UNIFYING ISOLATED AND OVERLAPPING AUDIO EVENT DETECTION WITH MULTI-LABEL MULTI-TASK CONVOLUTIONAL RECURRENT NEURAL NETWORKS

Huy Phan[*]*, Oliver Y. Chén*[*]*, Philipp Koch*[†]*, Lam Pham*[‡]
Ian McLoughlin[‡]*, Alfred Mertins*[†]*, and Maarten De Vos*[*]

[*] University of Oxford, Department of Engineering Science, UK
[‡] University of Kent, School of Computing, UK
[†] University of Lübeck, Institute for Signal Processing, Germany

ABSTRACT

We propose a multi-label multi-task framework based on a convolutional recurrent neural network to unify detection of isolated and overlapping audio events. The framework leverages the power of convolutional recurrent neural network architectures; convolutional layers learn effective features over which higher recurrent layers perform sequential modelling. Furthermore, the output layer is designed to handle arbitrary degrees of event overlap. At each time step in the recurrent output sequence, an output triple is dedicated to *each* event category of interest to jointly model event occurrence and temporal boundaries. That is, the network jointly determines whether an event of this category occurs, and when it occurs, by estimating onset and offset positions at each recurrent time step. We then introduce three sequential losses for network training: multi-label classification loss, distance estimation loss, and confidence loss. We demonstrate good generalization on two datasets: ITC-Irst for isolated audio event detection, and TUT-SED-Synthetic-2016 for overlapping audio event detection.

Index Terms— Audio event detection, isolated sound, overlapping sound, multi-label, multi-task, convolutional recurrent neural network

1. INTRODUCTION

Audio event detection (AED) [1, 2] is an important research area within the wider machine hearing field [3, 4], aiming at determining when and which target events occur in continuous audio. This task has recently attracted significant attention in the research community, manifested by rapidly increasing numbers of participants in related international challenges over the past few years [5]. Ideally, event instances of different categories of interest would occur in isolation so that there is at most one such occurrence at any time point in the signal [6, 7]. However, in practice, they may occur at the same time, leading to partial or full temporal overlap [8, 9], sometimes called polyphonic AED. Due to the mixture of multiple acoustic sources, detection of overlapping events is much more challenging than detection of isolated ones.

Isolated and overlapping AED literature often appear to derive from two separate methodological streams. For the former, there is a large body of work covering different perspectives: noise robustness [10, 7, 11], multichannel and multimodal fusion [12, 13, 14], weak labelling [15, 16], early event detection [17, 18, 19], event detection under scarcity scenarios [20, 21], as well as false positive reduction [22, 7]. Particularly, the multitasking approach that jointly performs event detection and event boundary estimation [23, 6, 24] has demonstrated state-of-the-art performance on different benchmark datasets. In the latter stream, overlapping events are either separated using source separation methods [25, 26] prior to detection, or recognized via a selection of local spectral features [11, 10, 7]. The most successful approach appears to be to directly classify the mixtures via multi-label classification [27, 28, 29, 8, 9]. But both streams have one aspect in common: they are efficient when coupled with underlying deep learning models [8, 30, 6, 20], particularly convolutional recurrent neural networks (CRNN) [8, 30]. This is partly due to their power in feature learning and partly due to their capability in performing complex modelling tasks, i.e. multi-label and multi-task. However, there exists a methodological gap between them. Audio events intrinsically possess temporal structures, and tailoring a network's output layer and loss functions for structure modelling has been shown to be efficient for the isolated AED [23, 6, 24]. However, this capacity has been uncharted for overlapping AED, and it remains questionable how to generalize a network's output layer and tailor its loss functions [20, 6] to accommodate arbitrary event overlap, i.e. from one to the maximum number of simultaneous or partially-simultaneous target events. Bridging this gap would allow us to unify how isolated and overlapping AED is trained and operated.

To this end, we present a multi-label multi-task CRNN framework to homogeneously deal with isolated and overlapping events. The network body makes use of a CRNN architecture as it has been shown to be efficient for both isolated [30, 8] and overlapping [8] AED. The idea is to use the convolutional layers to learn good time-frequency invariant features over which recurrent layers are leveraged to incorporate a long temporal context, i.e. hundreds of audio frames. The network sequential output layer is designed to accommodate all possible event overlaps. At each time step of the recurrent output, we tailor a set of output triplets each of which is dedicated to one event category of interest. The output consists of three elements: event activity, event onset distance, and event offset distance, to allow the network to determine whether or not an event of this category is happening at the current time index, and estimate the distances to its boundary, i.e. event onset and offset position, at the same time. As one output triplet is tied to one specific category, inference for all target event categories can be carried out individually no matter how many events of different classes occur concurrently. For training, three types of loss are proposed: sequential multi-label classification loss, sequential distance estimation loss, and sequential confidence loss. Combining the three losses, the network is penalized for both mistakes it makes on event activity determination and event bound-

The research was supported by the NIHR Oxford Biomedical Research Centre. Corresponding author: huy.phan@eng.ox.ac.uk

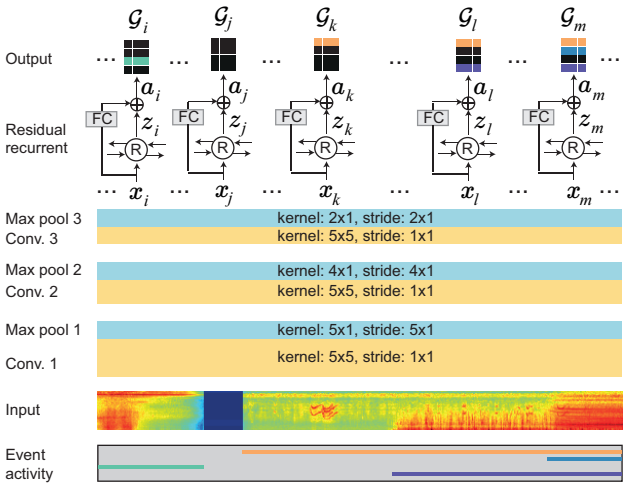

Fig. 1. Overview of the proposed multi-label multi-task CRNN.

ary estimation, integrated over all time steps of the recurrent output layer.

2. THE MULTI-LABEL MULTI-TASK CRNN

The proposed network architecture is illustrated in Fig. 1 and is described in detail in the following sections.

2.1. Input

An audio signal, sampled at 44100 Hz, is converted into a log Mel-scale spectrogram using $M = 40$ Mel-scale filters in the frequency range of [50, 22050] Hz. In addition, a window (i.e. frame) size of 40 ms with 50% overlap is used. C different event categories are considered in total. Since events from any of these categories may happen at a certain audio frame, in order to accommodate all possible occurrences it is necessary to annotate each audio frame with a set of C triplets $\mathcal{G} = \{(y^c, p^c, q^c)\}, 1 \leq c \leq C$, one of which is dedicated for each event category. $y^c \in \{0, 1\}$ equals to one if an event of class c is active at the current audio frame and equals to zero otherwise. $p^c, q^c \in R_+$ denote the distances from the current frame to the event onset and offset if it is active and are normalized to [0,1]. p^c and q^c are forced to be zero when the event is absent.

As a long context is crucial for audio event detection [8, 30], we use an audio segment of $T = 512$ frames as an input to the network. Hence, one sample, i.e. one audio segment, is represented by a time-frequency image $\mathbf{S} \in \mathbb{R}^{M \times T}$ and associated with a sequence of T triplet sets $\mathfrak{G} = (\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_T)$ where $\mathcal{G}_t = \{(y_t^c, p_t^c, q_t^c)\}$, $1 \leq c \leq C$, as described above, denotes the annotation of the frame at the time index t, $1 \leq t \leq T$. The network therefore plays the role of mapping: $\mathbf{S} \mapsto \mathfrak{G}$, which is multi-label (i.e. multiple classes may be active concurrently) and multi-task (i.e. joint modelling of event activity and event boundary).

2.2. Convolutional layers

The convolutional block of the network essentially consists of three convolutional layers, each followed by a max pooling layer. The convolutional layers are commonly designed to have F two-dimensional convolutional filters of size 5×5 with the stride set to one in both temporal and spectral directions during the convolution operation. Zero-padding (also known as *SAME* padding) is used in order to maintain a temporal size equal to T. After convolution, batch normalization [31] is applied on the feature maps, followed by Rectified Linear Units (ReLU) activation [32].

The three max pooling layers aim to improve spectral invariance while keeping the temporal size unchanged. Their pooling kernel size are set to 5×1, 4×1, and 2×1 with stride of 5×1, 4×1, 2×1, respectively. With these settings, the spectral dimension is reduced from an input of $M = 40$ to $8 \rightarrow 2 \rightarrow 1$ after the pooling layers, respectively. Concatenating all F pooled feature maps after the last pooling layer, we have transformed the original input into a convolutional image feature $\mathbf{X} \in \mathbb{R}^{F \times T}$.

2.3. Residual recurrent layer

The above convolutional output $\mathbf{X}$ can be interpreted as a sequence of T convolutional feature vectors, i.e. $\mathbf{X} \equiv (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$ where $\mathbf{x}_t \in \mathbb{R}^F$, $1 \leq t \leq T$. A bidirectional recurrent layer is then used to read the sequence of convolutional feature vectors into the sequence of recurrent feature vectors $\mathbf{Z} \equiv (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_T)$, where

$$\mathbf{z}_t = [\mathbf{h}_t^{\mathrm{b}} \oplus \mathbf{h}_t^{\mathrm{f}}]\mathbf{W}_z + \mathbf{b}_z, \tag{1}$$

$$\mathbf{h}_t^{\mathrm{f}} = \mathcal{H}(\mathbf{x}_t, \mathbf{h}_{t-1}^{\mathrm{f}}), \tag{2}$$

$$\mathbf{h}_t^{\mathrm{b}} = \mathcal{H}(\mathbf{x}_t, \mathbf{h}_{t+1}^{\mathrm{b}}). \tag{3}$$

Here, $\mathbf{h}_t^{\mathrm{f}}, \mathbf{h}_t^{\mathrm{b}} \in \mathbb{R}^H$ represent the forward and backward hidden state vectors of size H at recurrent time step t, respectively while $\oplus$ indicates vector concatenation. $\mathbf{W}_z \in \mathbb{R}^{2H \times 2H}$ denotes a weight matrix and $\mathbf{b}_z \in \mathbb{R}^{2H}$ denotes bias terms. $\mathcal{H}$ represents the hidden layer function of the recurrent layer and we use Gated Recurrent Units (GRU) [33] here to realize the function $\mathcal{H}$.

As a recurrent output vector $\mathbf{z}_t$ is expected to have context information from the entire sequence, to allow the network to explore possible combinations of local convolutional features $\mathbf{x}_t$ and contextual recurrent features $\mathbf{z}_t$, we aggregate them via a residual connection. As $\mathbf{x}_t \in \mathbb{R}^F$ and $\mathbf{z}_t \in \mathbb{R}^{2H}$ may have different sizes in practice, we transfer $\mathbf{x}_t$ through a fully-connected layer with a weight matrix $\mathbf{W}_x \in \mathbf{R}^{F \times 2H}$ and bias term $\mathbf{b}_x \in \mathbf{R}^{2H}$ to make their sizes compatible. The final residual feature vector at time step t is

$$\mathbf{a}_t = \mathrm{ReLU}(\mathbf{x_t}\mathbf{W}_x + \mathbf{b}_x) \oplus \mathbf{z}_t. \tag{4}$$

Batch normalization [31] is also applied to the fully-connected layer of the residual connection.

2.4. Output layer

The output layer consists of $T \times C \times 3$ entries in total which are orderly arranged in the output sequence $\hat{\mathfrak{G}} = (\hat{\mathcal{G}}_1, \hat{\mathcal{G}}_2, \ldots, \hat{\mathcal{G}}_T)$. At time index t, $\hat{\mathcal{G}}_t = \{(\hat{y}_t^c, \hat{p}_t^c, \hat{q}_t^c)\}$, $1 \leq c \leq C$, is the set of C output triplets, one dedicated to each event category. $\hat{y}_t^c$ indicates how likely an event of class c is occurring at t while $\hat{p}_t^c$ and $\hat{q}_t^c$ estimate the distances to its onset and offset from t. To obtain the output $\hat{\mathcal{G}}_t$ at time index t, the residual output $\mathbf{a}_t$ is transferred through a single fully-connected layer with *sigmoid* activation:

$$\mathbf{o}_{\hat{\mathcal{G}}_t} = \mathrm{sigmoid}(\mathbf{a}_t\mathbf{W}_a + \mathbf{b}_a), \tag{5}$$

where $\mathbf{W}_a \in \mathbb{R}^{2H \times 3C}$ and $\mathbf{b}_a \in \mathbb{R}^{3C}$. $\mathbf{o}_{\hat{\mathcal{G}}_t} \in [0, 1]^{3C}$ is the flattened vector including the entries of $\hat{\mathcal{G}}_t$ in a pre-determined order.

2.5. Losses

Similar to [20, 6], for network training, we want to penalize the network on both tasks: event activity determination and event boundary estimation. Assume output sequence $\hat{\mathfrak{G}} = (\hat{\mathcal{G}}_1, \hat{\mathcal{G}}_2, \ldots, \hat{\mathcal{G}}_T)$ is obtained from the network given an input $\mathbf{S}$ associated with groundtruth $\mathfrak{G} = (\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_T)$. Since the event activity determination task is addressed as a multi-label classification problem,

i.e. multiple events of different classes may be present at the same time, the multi-class cross-entropy loss cannot be used here. Therefore, we interpret the multi-label classification problem as multiple binary classification problems and employ *binary* cross-entropy loss penalization. Furthermore, integration over the network's output at different time steps is necessary to take into account all possible misclassifications. The *sequential multi-label classification loss* is

$$E_{\text{class}}(\mathfrak{G},\hat{\mathfrak{G}}) = \frac{1}{T}\sum_{t=1}^{T}\sum_{c=1}^{C}\Big(-y_t^c\log(\hat{y}_t^c) - (1-y_t^c)\log(1-\hat{y}_t^c)\Big). \quad (6)$$

Similarly, the *sequential distance loss* induced by errors in event onset and offset distance estimation is given by

$$E_{\text{dist}}(\mathfrak{G},\hat{\mathfrak{G}}) = \frac{1}{T}\sum_{t=1}^{T}\sum_{c=1}^{C}\Big(\|p_t^c-\hat{p}_t^c\|_2^2 + \|q_t^c-\hat{q}_t^c\|_2^2\Big). \quad (7)$$

The event boundary estimation errors can also be quantified as the $\{\text{intersection} : \text{union}\}$ ratio of the truth boundary and the estimated boundary [20, 6] and further penalized using the *sequential confidence loss*:

$$\begin{aligned} E_{\text{conf}}(\mathfrak{G},\hat{\mathfrak{G}}) &= \frac{1}{T}\sum_{t=1}^{T}\sum_{c=1}^{C}\Big\|y_t^c - \frac{\text{intersection}(\mathfrak{G},\hat{\mathfrak{G}})}{\text{union}(\mathfrak{G},\hat{\mathfrak{G}})}\Big\|_2^2 \\ &= \frac{1}{T}\sum_{t=1}^{T}\sum_{c=1}^{C}\Big\|y_t^c - \frac{\min(p_t^c,\hat{p}_t^c)+\min(q_t^c,\hat{q}_t^c)}{\max(p_t^c,\hat{p}_t^c)+\max(q_t^c,\hat{q}_t^c)}\Big\|_2^2. \quad (8) \end{aligned}$$

Note that we do not use the event activity likelihood to weight the $\{\text{intersection} : \text{union}\}$ ratio as in [20, 6], in to order to more aggressively penalize the network. Finally, the network is trained to minimize the accumulated unweighted multi-task losses over all training examples:

$$E = \sum\nolimits_i E_{\text{class}}(\mathfrak{G}_i,\hat{\mathfrak{G}}_i) + E_{\text{dist}}(\mathfrak{G}_i,\hat{\mathfrak{G}}_i) + E_{\text{conf}}(\mathfrak{G}_i,\hat{\mathfrak{G}}_i). \quad (9)$$

3. INFERENCE

Inference for joint event detection and boundary estimation can be carried out individually for different event categories of interest similar to [18]. However, we need to extend this to handle the sequential output of the network.

Let $m, n > 0$ both denote the time indices of a continuous test signal. Given a test audio segment $\mathbf{S}(m)$ of length T frames starting at the time index m and assuming its output sequence $\hat{\mathfrak{G}}(m)$, its contribution to the confidence score that a target event of class c occurs at time index n is given by

$$f_c\big(n \mid \mathbf{S}(m)\big) = \sum_{t=1}^{T}\hat{y}_t^c(m)\mathbb{1}\Big(\hat{y}_t^c(m) > \alpha_c\Big)\mathbb{1}\Big(n \in \Omega\big(p_t^c(m), q_t^c(m)\big)\Big),$$

where α_c denotes a class-specific threshold on event activity likelihood, $\Omega\big(p_t^c(m), q_t^c(m)\big)$ represents the *region of interest (ROI)* determined by $p_t^c(m)$ and $q_t^c(m)$, and $\mathbb{1}(\cdot)$ is the indicator function. In essence, we iterate over the output sequence $\hat{\mathfrak{G}}(m)$ and collectively integrate the event activity likelihoods into a confidence score. In addition, the event activity likelihood $\hat{y}_t^c(m)$ at time step t of the sequence is only counted if it is greater than the likelihood threshold α_c and where n lies inside the ROI $\Omega\big(p_t^c(m), q_t^c(m)\big)$, meaning

$$m + t - \hat{p}_t^c(m) \le n \le m + t + \hat{p}_t^c(m). \quad (10)$$

Note that the estimated onset and offset distances need to be denormalized to their original range before inference.

The confidence score obtained from all test audio segments sampled from the test signal is

$$f_c(n) = \sum\nolimits_m f_c\big(n \mid \mathbf{S}(m)\big). \quad (11)$$

A second class-specific detection threshold β_c is applied to the confidence score for joint event detection and segmentation. Although we do not study early detection of an ongoing event [17, 18] in this paper, the inference scheme described has such a capability.

4. EXPERIMENTS

4.1. Datasets

We conducted experiments on two datasets:

ITC-Irst [34] —created for studying isolated AED, this database consists of twelve recording sessions with 16 annotated event categories. Following the standard split used in previous works [34, 23, 18], nine out of twelve recordings were used for training and the remaining three were used for evaluation. In addition, evaluation was based on twelve out of 16 categories with the others considered as background sounds. For relevant parameter search (cf. Section 4.2), leave-one-recording-out cross-validation was conducted on the nine training recordings. The channel *TABLE_1* [34] was chosen for the experiments.

TUT-SED-Synthetic-2016 [8] —created for studying overlapping audio event detection, this database consists of 100 mixtures of 994 isolated event instances from 16 event categories. Further detail on the dataset creation procedure can be found in [8]. Out of 100 created mixtures, 60 were used for training, 20 for evaluation, and 20 for validation [8].

4.2. Network training and parameters

To form the training data, we sampled all possible audio segments of length T frames from the training recordings. The network was trained with a minibatch size of four for 100 and 10 epochs for ITC-Irst and TUT-SED-Synthetic-2016, respectively. $F = 256$ convolutional filters were used for the convolutional layers and the size of hidden state vectors of the recurrent layer was $H = 256$. The network was trained using the *Adam* optimizer [35] with learning rate 10^{-4}. For regularization, a dropout rate of 0.25 was applied to the convolutional layers, the recurrent layer, and the residual connection.

Following training, the network was exercised on audio segments sampled from a test signal without overlap to compute the confidence scores as described in Section 3. Particularly, for ITC-Irst, we utilized the cross-validation models for this purpose. The final confidence score on the test signal was averaged from the individual ones resulting from the cross-validation models. The detection confidence score was normalized to [0,1] and the category-specific thresholds α_c and β_c were selected to maximize the average F1-score on the validation set. α_c and β_c were searched in the range [0, 1] with a step size of 0.01 and 0.05, respectively.

4.3. Evaluation metrics

With the proposed multi-label multi-task CRNN coupled with the inference algorithm in Section 3, we are interested in detecting entire events and segmenting them from a continuous test signal. Therefore, we evaluated the detection performance based on two event-wise metrics: F1-score and detection error rate (ER).

4.4. Baseline

In addition to prior works, we implemented a multi-label CRNN baseline for comparison, as it has been demonstrated to achieve state-of-the-art performance on several similar AED datasets [21, 8, 30]. The baseline body architecture and parameters were maintained to be the same as the proposed network, except that its output layer only includes multi-label event activity output. As post-processing

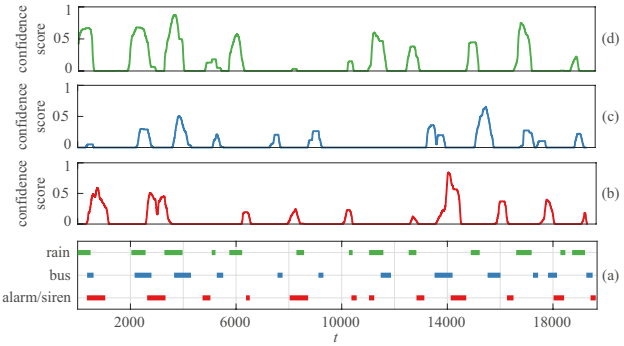

Fig. 2. The confidence scores for three different categories produced by the proposed network on one test recording of the TUT-SED-Synthetic-2016 dataset. (a) event activity, (b) "alarms and sirens", (c) "bus", and (d) "rain".

is important in assisting such a baseline system to yield good performance [21, 9, 29], category-specific likelihood threshold α_c was firstly applied to produce discrete labels which were then smoothed by median filtering. A grid search was conducted for α_c as in Section 4.2, while the window size of the median filter was searched in range of [1, 256] with a step of 6. Those values resulting in the best F1-score on the validation set were retained for evaluation.

4.5. Experimental results

The results obtained by the proposed multi-label multi-task CRNN and the multi-label CRNN baseline on the two experimental datasets are shown in Tables 1 and 2. To further compare with existing works on the ITC-Irst dataset, we also contrast their performance with the best result previously reported in [18] using a dual-DNN approach. On the TUT-SED-Synthetic-2016 dataset, no prior event-wise results were reported, so a CRNN-based system, similar to that reporting best frame- and segment-wise performance [8], is used for a baseline here.

Results show that the proposed system outperforms the baseline on both isolated and overlapping AED tasks. In the isolated AED case with ITC-Irst, an absolute gain of 0.9% was achieved on both average F1-score (i.e. the event *categories* are considered equally important) and overall F1-score (i.e. the event *instances* are considered equally important) although it brings up the overall ER by 0.8% absolute. The rise in ER is mainly due to insertion errors on "key jingle" and "phone ring", which are likely due to their bimodal behaviour. The improvement of the proposed system over the baseline becomes even more noticeable on the overlapping data in TUT-SED-Synthetic-2016. Absolute gains of 1.2% on average F1-score and of 4.1% on overall F1-score were achieved. Moreover, it also improves average and overall ER by 2.3% and 18.7% absolute, respectively. Fig. 2 further shows the confidence scores produced by the proposed system for three different event categories on one recording of TUT-SED-Synthetic-2016. Although the events overlap heavily (and events of other categories were also present but are excluded from the plot for clarity), the proposed network is still able to untangle the mixtures and recognize the individual instances. Meanwhile comparing results in Tables 1 and 2, it is clear that overlapping, or polyphonic, AED remains a more challenging task than isolated AED.

On the other hand, the proposed system significantly outperforms the prior work (i.e. the dual-DNN system [18]) that reported the best results on ITC-Irst. Absolute gains of 2.5% and 2.1% were achieved on average and overall F1-score, respectively, while average ER was lowered by 1.0% absolute.

Table 1. ITC-Irst: the detection performances obtained by the proposed approach and the baseline in comparison with the best reported existing work [18].

Event type	Proposed		Baseline		Best reported (Dual-DNN [18])	
	F1	ER	F1	ER	F1	ER
door knock	100.0	0.0	100.0	0.0	100.0	0.0
door slam	100.0	0.0	100.0	0.0	100.0	0.0
steps	100.0	0.0	100.0	0.0	91.7	16.7
chair moving	92.3	33.3	91.7	33.3	92.0	16.7
spoon cup jingle	100.0	0.0	100.0	0.0	100.0	0.0
paper wrapping	100.0	0.0	100.0	0.0	100.0	0.0
key jingle	95.7	25.0	95.7	8.3	95.7	8.3
keyboard clicking	96.0	8.3	86.7	33.3	91.7	16.7
phone ring	97.4	30.4	98.0	21.7	100.0	17.4
applause	100.0	0.0	100.0	0.0	100.0	0.0
cough	93.8	16.7	92.9	16.7	88.0	25.0
laugh	95.7	8.3	95.7	8.3	81.8	33.3
Average	**97.6**	10.2	96.7	10.1	95.1	11.2
Overall	**97.3**	11.0	96.4	10.2	95.2	11.0

Table 2. TUT-SED-Synthetic-2016: the detection performance obtained by the proposed approach and the baseline system.

Event type	Proposed		Baseline	
	F1	ER	F1	ER
alarms & sirens	72.6	50.4	78.7	37.2
baby crying	58.0	97.8	58.9	93.0
bird singing	63.2	97.5	61.4	89.8
bus	71.1	84.1	62.7	103.7
cat meowing	45.0	130.5	43.8	116.3
crowd applause	51.0	91.9	59.4	91.0
crowd cheering	71.6	49.5	75.2	43.1
dog barking	69.4	72.5	83.4	31.6
footsteps	56.4	99.0	46.9	133.6
glass smash	60.9	118.6	74.7	59.3
gun shot	70.6	72.2	47.7	216.5
horsewalk	60.2	102.3	49.0	110.0
mixer	81.0	50.5	86.6	35.3
motorcycle	49.6	89.9	44.3	94.9
rain	69.8	63.4	76.8	42.0
thunder	72.8	77.8	54.8	86.7
Average	**64.0**	**84.2**	62.8	86.5
Overall	**60.4**	**107.4**	56.3	126.1

5. CONCLUSION

This paper has proposed a multi-label multi-task CRNN in an effort to treat the isolated and overlapping audio event detection tasks homogeneously. Built on top of a CRNN architecture, the recurrent output layer of the network was designed to accommodate arbitrary numbers of overlapping sounds, i.e. from isolated to maximally polyphonic (all event categories occurring simultaneously), at every recurrent time step. For network training, three sequential losses, including the multi-label classification loss, the distance estimation loss, and the confidence loss, were introduced to penalize the network on both multi-label event activity classification errors and event boundary estimation errors. Evaluating on two datasets, namely ITC-Irst for isolated AED and TUT-SED-Synthetic-2016 for overlapping AED, we demonstrated that the proposed network outperforms the multi-class CRNN baseline with the same network body, as well as previously published state-of-the-art results.

6. REFERENCES

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

[2] A. Mesaros, T. Heittola, E. Benetos, P. Foster, M. Lagrange, T. Virtanen, and M. D. Plumbley, "Detection and classification of acoustic scenes and events: outcome of the DCASE 2016 challenge," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 2, pp. 379393, 2018.

[3] D. Wang and G. J. Brown, *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*, Wiley-IEEE Press, 2006.

[4] R. F. Lyon, "Machine hearing: An emerging field," *IEEE Signal Processing Magazine*, vol. 27, no. 5, pp. 131–139, 2010.

[5] "DCASE: Detection and classification of acoustic scenes and events," `http://dcase.community/`.

[6] 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.

[7] 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.

[8] 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*, vol. 5, no. 6, pp. 1291–1303, 2017.

[9] 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.

[10] 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.

[11] J. Dennis, Y. Qiang, T. Huajin, T. H. Dat, and L. Haizhou, "Temporal coding of local spectrogram features for robust sound recognition," in *Proc. ICASSP*, 2013, pp. 803–807.

[12] J. Kürby, R. Grzeszick, A. Plinge, and G. A. Fink, "Bag-of-features acoustic event detection for sensor networks," in *Proc. DCASE Workshop*, 2016, pp. 55–59.

[13] K. Imoto and N. Ono, "Spatial cepstrum as a spatial feature using a distributed microphone array for acoustic scene analysis," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 25, no. 6, pp. 1335–1343, 2017.

[14] H. Phan, M. Maass, L. Hertel, R. Mazur, and A. Mertins, "A multi-channel fusion framework for audio event detection," in *Proc. WASPAA*, 2015, pp. 1–5.

[15] A. Kumar and B. Raj, "Deep CNN framework for audio event recognition using weakly labeled web data," *arXiv Preprint arXiv:1707.02530*, 2017.

[16] Q. Kong, Y. Xu, I. Sobieraj, W. Wang, and M. D. Plumbley, "Sound event detection and time-frequency segmentation from weakly labelled data," *arXiv Preprint arXiv:1804.04715*, 2018.

[17] H. Phan, M. Maass, R. Mazur, and A. Mertins, "Early event detection in audio streams," in *Proc. ICME*, 2015, pp. 1–6.

[18] H. Phan, P. Koch, I. McLoughlin, and A. Mertins, "Enabling early audio event detection with neural networks," in *Proc. ICASSP*, 2018, pp. 141–145.

[19] I. McLoughlin, Y. Song, L. Pham, R. Palaniappan, H. Phan, and Y. Lang, "Early detection of continuous and partial audio events using CNN," in *Proc. Interspeech*, 2018.

[20] H. Phan, M. Krawczyk-Becker, T. Gerkmann, and A. Mertins, "Weighted and multi-task loss for rare audio event detection," in *Proc. ICASSP*, 2018, pp. 336–340.

[21] H. Lim, J. Park, and Y. Han, "Rare sound event detection using 1d convolutional recurrent neural networks," Tech. Rep., DCASE2017 Challenge, 2017.

[22] 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.

[23] 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.

[24] X. Xia, R. Togneri, F. Sohel, and D. Huang, "Random forest classification based acoustic event detection utilizing contextual-information and bottleneck features," *Pattern Recognition*, vol. 81, pp. 1–13, 2018.

[25] T. Heittola, A. Mesaros, T. Virtanen, and M. Gabbouj, "Supervised model training for overlapping sound events based on unsupervised source separation," in *Proc. ICASSP*, 2013.

[26] A. Mesaros, O. Dikmen, T. Heittola, and T. Virtanen, "Sound event detection in real life recordings using coupled matrix factorization of spectral representations and class activity annotations," in *Proc. ICASSP*, 2015.

[27] H. Tran and H. Li, "Jump function Kolmogorov for overlapping audio event classification," in *Proc. ICASSP*, 2011.

[28] 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.

[29] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, "Polyphonic sound event detection using multi label deep neural networks," in *Proc. IJCNN*, 2015.

[30] C.-C. Kao, W. Wang, M. Sun, and C. Wang, "R-CRNN: Region-based convolutional recurrent neural network for audio event detection," in *Proc. Interspeech*, 2018.

[31] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. ICML*, 2015, pp. 448–456.

[32] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. ICML*, 2010.

[33] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in *Proc. EMNLP*, 2014, pp. 1724–1734.

[34] 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.

[35] D. P. Kingma and J. L. Ba, "Adam: a method for stochastic optimization," in *Proc. ICLR*, 2015, pp. 1–13.