POLYPHONIC SOUND EVENT AND SOUND ACTIVITY DETECTION: A MULTI-TASK APPROACH

Arjun Pankajakshan[1], *Helen L. Bear*[1], *Emmanouil Benetos*[1,2]

[1] School of EECS, Queen Mary University of London, UK [2] The Alan Turing Institute, UK
{a.pankajakshan, h.bear, emmanouil.benetos}@qmul.ac.uk

ABSTRACT

Polyphonic Sound Event Detection (SED) in real-world recordings is a challenging task because of the dynamic polyphony level, intensity, and duration of sound events. Current polyphonic SED systems fail to model the temporal structure of sound events explicitly and instead attempt to look at which sound events are present at each audio frame. Consequently, the event-wise detection performance is much lower than the segment-wise detection performance. In this work, we propose a joint model approach to improve the temporal localization of sound events using a multi-task learning setup. The first task predicts which sound events are present at each time frame; we call this branch 'Sound Event Detection (SED) model', while the second task predicts if a sound event is present or not at each frame; we call this branch 'Sound Activity Detection (SAD) model'. We verify the proposed joint model by comparing it with a separate implementation of both tasks aggregated together from individual task predictions. Our experiments on the URBAN-SED dataset show that the proposed joint model can alleviate False Positive (FP) and False Negative (FN) errors and improve both the segment-wise and the event-wise metrics.

***Index Terms*—** Polyphonic sound event detection, sound activity detection, multi-task learning.

1. INTRODUCTION

Sound event detection (SED) [1] is the task of detecting the label, onset, and offset of sound events in audio streams. The overlapping nature of different sounds interfered with noise makes it difficult for accurate detection of sound events. Broadly speaking, the term *sound event* refers to a specific sound produced by a distinct physical sound source, such as 'a car passing by', 'a bird singing', 'a gunshot' or 'the melody of rain'. Different sound events have different duration and that can be dynamic too. For example, the event 'rain' can be considered as a continuous event at the same time 'a gunshot' is an instantaneous event. Typically, a soundscape contains multiple sound events that can occur simultaneously as in a real scenario or at separate time instants which are respectively termed as polyphonic sounds and monophonic sounds. In recent years, SED has been utilized in many applications including audio surveillance [2], health care monitoring [3], urban sound analytics [4], bio-acoustics [5, 6] and smart home devices [7].

1.1. Related Work

Most recent advances in polyphonic SED are largely attributed to the use of Machine Learning and Deep Neural Networks [8, 9, 10, 11, 12, 13]. In particular, the use of Convolutional Recurrent Neural Networks (CRNNs) has significantly improved SED performance in the past few years [14, 15, 16, 17]. However, there are three main disadvantages with current CRNN-based polyphonic SED approaches. 1) Most CRNN-based polyphonic SED systems use a frame-wise cost function for training; as a result these systems often show relatively high segment-wise accuracy but low event-wise accuracy. For example, in the DCASE 2016 task on event detection in real life audio [18], the F1 score is around 30% at segment level (frame length of 1 second), but only around 5% at event level (tolerance of 200 ms for onset and 200 ms or half length for offset). 2) These methods require pre-segmented training data. For RNNs to make a prediction at every frame, it is necessary to provide the exact start and end times of the sound events in the training data, making data annotation an extremely time-consuming process. 3) CRNN-based SED requires post-processing methods to transform the output into event sequences. Within these limitations of CRNN based polyphonic SED, in this work we attempt to enhance CRNN based polyphonic SED performance using auxiliary learning without any additional input data representation using a joint model approach.

1.2. Contributions of this work

To the best of our knowledge this is the first attempt to treat the polyphonic SED task in a joint model framework with sound activity detection (SAD) as an auxiliary task. We define SAD as the task of detecting the presence or absence of any sound events, analogous to voice activity detection in speech processing. Our experimental results show that re-weighting the predictions of the SED model with the SAD model predictions, lessen False Positive (FP) errors in both segments and events, and False Negative (FN) errors in events which in turn improve polyphonic sound event detection performance at both the segment and the event levels. Implicitly, the joint model helps in better temporal localization of sound events. To claim the effectiveness of the proposed auxiliary task on polyphonic SED, irrespective of the benefits of multi-task learning, we also verify the proposed approach by training the SED and SAD models separately and re-weight the predictions. The results are compared with our baseline SED model, aggregation of separately trained SED and SAD models, and with the joint model.

The rest of the paper is organized as follows. Section 2 covers the proposed joint model with details on model architecture, feature extraction, training, and experiments. Section 3 presents the dataset and evaluation metrics used in this work. In Section 4 the results, and discussions over the results are reported. Section 5 provides

AP is supported by a QMUL Principal's studentship. HLB is funded under EPSRC grant EP/R01891X/1. EB is supported by RAEng Research Fellowship RF/128 and a Turing Fellowship. This research was supported by an NVIDIA GPU Grant.

conclusions and future directions related to this work.

2. PROPOSED METHOD

The majority of frame-wise prediction models proposed for polyphonic SED attempt to predict the class labels of sound events present in each frame of an audio sample. We assume that, to some extent, the majority of environmental sounds are either harmonic or percussive in nature. With this assumption, we propose that it is much easier to predict the presence or absence of any events in an audio frame, irrespective of categorization of events. In this work, we take advantage of the frame-wise training approach to propose a joint model for polyphonic SED using multi-task learning. The first task is polyphonic SED that predicts the class labels of sound events present in each frame of input audio. We call this task as 'SED model', which is a frame based multi-label event classifier. The second task is an auxiliary model which is a binary classifier that predicts whether an event activity is present or not in each frame of the audio. We call this task as 'SAD model'. From a signal processing perspective, we formulate the effects of dynamic polyphony levels in the sound events similar to the masking effects in speech. Consequently, low energy events are masked by high energy events at each frame. This results in many FN errors in event predictions which degrade the performance of SED. However SAD predictions are not affected by masking effects between co-occurring sound events. Furthermore, SAD can exploit polyphony to ensure the presence of an event even if one event is masked by another event with similar or different acoustic properties. The proposed joint approach takes advantage of the auxiliary branch to re-weight the prediction of sound events at each frame which helps to mitigate the FN errors in SED. A block diagram of the proposed joint model is shown in Fig. 1.

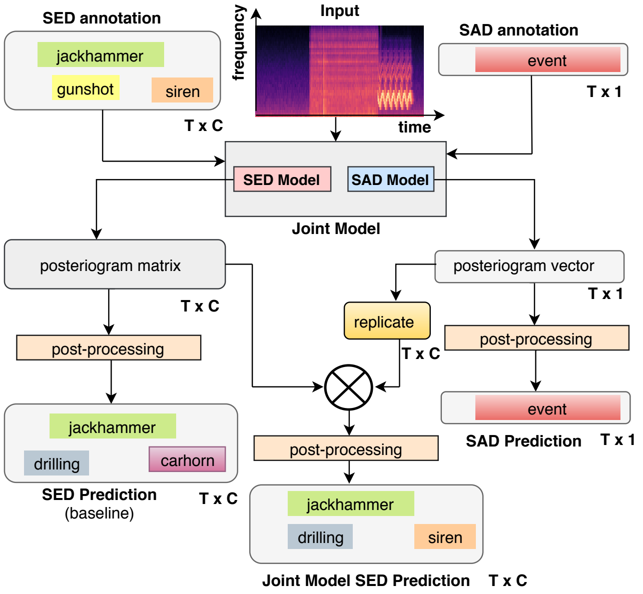

Figure 1: Block diagram of the proposed joint model for polyphonic SED (T is the number of frames in the input data representation and C is the total number of sound event classes in the dataset).

2.1. Model configuration

We use a state-of-the-art CRNN model architecture presented in [14] to build our SED and SAD models, with some modifications. Our focus is to prove the effectiveness of the proposed joint model framework in enhancing polyphonic sound event detection. With this aim, we lessen the complexity of the CRNN architecture from the baseline architecture [14]. Both the SED model and the SAD model have three blocks of convolutional layers, followed by a single Gated Recurrent Unit (GRU) layer. The SED model has a single dense layer after the GRU layer whereas the SAD model has two dense layers, which is the only difference between the two model architectures. The temporal dimension of the input data representation is unaltered in both the models. The detailed network architecture is shown in Fig. 2. The output of the SED model is a posteriogram matrix with dimensions $T \times C$, where T is the number of frames in the input data representation and C is the total number of sound event classes in the dataset. The output representation of the SAD model is a posteriogram vector with dimension T. The output of the joint model is the posteriogram matrix with dimensions $T \times C$, re-weighted on the SED output using the T dimensional vector of the SAD model for each event class. This re-weighted posteriogram matrix is converted into a binary matrix using a threshold value prior to evaluation.

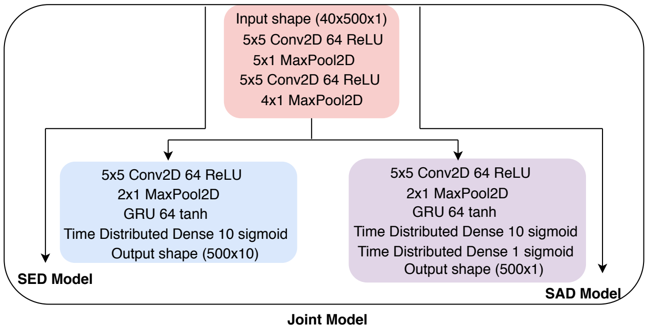

Figure 2: The proposed joint model architecture along with the individual SED and SAD model architectures.

2.2. Feature Extraction

We use librosa [19] to compute mel-scaled spectrograms from the input raw audio, which is the input data representation to all our models. Short-term Fourier transform (STFT) is employed to obtain the spectrogram from the input audio recordings with a hop length of 882, an FFT window of 2048, and a sample rate of 44.1kHz. This process converts a ten second (duration of recordings in the URBAN-SED [20] dataset) audio recording into a 1024×500 dimensional spectrogram representation. Each frame of this spectrogram is converted into a 40-dimensional vector of log filter bank energies using a Mel filterbank. We apply min-max normalization on the mel band energies. Hence, each 10-second audio recording is represented by a 40×500 dimensional Mel-spectrogram.

2.3. Training

We train both the SED model and the SAD model in a supervised manner. The dimension of the labels for the SED model is $T \times C$, and for the SAD model, it is T. The total loss of the joint model is the weighted sum of two cross-entropy losses; one from the SED model and one from the SAD model:

$$L_{joint\ model} = a \cdot L_{SED} + b \cdot L_{SAD}, \qquad (1)$$

where, $L_{joint\ model}$ denotes the loss function of the joint model, and L_{SED} and L_{SAD} denote the cross entropy losses of the SED and SAD models respectively. During training of the joint model, the individual losses are weighted differently to understand the influence of each task on the other; a and b denotes the respective loss weights. We also evaluate the influence of the number of shared layers on the joint model performance.

Batch normalization [21] is performed on the activations of every CNN layer and dropout [22] with a probability of 0.30 is used for regularization. We train the network for 200 epochs using a binary cross entropy loss function for both tasks and with Adam [23] optimizer with a learning rate of 0.001. Early stopping is used to reduce the overfitting of the network to the training data. The proposed joint model is implemented using Keras [24] with Tensorflow [25] as backend.

2.4. Experimental Setup

To validate the effectiveness of the proposed joint model in improving polyphonic SED performance we conduct a total of four experiments. In all experiments, we train the neural network models with the same input data representation as described in subsection 2.2. The posteriogram outputs from the models are thresholded in order to obtain the binary event matrices prior to evaluation. We investigate the effect of different threshold values (0.2, 0.3, 0.4, 0.5) on the performance of the standalone SED and SAD, and the joint model using the validation set. Based on the best results, we choose a threshold value of 0.2 for the SED models in Exp. 1, 3, and 4. For the SAD model in Exp. 2 we choose 0.5 as the threshold. The variables in capital bold letters refer to matrices and in small bold letters refer to vectors.

Experiment 1: We train a standalone SED model to evaluate SED performance. This is our baseline result. The model is trained with strong labels; $\mathbf{GT}_{SED}$ with dimension $T \times C$ denotes the ground truth binary event activity matrix for each sample. Model architecture and training details are explained in subsection 2.1 and 2.3. L_{SED} denotes the cross entropy loss of this model. For each sample, $\mathbf{P}_{SED}$ with dimensions $T \times C$ is the posteriogram matrix output of the SED model. $\mathbf{B}_{SED}$ denotes the binary equivalent of $\mathbf{P}_{SED}$ after thresholding over a constant. Eventually, we use the binary event prediction matrices for evaluation.

Experiment 2: We train a standalone SAD model for event activity prediction as explained in subsection 2.3. The ground truth labels $\mathbf{gt}_{SAD}$ used to train this model are the binary event activity vectors over the entire time span of the input data. The ground truth labels are created by truncating the strong label SED ground truths across the event classes as:

$$\mathbf{gt}_{SAD} := \mathbf{gt}_{SED_1} \vee \mathbf{gt}_{SED_2} \vee \ldots \vee \mathbf{gt}_{SED_C} \qquad (2)$$

where $\mathbf{gt}_{SED_C}$ denotes the ground truth event activity vector corresponding to event C of $\mathbf{GT}_{SED}$. L_{SAD} denotes the cross entropy loss of this model. For each sample, $\mathbf{p}_{SAD}$ with dimension T is the posteriogram vector output of the SAD model. $\mathbf{b}_{SAD}$ denotes the binary equivalent of $\mathbf{p}_{SAD}$ after thresholding over a constant. We use the binary event activity vectors for evaluation.

Experiment 3: To demonstrate the gain of carrying out sound activity detection on polyphonic SED performance, irrespective of the advantages of multi-task learning, we aggregate the output predictions of the standalone SED model of Exp. 1 and the SAD model of Exp. 2. The output posteriogram vectors from the SAD model are repeated to form posteriogram matrices as:

$$\mathbf{P}_{SAD} := \mathrm{r}epeat\mathbf{p}_{SAD} \qquad (3)$$

We use these posteriogram matrices to re-weight (Hadamard product) the outputs from the SED model to form the final predictions as:

$$\mathbf{P}_{joint\_model} := \mathbf{P}_{SED} \odot \mathbf{P}_{SAD} \qquad (4)$$

The final predictions are thresholded to get event activity binary matrices to compute the evaluation metrics.

Experiment 4: The proposed joint model approach for polyphonic SED as explained in this section is trained in a multi-task learning setup. The total loss (eq. 1) of the model is the weighted sum of the SED model loss and the SAD model loss. Exp. 4a and 4b respectively represent the proposed joint model (with equal loss weights of a = b = 0.5 and with loss weights of a = 0.3 and b = 0.7) for the SED and SAD tasks. Post-processing and evaluation is carried out as explained in Exp. 3.

3. DATASET AND METRICS

We use the URBAN-SED [20] dataset in all experiments. URBAN-SED is a dataset of 10,000 soundscapes with sound event annotations generated using Scaper [20], an open-source library for soundscape synthesis and augmentation. All recordings are of 10sec long, 16-bit mono and sampled at 44.1kHz. The annotations are strong, meaning for every sound event the annotations include the onset, offset, and label of the sound event. Every soundscape contains between 1-9 sound events from the list (air_conditioner, car_horn, children_playing, dog_bark, drilling, engine_idling, gun_shot, jackhammer, siren and street_music) and has a background of Brownian noise. The URBAN-SED [20] dataset comes with pre-sorted train, validation and test sets; we use this default data split. Among 10,000 soundscapes, 6000 samples are used for training, 2000 samples for validation and 2000 samples for testing.

In all experiments, we use the F-score and Error Rate (ER) for performance evaluation, with F-score as the primary evaluation metric. The evaluation metrics are computed in both segment-wise and event-wise manners using the sed_eval tool [26]. Segment-wise metrics compare system output and reference in short time segments. Event-wise metrics compare system output and corresponding reference event by event, with the metric showing how well the system detects event instances with the correct onset and offset. The evaluation scores presented in this work are micro averaged values, computed by aggregating intermediate statistics over all test data; each instance has equal influence on the final metric value. The ideal value of F-score is 100 and ER is zero. We use a segment length of one second to compute segment based metrics. The event-based metrics are calculated with respect to event instances by evaluating only onsets with a time collar of 250 ms.

4. EVALUATION

Both segment-based and event-based F-score and error rates are computed on the test set of the URBAN-SED [20] dataset for each of the experiments listed in Section 2. Tables 1 and 2 show the results of Exp. 4, which respectively summarize the influence of shared layers and loss weights on the performance of the joint

Table 1: Results wrt number of shared layers for the joint model of Exp. 4.

	F1 (%)		Error rate	
Layers shared	Segment	Event	Segment	Event
Conv 1	39.40	8	1.08	3.79
Conv 1,2	**41.03**	**8.76**	**0.97**	**3.58**
Conv 1,2,3	36.47	6.19	1.31	3.75

Table 2: Results wrt different loss weights for the model of Exp. 4.

loss weights	F1 (%)		Error rate	
(a, b)	Segment	Event	Segment	Event
(0.7, 0.3)	36.12	5.62	1.08	4.41
(0.5, 0.5)	**41.03**	8.76	0.97	3.58
(0.3, 0.7)	34.18	**10.28**	**0.91**	**2.36**

Table 3: Results for Experiments 1, 2, 3, and 4.

	F1 (%)		Error rate	
Case	Segment	Event	Segment	Event
Exp. 1	35.48	7.34	1.54	3.81
Exp. 2	97.48	43.14	0.05	0.78
Exp. 3	39.25	**11.13**	1.21	2.90
Exp. 4a	**41.03**	8.76	0.97	3.58
Exp. 4b	34.18	10.28	**0.91**	**2.36**

Table 4: Results on individual branches for the model of Exp. 4a.

	F1 (%)		Error rate	
Model	Segment	Event	Segment	Event
J_{SAD}	98.53	46.23	0.03	0.72
J_{SED}	40.99	8.28	0.97	3.65
J_{SED_SAD}	41.03	8.76	0.97	3.58

Table 5: Precision and Recall for SED for Exps. 1, 3, and 4a.

	P (%)		R (%)	
Case	Segment	Event	Segment	Event
Exp. 1	26.69	4.70	**52.88**	16.85
Exp. 3	31.82	**7.57**	51.19	**21**
Exp. 4a	**36.84**	5.67	46.29	19.26

model. To understand the effect of shared layers on joint model performance, the experiments are carried out with equal loss weights (a = b = 0.5) as described in eq. (1). The best scores (F-scores of 41.03% (segment) and 8.76% (event)) are achieved with sharing the first two convolutional layers. We use this model architecture to evaluate the influence of individual loss weights and in all further comparisons. Analysing the results of Table 2, the best metric scores are obtained by using the same weight of (a = b = 0.5) for both the SED and the SAD losses. An interesting observation is that the event based metrics improve (from 8.76% to 10.28%) when a slightly higher weight is given to the SAD loss which intuitively validates the influence of SAD task on efficient temporal localization of sound events.

Table 3 summarizes the evaluation results for Experiments 1-4. Exp. 1 represents the baseline results of the standalone SED model. Exp. 2 shows the metric values of the standalone SAD model. The high scores of the SAD model (F-scores of 97.48% and 43.14% at segment and event levels) for event activity prediction justify the intuition of using the SAD model as an auxiliary branch to improve polyphonic SED performance. Exp. 3 represents the proposed joint model approach with separate training for the SED and SAD tasks. Analyzing the overall results, the segment based F-score increases from the baseline score of 35.48% to 39.25% and 41.03% respectively for separate training and joint training of the proposed method. Similarly, there is improvement in event based F-score from a baseline value of 7.34% to 11.13% and 8.76% respectively for separate training and joint training of the proposed method. With the proposed method error rates are also reduced at both segment and event levels. The event based F-score for the jointly trained model is slightly lower than the separately trained and aggregated model. However the improved F-score for segments and events for the separately trained model in Exp. 3 validates our method. With further improvements to the architecture and multi-task learning setup we believe the joint model can further improve detection performance. The results clearly indicate the effectiveness of the proposed method for improving polyphonic sound event detection performance.

Table 4 shows another interesting observation from the best joint model of Exp. 4a. J_{SAD} represents the sound activity detection performance of the 'SAD model' of the joint model, similarly J_{SED} and J_{SED_SAD} respectively represents sound event detection performance with the 'SED model' and aggregation of predictions from 'SED model' and 'SAD model' of the joint model. The results show that with the best combination of loss weights and number of shared layers, the 'SED model' of the joint model achieves almost the same results to that of the aggregation of the SED and SAD predictions. This means that with a good joint training procedure it is possible to achieve the best performance without the aggregation of individual SED and SAD predictions at a post-processing stage.

We also verify that the proposed method can lessen FP errors in both segments and events, and FN errors in events. We compute the segment-based and event-based precision (P) and recall (R) for Experiments 1, 3, and 4a (see Table 5). Based on these results we also claim that to some extent the proposed approach helps in temporal localization of sound events. At the same time, we understand that our proposed method has the following limitations: 1) The proposed method is not very successful in decreasing the gap between segment-wise and event-wise scores in polyphonic SED; 2) Our work has not addressed the explicit modeling of sound events.

5. CONCLUSIONS AND FUTURE WORK

Within the limitations of current frame-based CRNN training methods, we proposed an auxiliary learning branch for event activity detection in order to improve polyphonic SED performance. We successfully evaluated the effectiveness of the proposed method on the URBAN-SED dataset. From our experimental results we conclude that: 1) the proposed joint model can improve polyphonic SED performance at both the segment and event levels and, 2) the proposed joint model can alleviate FP errors in both segments and events, and FN errors in events; which in turn improve the temporal localization of sound events. To further validate the method, we need to conduct similar experiments on more unbalanced and real-world datasets. We also plan to add one more branch to the existing method to predict frames conditioned on the event onsets as demonstrated in [27]. We hope such an implementation can bring down the error rate and also help in explicit modeling of sound events.

6. REFERENCES

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

[2] P. Foggia, N. Petkov, A. Saggese, N. Strisciuglio, and M. Vento, “Reliable detection of audio events in highly noisy environments,” *Pattern Recognition Letters*, vol. 65, pp. 22–28, 2015.

[3] S. Goetze, J. Schroder, S. Gerlach, D. Hollosi, J.-E. Appell, and F. Wallhoff, “Acoustic monitoring and localization for social care,” *Journal of Computing Science and Engineering*, vol. 6, no. 1, pp. 40–50, 2012.

[4] J. Salamon and J. P. Bello, “Feature learning with deep scattering for urban sound analysis,” in *23rd European Signal Processing Conference (EUSIPCO)*, 2015, pp. 724–728.

[5] D. Stowell and D. Clayton, “Acoustic event detection for multiple overlapping similar sources,” in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2015, pp. 1–5.

[6] V. Morfi and D. Stowell, “Deep learning for audio event detection and tagging on low-resource datasets,” *Applied Sciences*, vol. 8, no. 8, p. 1397, 2018.

[7] S. Krstulović, “Audio event recognition in the smart home,” in *Computational Analysis of Sound Scenes and Events*. Springer, 2018, pp. 335–371.

[8] 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, pp. 316–320.

[9] T. Komatsu, T. Toizumi, R. Kondo, and Y. Senda, “Acoustic event detection method using semi-supervised non-negative matrix factorization with a mixture of local dictionaries,” in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016)*, 2016, pp. 45–49.

[10] Z. Zhang and B. Schuller, “Semi-supervised learning helps in sound event classification,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 333–336.

[11] E. Benetos, G. Lafay, M. Lagrange, M. D. Plumbley, E. Benetos, G. Lafay, M. Lagrange, and M. D. Plumbley, “Polyphonic sound event tracking using linear dynamical systems,” *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 25, no. 6, pp. 1266–1277, 2017.

[12] A. Jansen, M. Plakal, R. Pandya, D. P. Ellis, S. Hershey, J. Liu, R. C. Moore, and R. A. Saurous, “Unsupervised learning of semantic audio representations,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 126–130.

[13] A. Kumar and B. Raj, “Audio event and scene recognition: A unified approach using strongly and weakly labeled data,” in *Neural Networks (IJCNN), 2017 International Joint Conference on*. IEEE, 2017, pp. 3475–3482.

[14] 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.

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

[16] E. Çakir and T. Virtanen, “End-to-end polyphonic sound event detection using convolutional recurrent neural networks with learned time-frequency representation input,” in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–7.

[17] S. Adavanne and T. Virtanen, “Sound event detection using weakly labeled dataset with stacked convolutional and recurrent neural network,” *arXiv preprint arXiv:1710.02998*, 2017.

[18] “Sound event detection in real life audio, dcase 2016 challenge results,” http://www.cs.tut.fi/sgn/arg/dcase2016/task-results-sound-event-detection-in-real-life-audio.

[19] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, “librosa: Audio and music signal analysis in python,” in *Proceedings of the 14th python in science conference*, 2015, pp. 18–25.

[20] 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)*, 2017, pp. 344–348.

[21] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *arXiv preprint arXiv:1502.03167*, 2015.

[22] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[23] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[24] “Keras: The python deep learning library, keras v2.2.4,” https://keras.io/.

[25] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, “Tensorflow: A system for large-scale machine learning,” in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI16)*, 2016, pp. 265–283.

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

[27] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. Engel, S. Oore, and D. Eck, “Onsets and frames: Dual-objective piano transcription,” *International Conference of Music Information retrieval (ISMIR)*, 2018.