of 16 kHz, and a single channel. Although the two methods are invariant to the length of the input audio, the Audioset recordings used are of a constant length of ten seconds. The complete Audioset has $C$ = 527 classes with a highly imbalanced distribution (see [20] for more details).

### 3.2. Metrics

The mean Average Precision (mAP) metric is used to evaluate the performance of our methods for SEC, due to class imbalance, similar to that in prior studies on Audioset [20–22]. The mAP is defined as the mean of the area under the precision-recall curve across the $C$ classes,

$$mAP = \frac{1}{C} \sum_{i=1}^{C} \sum_{m=1}^{M} P_{m,i}(R_{m,i} - R_{m-1,i}), \tag{2}$$

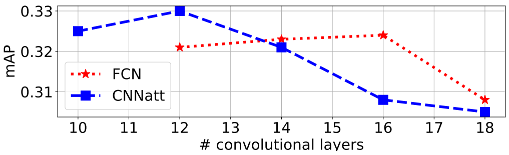


Figure 2: The mAP scores obtained with respect to different number of convolutional layers of FCN and CNNatt.

where $P_{m,i}$ and $R_{m,i}$ are the precision and recall values of class $i$ at $M$ different threshold values.

Finally, since the strong labels of Audioset are unavailable, we only visualize and manually inspect the SED performance.

### 3.3. Baseline Methods

We compare the performance of the two methods with four baseline methods [14, 21, 22]. Note that the dense method in [14] is the only method that was trained using the complete Audioset dataset; the other three baseline methods [14, 21, 22] study the performance on the Audioset embeddings obtained using a network trained on a dataset larger than Audioset.

The first method proposed in [14] uses log mel-band energy features similar to those used in this paper. A 64-band feature is mapped to 527 classes of Audioset using a multi-layered fully connected dense network with SLAT.

Among the methods using Audioset embeddings, the embeddings used in the second method proposed in [14] are different from the ones used in [21] and [22]. The embeddings in [14] were obtained using a ResNet-50 network and SLAT on a much larger YT-100M dataset (20× larger than Audioset — not available publicly). This network was then used as a feature extractor to generate embeddings from each of the Audioset recordings, hereafter referred to as ResNet embeddings. Finally, the benchmark scores on the ResNet embeddings were obtained using a multi-layered fully connected dense network similar to the one described above.

In comparison, the two recent methods — single attention [21] and multiple attention [22] — use embeddings that were generated using a VGGish network instead of a ResNet-50 and SLAT on a YT-8M dataset instead of a YT-100M dataset. This is referred to as VGGish embeddings hereafter and is publicly available. Unlike YT-100M, the YT-8M dataset is publicly available, but the exact splits used to learn the VGGish embeddings described above are unknown. The single attention method [21] uses multiple layers of a fully connected dense network, followed by an attention layer as the classification layer, whereas the multiple attention method [22] uses multiple attention layers located between fully connected layers, and concatenates the output of these attention layers to perform the final classification.

### 3.4. Experiments

Hyper-parameter tuning of both methods is performed to identify the best configuration for Audioset. Since the attention and dense layers of CNNatt and the final classification convolutional layer of FCN are dependent on the output number of classes, the only tunable part is the CNN block. In order to restrict the number of possible options to tune, we made sure that the number of filters in a convolutional layer doubles after every two layers. For example, $C_2 = 2C_1$ in Figure 1. The number of layers in the CNN block was tuned randomly [26] in the range of five to twenty with the number of filters in the first layer varying in the set $\in \{16, 32, 64\}$. In order to study the effect of regularization, the dropout layer was tuned in the set $\in \{0, 0.15, 0.3, 0.5, 0.75\}$.

Table 1: The mAP scores on Audioset with different methods

| Methods on Audioset recordings | mAP |
|---|---|
| Random chance | 0.005 |
| Dense [14] | 0.137 |
| FCN | 0.324 |
| CNNatt | **0.330** |
| | |
| Method on Audioset ResNet embeddings*+ | |
| Dense [14] | 0.314 |
| | |
| Method on Audioset VGGish embeddings*# | |
| Single attention [21] | 0.327 |
| Multiple attention [22] | **0.360** |

*Embeddings from network trained on dataset larger than Audioset.
+The YT-100M dataset used to train the ResNet is not publicly available.
#The YT-8M dataset used to train the embeddings network is publicly available, but the exact splits used to produce the embeddings are unknown.

The SEC performance is evaluated on the evaluation split of Audioset and compared with the existing baselines using Audioset recordings [14] and embeddings [14, 21, 22].

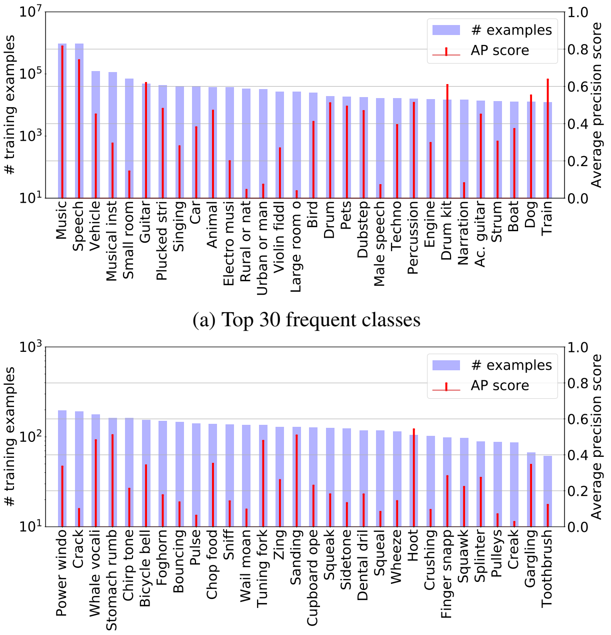
Finally, since the Audioset dataset lacks strong labels, we only perform a subjective analysis of SED through manually listening and visualizing the SED output of the two methods on a subset of Audioset examples.

## 4. RESULTS AND DISCUSSION

The best configuration for the FCN that obtained the highest mAP score had 16 convolutional layers including the (last) classification convolutional layer, with the first layer having 16 filters. The best CNNatt configuration had 12 convolutional layers in the CNN block, starting with 16 filters in the first layer, and followed by an attention and dense layer. Further, using zero dropout gave the best results for both methods. In terms of the number of parameters, the CNNatt uses only about 20% of the 25M parameters in FCN. The performance for other configurations of FCN and CNNatt when the first convolutional layer had 16 filters is visualized in Figure 2. Here, it can be observed that the CNNatt achieves better mAP scores than FCN with just 10 convolutional layers.

A classifier generating random results on Audioset obtains a mAP score of 0.005, as seen in Table 1. In comparison, the baseline dense method [14] trained on Audioset recordings obtained a mAP score of 0.137. This is a 27× improvement over the random results generating classifier. The FCN improved 2.36× over the dense method [14] and obtained a mAP score of 0.324. In fact, this score is higher than the dense method using ResNet embeddings [14], which obtained a mAP score of 0.314. This suggests that the FCN with WLAT outlearns the ResNet-50 with SLAT on a much larger YT-100M dataset.

The second method, CNNatt, obtained a best mAP score of 0.330. This is a significant result considering that the CNNatt learns to perform SEC better than FCN using only 20% of FCN's parameters. In fact, CNNatt performs better than the single attention method [21] trained using VGGish embeddings obtained from a VGG network and SLAT on a dataset larger than Audioset. This makes CNNatt the state-of-the-art for SEC using the complete Audioset recordings. The class-wise average precision score obtained with CNNatt on the evaluation split, and the corresponding number of examples in the development split is visualized in Figure 3. Among the top 30 frequent classes in Fig 3a we observe that the CNNatt performs better on *sound event* classes (E.g. Speech, Music, Car, Guitar, and Dog), and poorly on *sound scene* classes such



(a) Top 30 frequent classes

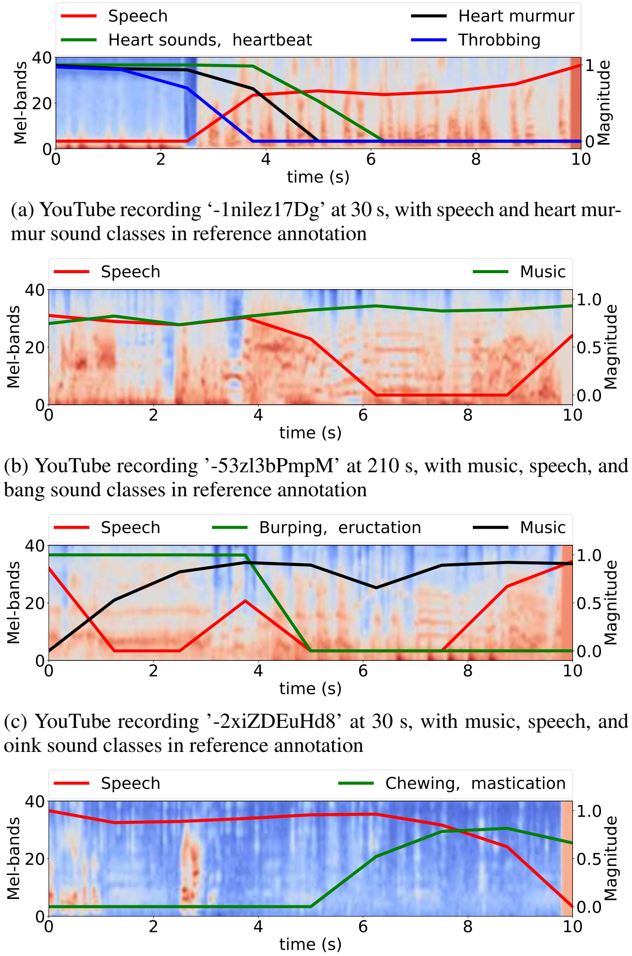
(b) Bottom 30 frequent classes

Figure 3: Visualization of the class-wise number of training examples and the corresponding average precision (AP) scores obtained with the CNNatt.

as Small room, Large room, Rural, and Urban classes.

The SED results obtained with the FCN for recordings in the evaluation split of Audioset are visualized in Figures 4a, 4b, 4c, and 4d. For example, in Figure 4a according to the reference annotation, the recording contains the classes: speech and heart murmur. From the spectrogram, we can distinguish a heartbeat-like repetitive structure in the first 2.5 s, and speech beyond 2.5 s. Similar temporal activity is observed in the overlaid class magnitude plot, which shows the SED output from the FCN. To simplify the visualization, we only show the classes whose SED output magnitude is greater than 0.5 throughout the recording. In addition to the heart murmur class, the FCN also recognized the first 2.5 s as heart sounds and throbbing, which are classes that sound similar to a heart murmur. Similarly, in Figure 4b, among the reference classes, the FCN detected the speech and music classes successfully, but missed the bang class (occurs from 1.5 s to 2.1 s) that was part of the music. In Figure 4c, the FCN detected the reference speech and music classes correctly, but missed the oink class (occurs from 6.3 s to 8.2 s), and successfully detected the burping class (first 3 s) that was missing in the reference annotation. Finally, in Figure 4d, the FCN missed the sniff sound class but successfully detected the chewing class that was missing in the reference annotation. Additionally, the FCN also over-predicted the speech class beyond 4 s.

In general, although the FCN missed detection of few short duration and low prior sound classes, we observe from the SED Figures 4a, 4b, 4c, and 4d a good recall of most of these low prior ($10^{-5}$) sound classes such as chewing, burping, heart sounds, and murmur. Another observation from these figures is that the onset and offset of the sound events are not of high precision. We believe that this is a result of both the dimensionality reduction (max pooling) operation within both methods and the limitation of learning strong labels from a weakly labeled dataset. Similar SED results were observed in all the recordings studied. Further, the SED performance of the CNNatt was comparable to that of the FCN with no characteristic difference. This suggests that given only the weak labels, the proposed methods can estimate their temporal activities with good confidence.



(a) YouTube recording '-1nilez17Dg' at 30 s, with speech and heart murmur sound classes in reference annotation

(b) YouTube recording '-53zl3bPmpM' at 210 s, with music, speech, and bang sound classes in reference annotation

(c) YouTube recording '-2xiZDEuHd8' at 30 s, with music, speech, and oink sound classes in reference annotation

(d) YouTube recording '-21_SXelVNo' at 30 s, with speech and sniff sound classes in reference annotation annotation

Figure 4: Visualization of SED results from FCN, and the input features for selected recordings from Audioset evaluation split.

## 5. CONCLUSION

In this paper, we studied two methods that perform joint SEC and SED using weakly labeled data, and evaluated the methods on the largest weakly labeled dataset — Audioset. The first method was based on a Fully Convolutional Network (FCN) and obtained a mean Average Precision (mAP) score of 0.324. The second novel method comprised multiple convolutional layers followed by an attention layer. This method was seen to perform better than the FCN with only 20% of the 25 M parameters in the FCN, and obtained a state-of-the-art mAP score of 0.330. In comparison to the baseline method trained on Audioset recordings, which was the previous state-of-the-art, the two methods improve the mAP score by at least a factor of 2.36. In fact, the two methods performed better than methods trained on Audioset embeddings that were obtained from learning on datasets larger than Audioset. This improvement in performance is a result of using a more powerful classifier and not assuming that the weak labels are active throughout the recording during training.

## 6. REFERENCES

[1] M. Xu, C. Xu, L. Duan, J. S. Jin, and S. Luo, “Audio keywords generation for sports video analysis,” in *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 2008.

[2] A. Harma, M. F. McKinney, and J. Skowronek, “Automatic surveillance of the acoustic activity in our living environment,” in *IEEE International Conference on Multimedia and Expo (ICME)*, 2005.

[3] M. Crocco, M. Cristani, A. Trucco, and V. Murino, “Audio surveillance: A systematic review,” in *ACM Computing Surveys (CSUR)*, 2016.

[4] E. Browning, R. Gibb, P. Glover-Kapfer, and K. E. Jones, “Passive acoustic monitoring in ecology and conservation,” in *World Wildlife Fund Conservation Technology Series 1(2)*, 2017.

[5] K. J. Piczak, “Environmental sound classification with convolutional neural networks,” in *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2015.

[6] B. J. Furnas and R. L. Callas, “Using automated recorders and occupancy models to monitor common forest birds across a large geographic region,” in *Journal of Wildlife Management*, vol. 79, no. 2, 2014, p. 325337.

[7] “Sound event detection in real life audio.” Detection and Classification of Acoustic Scenes and Events (DCASE), 2016. [Online]. Available: http://www.cs.tut.fi/sgn/arg/dcase2016/task-sound-event-detection-in-real-life-audio

[8] S. Adavanne, P. Pertilä, and T. Virtanen, “Sound event detection using spatial features and convolutional recurrent neural network,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

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

[10] H. Zhang, I. McLoughlin, and Y. Song, “Robust sound event recognition using convolutional neural networks,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

[11] S. Adavanne, K. Drossos, E. Cakir, and T. Virtanen, “Stacked convolutional and recurrent neural networks for bird audio detection,” in *European Signal Processing Conference (EUSIPCO)*, 2017.

[12] “Audio tagging with noisy labels.” Detection and Classification of Acoustic Scenes and Events (DCASE), 2019. [Online]. Available: http://dcase.community/challenge2019/task-audio-tagging-results#machine-learning-characteristics

[13] E. Çakır, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, “Convolutional recurrent neural networks for polyphonic sound event detection,” in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, 2017.

[14] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. Weiss, and K. Wilson, “Cnn architectures for large-scale audio classification,” in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[15] A. Kumar and B. Raj, “Audio event detection using weakly labeled data,” in *ACM on Multimedia Conference*, 2016.

[16] T.-W. Su, J.-Y. Liu, and Y.-H. Yang, “Weakly-supervised audio event detection using event-specific gaussian filters and fully convolutional networks,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[17] A. Kumar and B. Raj, “Deep CNN framework for audio event recognition using weakly labeled web data,” in *Machine Learning for Audio Signal Processing Workshop at NIPS*, 2017.

[18] S. Adavanne and T. Virtanen, “Sound event detection using weakly labeled dataset with stacked convolutional and recurrent neural network,” in *Workshop on Detection and classification of acoustic scenes and events (DCASE)*, 2017.

[19] A. Kumar, M. Khadkevich, and C. Fügen, “Knowledge transfer from weakly labeled audio using convolutional neural network for sound events and scenes,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[20] J. F. Gemmeke, D. P. W. 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 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[21] Q. Kong, Y. Xu, W. Wang, and M. D. Plumbley, “Audio set classification with attention model: A probabilistic perspective,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[22] C. Yu, K. S. Barsim, Q. Kong, and B. Yang, “Multi-level attention model for weakly supervised audio classification,” in *European Signal Processing Conference (EUSIPCO)*, 2018.

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

[24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” in *Journal of Machine Learning Research (JMLR)*, 2014.

[25] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *International Conference on Learning Representations (ICLR)*, 2015.

[26] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” in *Journal of Machine Learning Research (JMLR)*, 2012.