

ENABLING EARLY AUDIO EVENT DETECTION WITH NEURAL NETWORKS

Huy Phan*, Philipp Koch[‡], Ian McLoughlin[†], and Alfred Mertins[‡]

* University of Oxford, Department of Engineering Science, Oxford, UK

[†] University of Kent, School of Computing, Kent, UK

[‡] University of Lübeck, Institute for Signal Processing, Lübeck, Germany

huy.phan@eng.ox.ac.uk, ivm@kent.ac.uk, koch,mertins@isip.uni-luebeck.de

ABSTRACT

This paper presents a methodology for early detection of audio events from audio streams. Early detection is the ability to infer an ongoing event during its initial stage. The proposed system consists of a novel inference step coupled with dual parallel tailored-loss deep neural networks (DNNs). The DNNs share a similar architecture except for their loss functions, i.e. weighted loss and multitask loss, which are designed to efficiently cope with issues common to audio event detection. The inference step is newly introduced to make use of the network outputs for recognizing ongoing events. The monotonicity of the detection function is required for reliable early detection, and will also be proved. Experiments on the ITC-Irst database show that the proposed system achieves state-of-the-art detection performance. Furthermore, even partial events are sufficient to achieve good performance similar to that obtained when an entire event is observed, enabling early event detection.

Index Terms— Audio event detection, early detection, deep neural networks, monotonicity

1. INTRODUCTION

Great progress has been made in recent years on the problem of audio event detection, in both methodologies [1, 2, 3, 4, 5] as well as available datasets [6, 7, 8, 9]. However, many previous works focused only on classifying audio events after fully observing an entire event. We are still missing an important aspect of audio event detection: namely, the early detection of ongoing events, such as from live audio streams. Early AED differs from standard AED by requiring ongoing events to be recognized as early as possible, or when given only a partial observation of the beginning of an event [10]. The ability to *reliably* detect ongoing events at their early stage is important in many scenarios, such as surveillance and safety-related applications, which require low latency reaction to potentially dangerous events. However, this requires monotonicity in the detection function, which is not easily fulfilled with methods used to recognize complete events [10, 11].

There has been an influx of works employing deep networks, such as DNNs [1, 12], convolutional neural networks (CNNs) [2, 13, 4, 14], and recurrent neural networks (RNNs) [3, 15], for audio event analysis. However, these works primarily focused on new network architectures. Little attention has been paid on loss functions to address the common issues of audio event detection in audio streams and, more importantly, to enable inferring an ongoing event at its early stage. To this end, we propose two tailored loss functions: (1) weighted loss for unbalanced foreground/background classification

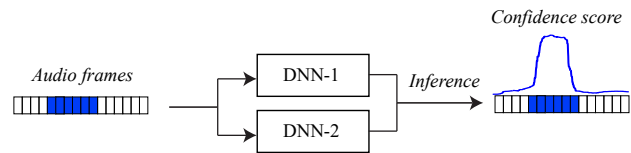


Fig. 1. An overview of the proposed system.

and (2) multitask loss for jointly modelling event class distributions (i.e. event classification) and event temporal structures (i.e. regression for event boundary estimation). These loss functions are used with two DNNs which are operating in parallel as demonstrated in Fig. 1. Furthermore, a novel inference scheme is introduced to make use of the network outputs to infer ongoing events for early detection. Intuitively, the confidence score of a given target event occurring will gradually accumulate when each incoming audio frames describing that event is analysed. Since the confidence score is a monotonic function (see Section 2.4), an event that is detectable by the system can be reliably detected early in time as soon as the accumulated confidence score reaches a predetermined threshold.

This work can be seen as an extension of our previous work [16, 10, 17] into the deep learning paradigm. In [16, 10], different decision forests were learned separately for event classification and event boundary estimation. Although these two tasks can be jointly learned with classification-regression forests [17], they are class-specific. In contrast, the proposed multitask DNN is a single model for multiple classes and multiple tasks simultaneously. It is also worth mentioning that the tailored loss functions were used by our AED system [18] for the DCASE 2017 challenge [7]. However, the simple median filtering used in that preliminary work is not capable of early event detection. The novel inference scheme proposed in this work is explicitly designed to overcome this limitation.

2. THE PROPOSED APPROACH

An overview of the proposed system is illustrated in Fig. 1. A continuous audio signal is firstly decomposed into frames. Each frame is then presented to the weighted-loss and multitask-loss DNNs denoted as *DNN-1* and *DNN-2*, respectively. The former is used to determine whether the input frame is foreground or background whereas the latter is employed for joint event classification and event boundary estimation. *DNN-1* and *DNN-2* share a similar architecture, including three fully connected layers as demonstrated in Fig. 2 and Table 1. The differences are the dropout probability [19], which is 0.5 and 0.2 for *DNN-1* and *DNN-2* respectively, the output layer, and the loss function. Note that the DNNs operate in parallel rather than as a cascade in [18, 10, 16]. The network outputs are then used in the inference step to compute a confidence score that a target event is occurring at a certain time index.

*The work was performed when H. Phan was at the Institute of Signal Processing, University of Lübeck.

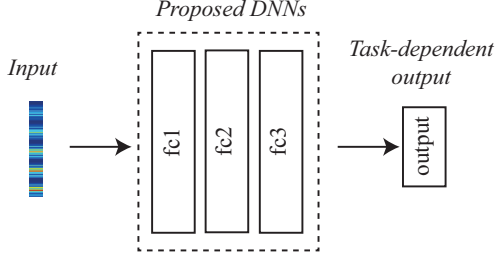


Fig. 2. The proposed DNN architecture.

2.1. DNN-1: Fore-/background classification with weighted loss

In general, for audio event detection in continuous streams, the number of background frames is significantly larger than of foreground ones. This leads to a skewed classification problem with a dominance of the background samples. Since foreground samples are more valuable than background ones, we penalize the network more for false negative errors than for false positives. The weighted loss is designed for this purpose.

Let $\{(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_N, \mathbf{y}_N)\}$ denote a training set of N examples where $\mathbf{x} \in \mathbb{R}^D$ denotes a feature vector of size D . $\mathbf{y} \in \{0, 1\}^2$ denotes a binary one-hot encoding vector where 0 and 1 indicate background and foreground categories, respectively. The weighted loss reads,

$$E_w(\boldsymbol{\theta}) = -\frac{1}{N} \left(\lambda_{fg} \sum_{n=1}^N \mathbb{I}_{fg}(\mathbf{x}_n) \mathbf{y}_n \log(\hat{\mathbf{y}}_n(\mathbf{x}_n, \boldsymbol{\theta})) + \lambda_{bg} \sum_{n=1}^N \mathbb{I}_{bg}(\mathbf{x}_n) \mathbf{y}_n \log(\hat{\mathbf{y}}_n(\mathbf{x}_n, \boldsymbol{\theta})) \right) + \frac{\lambda}{2} \|\boldsymbol{\theta}\|_2^2, \quad (1)$$

where $\boldsymbol{\theta}$ denotes the network's trainable parameters. $\mathbb{I}_{fg}(\mathbf{x})$ and $\mathbb{I}_{bg}(\mathbf{x})$ are indicator functions which specify whether the sample \mathbf{x} is foreground or background, respectively. λ_{fg} and λ_{bg} are *penalization weights* for false negative errors and false positive errors, respectively. The hyper-parameter λ is used to trade off the error terms and the ℓ_2 -norm regularization term, $\|\cdot\|_2^2$. The posterior probability $\hat{\mathbf{y}}(\mathbf{x}, \boldsymbol{\theta})$ is obtained by applying a softmax to the output layer.

2.2. DNN-2: Joint event classification and boundary estimation with multitask loss

We enforce DNN-2 to jointly model the class distribution for event classification and the event temporal structures for event boundary estimation, similarly to [16, 17]. The proposed multi-task loss is specialized for this purpose. Multitask modelling can also be interpreted as implicit regularization, which is expected to improve generalization of a network [20, 21].

In addition to the one-hot encoding vector $\mathbf{y} \in \{0, 1\}^C$, where C is the number of event categories, we associate a sample \mathbf{x} with a distance vector $\mathbf{d} = (d_{on}, d_{off}) \in \mathbb{R}^2$. d_{on} and d_{off} denote the distances from the audio frame \mathbf{x} to the corresponding event onset

Table 1. The parameters of the DNNs. A dropout probability of 0.5 and 0.2 is used for DNN-1 and DNN-2, respectively.

Layer	Size	Activation	Dropout
fc1	512	ReLU	0.5/0.2
fc2	256	ReLU	0.5/0.2
fc3	512	ReLU	0.5/0.2

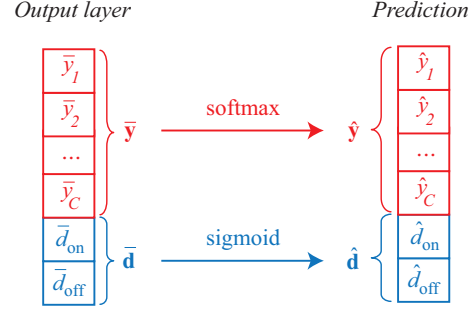


Fig. 3. The output layer and the prediction of the multi-task DNN.

and offset [16, 10]. The distances are further normalized to $[0, 1]$ by dividing by their maximum values in the training data.

The output layer of DNN-2 consists of two variables: $\bar{\mathbf{y}} = (\bar{y}_1, \bar{y}_2, \dots, \bar{y}_C)$ and $\bar{\mathbf{d}} = (\bar{d}_{on}, \bar{d}_{off})$ as illustrated in Fig. 3. The network predictions for class posterior probability $\hat{\mathbf{y}} = (\hat{y}_1, \hat{y}_2, \dots, \hat{y}_C)$ and distance vector $\hat{\mathbf{d}} = (\hat{d}_{on}, \hat{d}_{off})$ are then obtained by:

$$\hat{\mathbf{y}} = \text{softmax}(\bar{\mathbf{y}}), \quad (2)$$

$$\hat{\mathbf{d}} = \text{sigmoid}(\bar{\mathbf{d}}). \quad (3)$$

Given a training set $\{(\mathbf{x}_1, \mathbf{y}_1, \mathbf{d}_1), \dots, (\mathbf{x}_N, \mathbf{y}_N, \mathbf{d}_N)\}$ of N examples, DNN-2 is trained to minimize the multi-task loss:

$$E_{mt}(\boldsymbol{\theta}) = \lambda_{\text{class}} E_{\text{class}}(\boldsymbol{\theta}) + \lambda_{\text{dist}} E_{\text{dist}}(\boldsymbol{\theta}) + \lambda_{\text{conf}} E_{\text{conf}}(\boldsymbol{\theta}) + \frac{\lambda}{2} \|\boldsymbol{\theta}\|_2^2, \quad (4)$$

where

$$E_{\text{class}}(\boldsymbol{\theta}) = -\frac{1}{N} \sum_{n=1}^N \mathbf{y}_n \log(\hat{\mathbf{y}}_n(\mathbf{x}_n, \boldsymbol{\theta})), \quad (5)$$

$$E_{\text{dist}}(\boldsymbol{\theta}) = -\frac{1}{N} \sum_{n=1}^N \left\| \mathbf{d} - \hat{\mathbf{d}}_n(\mathbf{x}_n, \boldsymbol{\theta}) \right\|_2^2, \quad (6)$$

$$E_{\text{conf}}(\boldsymbol{\theta}) = -\frac{1}{N} \sum_{n=1}^N \left\| \mathbf{y}_n - \hat{\mathbf{y}}_n \frac{I(\mathbf{d}_n, \hat{\mathbf{d}}_n(\mathbf{x}_n, \boldsymbol{\theta}))}{U(\mathbf{d}_n, \hat{\mathbf{d}}_n(\mathbf{x}_n, \boldsymbol{\theta}))} \right\|_2^2. \quad (7)$$

E_{class} , E_{dist} , and E_{conf} in the above equations are so-called *class loss*, *distance loss*, and *confidence loss*, respectively. The terms λ_{class} , λ_{dist} , and λ_{conf} represent the weighting coefficients for three corresponding loss types. The *class loss* complies with the common cross-entropy loss to penalize classification errors whereas the *distance loss* penalizes event onset and offset distance estimation errors. Furthermore, the *confidence loss* penalizes both classification errors and distance estimation errors. The functions $I(\mathbf{d}, \hat{\mathbf{d}})$ and $U(\mathbf{d}, \hat{\mathbf{d}})$ in (7) calculate the intersection and the union of the ground-truth event boundary and the predicted one respectively, given by:

$$I(\mathbf{d}, \hat{\mathbf{d}}) = \min(d_{on}, \hat{d}_{on}) + \min(d_{off}, \hat{d}_{off}), \quad (8)$$

$$U(\mathbf{d}, \hat{\mathbf{d}}) = \max(d_{on}, \hat{d}_{on}) + \max(d_{off}, \hat{d}_{off}). \quad (9)$$

While the network may favour optimizing the class loss or the distance loss to reduce the total loss $E_{mt}(\boldsymbol{\theta})$, the confidence loss encourages it to optimize both losses at the same time. This is expected to accelerate and facilitate the learning process.

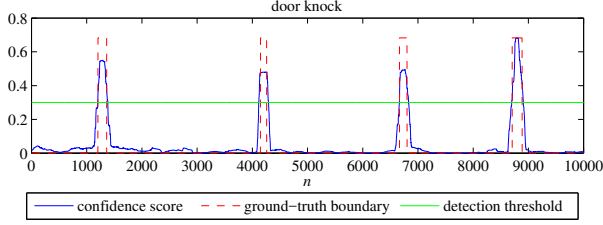


Fig. 4. Alignment of the “door knock” confidence score and the ground-truth boundaries. The regions of confidence score above the threshold are considered as detected “door knock” events.

2.3. Inference

Our prior work [18] employed a simple inference step for event detection, which only used the output of one DNN plus the predicted class labels of a second DNN, combined with median filtering for label smoothing. The estimated event boundaries were completely ignored. Furthermore, that inference scheme could not guarantee reliably early event detection ability since the detection function did not fulfil the monotonicity requirement [11, 10]. In this paper we propose an inference scheme that utilizes all available predicted quantities about target events. We will prove the monotonicity of the detection function, which is key to enabling the network to reliably detect target events early in time.

Let $n, m > 0$ both denote the frame time indices. Given a test audio frame \mathbf{x}_m at the time index m , the time index n is considered to be in the *region of interest* (ROI) of the network prediction if the following condition is fulfilled:

$$m - \hat{d}_{\text{on}}(\mathbf{x}_m) \leq n \leq m + \hat{d}_{\text{off}}(\mathbf{x}_m), \quad (10)$$

where $\hat{d}_{\text{on}}(\mathbf{x}_m)$ and $\hat{d}_{\text{off}}(\mathbf{x}_m)$ represent the event onset and offset distances predicted by DNN-2. Note that these predicted distances need to be restored their original scales beforehand. The confidence score that a target event of class $c \in \{1, 2, \dots, C\}$ occurs at the time index n is then given by,

$$f_c(n | \mathbf{x}_m) = \begin{cases} P_1(1 | \mathbf{x}_m) P_2(c | \mathbf{x}_m) & \text{if (10) holds,} \\ 0 & \text{otherwise.} \end{cases} \quad (11)$$

In (11), $P_1(1 | \mathbf{x}_m)$ represents the posterior probability for \mathbf{x}_m being classified as foreground by DNN-1 and $P_2(c | \mathbf{x}_m)$ denotes the posterior probability for \mathbf{x}_m being classified as the target class c by DNN-2. The confidence score obtained by the network predictions given all audio frames then reads as,

$$f_c(n) = \sum_m f_c(n | \mathbf{x}_m). \quad (12)$$

Fig. 4 demonstrates the confidence score obtained for “door knock” events occurring in a test audio signal of the ITC-Irst dataset [22]. A class-specific threshold β_c is then applied to the confidence score $f_c(n)$ for detection purposes, as demonstrated in Fig. 4.

2.4. Monotonicity of the detection function

The monotonicity of the detection confidence score can be proved easily. Let $f_{\bar{m}}(n)$ denote the accumulated confidence score up to the current time index $\bar{m} > 0$. That is,

$$f_{\bar{m}}(n) = \sum_{m=1}^{\bar{m}} f(n | \mathbf{x}_m), \quad (13)$$

where $f(n)$ is given in (12). Note that we ignore the class label here for simplicity. Formally, we then have,

$$\begin{aligned} f_{\bar{m}}(n) &= \sum_{m=1}^{\bar{m}} f(n | \mathbf{x}_m) \leq \sum_{m=1}^{\bar{m}} f(n | \mathbf{x}_m) + f(n | \mathbf{x}_{\bar{m}+1}) \\ &= \sum_{m=1}^{\bar{m}+1} f(n | \mathbf{x}_m) = f_{\bar{m}+1}(n). \end{aligned} \quad (14)$$

The monotonicity property is guaranteed since, from (11), $f(n | \mathbf{x}_m) \geq 0$ for all $m \geq 0$. The monotonicity can be interpreted as: the more the detector knows about the target event, the higher confidence it gains about occurrence of the target event. As soon as the accumulated confidence score reaches a pre-determined detection threshold, the event is considered detected. It is unnecessary for the system to see the entire event before triggering the detection.

3. EXPERIMENTS

3.1. Experimental setup

Dataset. We conducted experiments on the ITC-Irst dataset of the CLEAR 2006 challenge [22, 23]. The data was recorded in a meeting-room environment with twelve recording sessions in total. There are 16 semantic event categories each of which has approximately 50 occurrences in the recordings. Following the CLEAR 2006 challenge setup [22], twelve out of 16 event categories were evaluated while the rest was considered as background. Furthermore, nine out of twelve recordings were used for training and the remaining three were used for evaluation. Only the single channel named *TABLE.1* was used in the experiments.

Features. An audio signal was decomposed into frames of length 100 ms with a hop size of 10 ms. 64 log-Gammatone spectral coefficients [24] in the frequency range of 50 Hz to 22050 Hz were then extracted for each frame. In addition, we considered a context of five frames for classification purpose. The feature vector for a context window was formed by simply concatenating feature vectors of its five constituent frames.

Parameters. For the weighted loss in (1), we set $\lambda_{\text{fg}} = 2$ and $\lambda_{\text{bg}} = 1$ which are inversely proportional to the ratio of foreground and background examples in the training data. As a result, false negatives are penalized twice as much as false positives. The associated weights of the multi-task loss in (4) were set to $\lambda_{\text{class}} = 1$, $\lambda_{\text{dist}} = 2$, and $\lambda_{\text{conf}} = 1$. We set λ_{dist} larger than λ_{class} and λ_{conf} to encourage DNN-2 to focus more on modelling event temporal structures. In addition, we set the regularization parameter $\lambda = 10^{-3}$ for both losses. The DNNs were trained using the *Adam* optimizer [25] with a learning rate of 10^{-4} . DNN-1 was trained for 25 epochs with a batch size of 256 whereas DNN-2 was trained for 25 epochs with a batch size of 128.

The detection confidence scores were normalized to $[0, 1]$ and the class-wise detection thresholds were searched in the range of $[0, 1]$ with a step size of 0.1 via 9-fold cross-validation on the training data. Those threshold values which yielded maximum class-specific F1-scores were retained.

Baseline. For comparison, we used the early event detection system based on random regression forests proposed in [10] as the baseline. The setting for the random-forest classification and random regression forests was unchanged from [10] apart for a minor improvement in the inference step. That is, we allowed audio frames to contribute to boundary estimations of all event classes instead of a single class as in [10]. The class-wise contribution was weighted by the posterior probability that the frame is classified into the corresponding class. The baseline was run for five times and its average performance was reported.

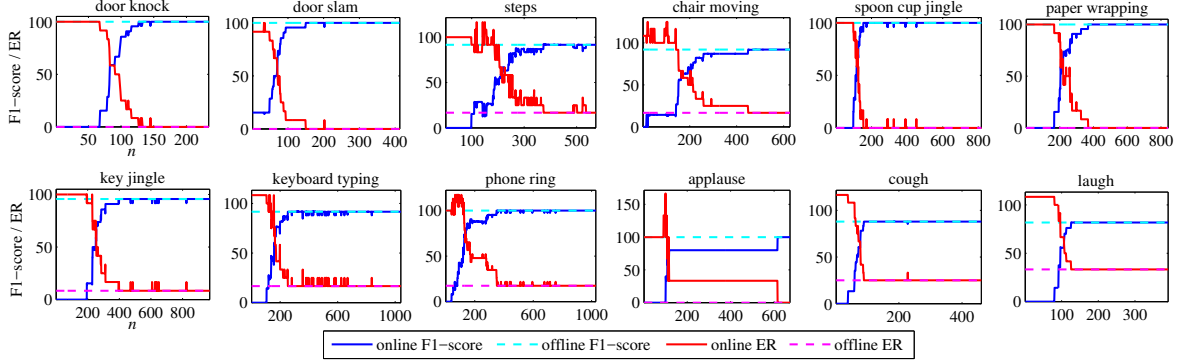


Fig. 5. Development of the online performance curves of different categories. The offline performances are used as reference.

Table 2. ER (%) and F1-score (%) obtained by different detection systems. Bold denotes where the proposed system performed equally to or better than the baseline.

Event Type	ER		F1-score	
	Reg. Forests	DNN	Reg. Forests	DNN
door knock	1.7	0.0	99.2	100.0
door slam	0.0	0.0	100.0	100.0
steps	15.0	16.7	92.5	91.7
chair moving	48.3	16.7	84.4	92.0
spoon cup jingle	3.3	0.0	98.3	100.0
paper wrapping	0.0	0.0	100.0	100.0
key jingle	5.0	8.3	97.7	95.7
keyboard typing	41.7	16.7	81.5	91.7
phone ring	23.5	17.4	89.9	100.0
applause	0.0	0.0	100.0	100.0
cough	16.7	25.0	92.2	88.0
laugh	16.7	33.3	90.9	81.8
Overall	15.1	11.0	93.1	95.2

Evaluation metrics. Two metrics were used for evaluation: event-wise detection error rate (ER) and event-wise detection accuracy in terms of F1-score.

3.2. Experimental results

Offline performance. The detection performance achieved by the proposed system as well as the baseline when entire events are seen by the systems are shown in Table 2. Overall, the proposed system significantly outperforms the regression-forest baseline, improving the F1-score from 93.1% to 95.2% and reducing the ER from 15.1% to 11.0%. Note that the performance of the regression-forest baseline reported here is slightly better than that in [10] due to the improvement in the inference step as mentioned in Section 3.1. The results obtained by the proposed system are even better than those obtained by the regression-forest system with verification (i.e. 94.6% on F1-score [26]).

In term of class-wise performance, the proposed DNN system performed reasonably well on most of the target event categories even though the employed DNN architecture is relatively simple. Interestingly, events that involve the human vocal tract were the exception, which we postulate is due to the lack of feature invariance caused by vocal tract length variation. A more complex neural network, e.g. a CNN [18], would likely circumvent this issue. However, fine-tuning the network architecture is not the main focus of this work, it is primarily the early detection capability.

Online performance. To verify the early detection ability of the proposed system, as in [10], a test audio stream was simulated as a sequence of audio frames coming to the system sequentially one-by-one. As a new event frame was available, the detection performance was re-evaluated and recorded. The offline performance was used as reference. Fig. 5 illustrates how the online performance curves develop as functions of the number of observed event frames. As expected, for all categories, as more event frames are seen by the system, the online F1-scores continually increase while the online ERs scores decrease, until both match the offline scores. More importantly, the online curves always reach the offline ones before the events end, meaning that those events detectable by the system are always detected before they finish. Consider the “laugh” category as an example; about 50% of events are correctly detected within the first 100 audio frames (equivalent to 1.0 seconds). The curve reaches the offline F1-score (i.e. 81.8%) after observing about 120 frames (equivalent to 1.2 seconds). As “laugh” events last for approximately 400 frames, the online system needs less than 30% of the event intervals to achieve the same detection accuracy as the offline system.

Discussion. One can see small fluctuations of the online F1-score and ER curves from Fig. 5. This does not mean that confidence scores themselves fluctuate. In fact, this highlights the way an audio event is considered to be detected: the center of the detected event must fall inside the corresponding ground-truth event and vice versa. Furthermore, with the proposed inference scheme, the region of segmented events is quite narrow, particularly at their early stages. As a result, they can be off-center with respect to the ground-truths and hence counted as detection errors. This is valid for applications in which event segmentation is unimportant, but does lead to room for further investigation into better segmentation strategies.

4. CONCLUSIONS

We presented an AED system which is able to infer ongoing events in audio streams and reliably detect them at their early stages. The key components of the proposed system are a pair of tailored-loss DNNs coupled with a novel inference scheme. The weighted-loss DNN was designed to cope with unbalanced foreground/background classification while the multitask-loss DNN was encouraged to jointly model event class distribution and event boundary estimation. Finally, the inference step made use of the network outputs to compute a confidence score that a target event is occurring at a certain time index. The monotonicity of the detection function, needed for reliable early detection, was also proved. Experiments on the standard ITC-Irst dataset yielded not only state-of-the-art detection performance, but also a demonstration of reliable early detection abilities of the proposed system.

5. REFERENCES

- [1] I. McLoughlin, H. Zhang, Z. Xie, Y. Song, and W. Xiao, “Robust sound event classification using deep neural networks,” *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 540–552, 2015.
- [2] I. McLoughlin, H. Zhang, Z. Xie, Y. Song, W. Xiao, and H. Phan, “Continuous robust sound event classification using time-frequency features and deep learning,” *PLoS ONE*, vol. 12, no. 9, 2017.
- [3] E. Çakir, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, “Convolutional recurrent neural networks for polyphonic sound event detection,” *IEEE/ACM Trans. on Audio, Speech, and Language Processing (TASLP)*, vol. 5, no. 6, pp. 1291–1303, 2017.
- [4] N. Takahashi, M. Gygli and B. Pfister, and L. Van Gool, “Deep convolutional neural networks and data augmentation for acoustic event recognition,” in *Proc. INTERSPEECH*, 2016, pp. 2982–2986.
- [5] H. Phan, L. Hertel, M. Maass, and A. Mertins, “Robust audio event recognition with 1-max pooling convolutional neural networks,” in *Proc. Interspeech*, 2016, pp. 3653–3657.
- [6] “DCASE2016: Detection and classification of acoustic scenes and events,” <http://www.cs.tut.fi/sgn/arg/dcase2016/>.
- [7] “DCASE2017: Detection and classification of acoustic scenes and events,” <http://www.cs.tut.fi/sgn/arg/dcase2017/>.
- [8] 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 *Proc. the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*, 2017.
- [9] A. Mesaros, T. Heittola, and T. Virtanen, “TUT database for acoustic scene classification and sound event detection,” in *Proc. EUSIPCO*, 2016.
- [10] H. Phan, M. Maass, R. Mazur, and A. Mertins, “Early event detection in audio streams,” in *Proc. ICME*, 2015, pp. 1–6.
- [11] Minh Hoai and Fernando De la Torre, “Max-margin early event detectors,” *International Journal of Computer Vision*, vol. 107, no. 2, pp. 191–202, 2014.
- [12] P. Laffitte, D. Sodoyer, C. Tatkeu, and L. Girin, “Deep neural networks for automatic detection of screams and shouted speech in subway trains,” in *Proc. ICASSP*, 2016, pp. 6460–6464.
- [13] A. Kumar and B. Raj, “Deep CNN framework for audio event recognition using weakly labeled web data,” *arXiv:1707.02530*, 2017.
- [14] H. Zhang, I. McLoughlin, and Y. Song, “Robust sound event recognition using convolutional neural networks,” in *Proc. ICASSP*, 2015, pp. 559–563.
- [15] G. Parascandolo, H. Huttunen, and T. Virtanen, “Recurrent neural networks for polyphonic sound event detection in real life recordings,” in *Proc. ICASSP*, 2016, pp. 6440–6444.
- [16] H. Phan, M. Maaß, R. Mazur, and A. Mertins, “Random regression forests for acoustic event detection and classification,” *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 20–31, 2015.
- [17] H. Phan, L. Hertel, M. Maass, P. Koch, and A. Mertins, “CaR-Forest: Joint classification-regression decision forests for overlapping audio event detection,” *arXiv:1607.02306*, 2016.
- [18] H. Phan, M. Krawczyk-Becker, T. Gerkmann, and A. Mertins, “DNN and CNN with weighted and multi-task loss functions for audio event detection,” *arXiv:1708.03211*, 2017.
- [19] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research (JMLR)*, vol. 15, pp. 1929–1958, 2014.
- [20] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proc. CVPR*, 2016, pp. 779–788.
- [21] S. Ruder, “An overview of multi-task learning in deep neural networks,” *arXiv:1706.05098*, 2017.
- [22] A. Temko, R. Malkin, C. Zieger, D. Macho, C. Nadeu, and M. Omologo, “CLEAR evaluation of acoustic event detection and classification systems,” *Lecture Notes in Computer Science*, vol. 4122, pp. 311–322, 2007.
- [23] Christian Zieger and Maurizio Omologo, “Acoustic event detection - ITC-irst AED database,” Tech. Rep., Internal ITC report, 2005.
- [24] D. P. W. Ellis, “Gammatone-like spectrograms,” 2009.
- [25] D. P. Kingma and J. L. Ba, “Adam: a method for stochastic optimization,” in *Proc. ICLR*, 2015, pp. 1–13.
- [26] H. Phan, P. Koch, F. Katzberg, M. Maass, R. Mazur, I. McLoughlin, and A. Mertins, “What makes audio event detection harder than classification?,” in *Proc. EUSIPCO 2017*, 2017.