Towards duration robust weakly supervised sound event detection

Heinrich Dinkel, *Student Member, IEEE,* Mengyue Wu, *Member, IEEE,* and Kai Yu, *Senior Member, IEEE,*

***Abstract*—Sound event detection (SED) is the task of tagging the absence or presence of audio events and their corresponding interval within a given audio clip. While SED can be done using supervised machine learning, where training data is fully labeled with access to per event timestamps and duration, our work focuses on weakly-supervised sound event detection (WSSED), where prior knowledge about an event's duration is unavailable. Recent research within the field focuses on improving segment- and event-level localization performance for specific datasets regarding specific evaluation metrics. Specifically, well-performing event-level localization work requires fully labeled development subsets to obtain event duration estimates, which significantly benefits localization performance. Moreover, well-performing segment-level localization models output predictions at a coarse-scale (e.g., 1 second), hindering their deployment on datasets containing very short events ($<$ 1 second). This work proposes a duration robust CRNN (CDur) framework, which aims to achieve competitive performance in terms of segment- and event-level localization. In the meantime, this paper proposes a new post-processing strategy named "Triple Threshold" and investigates two data augmentation methods along with a label smoothing method within the scope of WSSED. Event-, Segment-, and Tagging-F1 scores are utilized as primary evaluation metrics. Evaluation of our model is done on three publicly available datasets: Task 4 of the DCASE2017 and 2018 datasets, as well as URBAN-SED. Our model outperforms other approaches on the DCASE2018 and URBAN-SED datasets without requiring prior duration knowledge. In particular, our model is capable of similar performance to strongly-labeled supervised models on the URBAN-SED dataset. Lastly, we run a series of ablation experiments to reveal that without post-processing, our model's localization performance drop is significantly lower compared with other approaches.**

***Index Terms*—weakly supervised sound event detection, convolutional neural networks, recurrent neural networks, semi-supervised duration estimation**

I. Introduction

SOUND event detection (SED) research classifies and localizes particular audio events (e.g., dog barking, alarm ringing) within an audio clip, assigning each event a label along with a start point (onset) and an endpoint (offset). Label assignment is usually referred to as tagging, while the onset/offset detection is referred to as localization. SED can be used for query-based sound retrieval [1], smart cities, and homes [2], [3], as well as voice activity detection [4]. Unlike common classification tasks such as image or speaker recognition, a single audio clip might contain multiple different sound events (multi-output), sometimes occurring simultaneously (multi-label). In particular, the localization task escalates the difficulty within the scope of SED, since different sound events have various time lengths, and each occurrence is unique. Two main approaches exist to train an effective localization model: Fully supervised SED and weakly supervised SED (WSSED). Fully supervised approaches, which potentially perform better than weakly supervised ones, require manual time-stamp labeling. However, manual labeling is a significant hindrance for scaling to large datasets due to the expensive labor cost. This paper primarily focuses on WSSED, which only has access to clip event labels during training yet requires to predict onsets and offsets at the inference stage.

Challenges such as the Detection and Classification of Acoustic Scenes and Events (DCASE) exemplify the difficulties in training robust SED systems. DCASE challenge datasets are real-world recordings (e.g., audio with no quality control and lossy compression), thus containing unknown noises and scenarios. Specifically, in each challenge since 2017, at least one task was primarily concerned with WSSED. Most previous work focuses on providing single target task-specific solutions for WSSED on either tagging-, segment- or event-level. Tagging-level solutions are often capable of localizing event boundaries, yet their temporal consistency is subpar to segment- and event-level methods. This has been seen during the DCASE2017 challenge, where no single model could win both tagging and localization subtasks. Solutions optimized for segment level often utilize a fixed target time resolution (e.g., 1 Hz), inhibiting fine-scale localization performance (e.g., 50 Hz). Lastly, successful event-level solutions require prior knowledge about each events' duration to obtain temporally consistent predictions. Previous work in [5] showed that successful models such as the DCASE2018 task 4 winner are biased towards predicting tags from long-duration clips, which might limit themselves from generalizing towards different datasets (e.g., deploy the same model on a new dataset) since new datasets possibly contain short or unknown duration events. In contrast, we aim to enhance WSSED performance, specifically in duration estimation regarding short, abrupt events, without a pre-estimation of each respective event's individual weight.

II. Related Work

Most current approaches within SED and WSSED utilize neural networks, in particular convolutional neural networks [6], [7] (CNN) and convolutional recurrent neural networks [5], [4] (CRNN). CNN models generally excel at audio tagging [8], [9] and scale with data, yet falling behind CRNN approaches in onset and offset estimations [10].

Apart from different modeling methods, many recent works propose other approaches for the localization conundrum. A plethora of temporal pooling strategies are proposed, aiming to

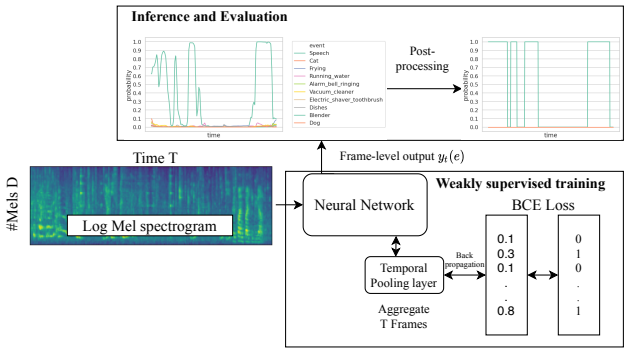


Fig. 1. Basic WSSED framework used in our work. During training only the aggregated per event probabilities are learned via LinSoft, while during inference additional per event time-stamps need to be estimated and enhanced via post-processing.

summarize frame-level beliefs into a single clip-wise probability. McFee *et al.* [7] analyzes the common problem of temporal mean and max pooling for WSSED, indicating that temporal max-pooling preserves event boundaries well; however, it can only backpropagate a single weight. Therefore, [7] proposed soft-max pooling, an interpolation between mean and max pooling, as well as AutoPool, which is a per-event weighted soft-max pooling approach. Subsequent work from Wang *et al.* [11] extends the idea of [7] and compares other similar pooling methods such as attention to two newly proposed approaches, namely linear and exponential softmax. Their results on the DCASE2017 task4 dataset indicate that linear softmax could be potentially superior to attention and max-pooling methods in terms of false positives. By contrast, recent work in [12] reports that temporal max-pooling seems to be advantageous in rare sound event scenarios. However, though different pooling strategies have been proposed, the correlation between the pooling methods and the classifier backends might be under investigation. Attention level pooling methods seem to be preferred for models without sequential model capabilities [6], [13] (e.g., CNN), whereas max and linear softmax pooling functions have seen success in CRNN frameworks [11], [5].

In addition to pooling strategies, a recent popular approach in dealing with different duration lengths is disentanglement. Lin *et al.* [6] presents impressive results on the DCASE2018 task 4 and DCASE19 task 4 datasets, achieving 38.6% and 42.7% Event-F1, respectively, by estimating an individual weight sequence for each individual sound event. Moreover, it is indicated that feature-level aggregation methods (e.g., hidden layer) should be preferred over event-level methods (e.g., output layer). Recent work in [13] proposes multi-branch learning, similar to multi-task learning, which utilizes a multitude of temporal pooling strategies in order to prevent overfitting towards a single method. Similar methods regarding disentanglement are spectral event-specific masking, as introduced in [14]. The main idea is to estimate time-frequency masks via a CNN regarding a target sound event. The advantage of this method is that it can jointly localize as well as filter audio clips.

Lastly, another research category within the field focuses on constructing frame-level losses during training. An example of such a method is [15], which introduces a cosine penalty between different time-event predictions, aiming to enhance the per time step discriminability of each event. This idea is similar to large margin softmax (L-softmax) [16] and has resulted in an F1 score of 32.42% on the DCASE2018 task4 development dataset.

The success of approaches such as [6], [17], [18] relies on prior knowledge about event-wise duration or a target resolution. These methods, in turn, have the following downsides: 1) They are costly since manual labor is required for time-stamp labeling and 2) they rely on consistent duration labeling for each event between available development and unseen evaluation sets, as well as 3) become near impossible to estimate for large quantities of event labels and lastly 4) can only be used for their specific purpose, e.g., tagging, event-level or 1 s segment estimation. Besides, previously introduced models [6], [17], [18], which excel on a specific dataset, have yet to be shown to work on others successfully.

Contribution: In our work, we modify and extend the framework of [5] further towards other datasets and aim to analyze the benefits and the limits of duration robust training. Our main goal with this work is to bridge the gap between real-world SED and research models and facilitate a common framework that works well on both tagging and localization-level without utilizing dataset-specific knowledge. Our contributions are:

- A new, lightweight, model architecture for WSSED using L4-norm temporal subsampling.
- A novel thresholding technique named triple threshold, bridging the gap between tagging and localization performance.
- Verification of our proposed approach across three publicly available datasets, without the requirement of manually optimizing towards dataset-specific hyperparameters.

This paper is organized as follows: In Section III, we state our approach using a CRNN framework for duration robust sound event detection. Further, in Section IV, the experimental setup is described, including our evaluation metrics. Then in Section V we provide our results and compare our approach to other models. Additionally, we include an ablation in Section VI study, which aims to display the duration stability of our model and investigate our model's limitations. Lastly, the paper culminates in Section VII, where a conclusion is drawn.

III. Approach

Contrary to supervised learning approaches, where a (strong) one-to-one correspondence between an instance x_j (here audio frame) and a label $\hat{y}_j$ exists, WSSED is labeled as a (weak) many-to-one relationship. WSSED aims to distinguish between multiple events in an audio clip (multi-class), as well as separate apart co-occurring events (multi-label) at a time. In particular, during training only an entire frame-level feature sequence $[x_1, x_2, \ldots]$ has access to a single, clip-level label $\hat{y}(e) \in \{0, 1\}, e \in [1, \ldots, E]$, where e represents each individual event. This scenario, in its essence, is multiple instance learning (MIL) [19].

It should be evident that within WSSED, localizing an event is much more complicated than tagging, since for tagging, training and evaluation criteria match, while for localization, important information such as duration is missing, meaning that a model needs to "learn" this information without guidance. Due to this mismatch between training and evaluation, post-processing is vital to improve performance [5]. The main idea in our work extends to the notion of duration robustness [5]. We refer to duration robustness as the capability to precisely tag and, more importantly, estimate on- and off-sets of the aforementioned tagged event. A duration robust model should also be capable of predicting on- and off-sets for both short and long duration events.

Our approach's framework can be seen in Figure 1, where the backbone CRNN with temporal pooling and subsampling layers learns clip-level event probabilities during training. For inference, we propose triple thresholding to enhance the duration robustness.

Our proposed model, which we further refer to as CDur (CRNN duration, see Table I), consists of a five-layer CNN followed by a Gated Recurrent Unit (GRU). The model is based on the results in [5], yet it is modified to enhance performance further. Specifically, CDur utilizes double convolution blocks and three subsampling stages instead of five. Moreover, our model experiences substantial gains by utilizing a batch-norm, convolution, activation block structure, especially when paired with Leaky rectified linear unit (LeakyReLU). The abstract representations extracted by the CNN front-end are then processed by a bidirectional GRU with 128 hidden units.

TABLE I
THE CDUR ARCHITECTURE USED IN THIS WORK. ONE BLOCK REFERS TO AN INITIAL BATCH NORMALIZATION, THEN A CONVOLUTION, AND LASTLY, A LEAKYRELU (SLOPE -0.1) ACTIVATION. ALL CONVOLUTIONS USE PADDING IN ORDER TO PRESERVE THE INPUT SIZE. THE NOTATION $t \uparrow / \downarrow d$ REPRESENTS UP/DOWN-SAMPLING TIME DIMENSION BY t AND THE FREQUENCY DIMENSION BY d. THE MODEL HAS TWO OUTPUTS: ONE CLIP-LEVEL, WHICH CAN BE UPDATED DURING TRAINING, AND ONE FRAME-LEVEL USED FOR EVALUATION.

Layer	Parameter	
Block1	32 Channel, 3×3 Kernel	
L4-Sub	$2 \downarrow 4$	
Block2	128 Channel, 3×3 Kernel	
Block3	128 Channel, 3×3 Kernel	
L4-Sub	$2 \downarrow 4$	
Block4	128 Channel, 3×3 Kernel	
Block5	128 Channel, 3×3 Kernel	
L4-Sub	$1 \downarrow 4$	
Dropout	30%	
BiGRU	128 Units	
Linear	E Units	
Output	LinSoft Clip-level	Upsample $4 \uparrow 1$ Frame-level

A. Temporal subsampling

We propose the use of subsampling towards practical, generalized sound event detection, based on the following three reasons: 1) Subsampling discourages disjoint predictions, e.g., short (frame), consecutive zero-one outputs $[0, 1, 0, 1, \ldots] \rightarrow [0, 0, 0, 0, \ldots]$. 2) Reduces the number of time-steps a recurrent neural network (RNN) needs to remember; therefore, it is similar to a chunking mechanism [20], which summarizes parts of a time-sequence. 3) By subsampling in the time-frequency domain via L-norm pooling, abstract time-frequency representations can be learned by the model.

Please note that, in essence, subsampling and pooling are identical operations. In our work, the term subsampling refers to intermediate operations within a local kernel (i.e., 2×2) on a spectrogram's time-frequency domain. In contrast, pooling refers to reducing a time variable signal to a single value (i.e., 500 probabilities $\mapsto$ 1 probability).

Subsampling is commonly done using average or max operators. However, previous work in [5] showed that L-norm subsampling largely benefits duration robustness, specifically enabling the detection of short, sporadic events. L-norm subsampling within a local kernel with size K is defined as:

$$L_p(x) = \sqrt[p]{\sum_{x \in K} x^p},$$

where, L_p reduces to mean subsampling for a norm factor of $p = 0$ and to max subsampling for $p = \inf$. L_p-norm subsampling is an interpolation between mean and max operations, preserving temporal consistency similar to the mean operation while also extracting the most meaningful feature similar to the max operation. In line with [5], we exclusively set $p = 4$ (referred to as L4-Sub) for CDur. An ablation study on the subsampling factor is also provided in Section VI-B.

Subsampling is done in three stages (s_1, s_2, s_3), where each stage subsamples the temporal dimension by a factor of $s_j, j \in [1, 2, 3]$. The time resolution is overall subsampled by a factor of $v = 4$, with the three subsampling stages being $(2, 2, 1)$. At each of the three subsampling stages, the frequency dimension is reduced by a factor of 4, thus after the last stage the input frequency dimension is reduced by a factor of 64. Additionally, a linear upsampling operation is added after the final parameter layer, which restores the original temporal input resolution $\frac{T}{v} \rightarrow T$. We investigated utilizing upsampling during training and parameterized upsampling operations (e.g., transposed convolutions) but did not observe any performance gains. Due to the above remarks, linear upsampling is only utilized during inference. Therefore, CDur outputs predictions at a resolution of 50 Hz (or 20 ms/frame), enabling short sound detection.

B. Temporal Pooling

CDur outputs a frame-level probability $y_t(e) \in [0, 1]$ for each input feature x_t at time t. With temporal pooling, for event e, its event probability at clip-level $y(e) \in [0, 1]$ is an aggregated probability of frame-level outputs $y(e) = \text{agg}_{t \in T}(y_t(e))$. As introduced, temporal pooling is one of the significant contributors to tagging and localization performance. In our work, we exclusively use linear softmax (LinSoft), introduced in [11] (Equation (1)).

$$y(e) = \frac{\sum_t^T y_t(e)^2}{\sum_t^T y_t(e)} \tag{1}$$

In contrast to the attention pooling approach [8], [9], [6], [17], as well as AutoPool [7], LinSoft is a weighted average

algorithm that is not learned. Instead, it can be interpreted as a self-weighted average algorithm.

CDur has two outputs: 1) The clip-level aggregate $y(e)$ and 2) The frame-level sequence prediction $y_t(e)$. Since only clip-level labels are provided during training, only the clip-level aggregate $y(e)$ can be back-propagated and model parameters updated. The frame-level output $y_t(e)$ is solely utilized for evaluation.

C. Post-processing

Since WSSED is a multi-label multi-class classification task, with overlapping events, post-processing is required during inference to transform a per event probability estimate $y_t(e)$ to a binary representation $\bar{y}_t(e)$, which determines whether a label is present ($\bar{y}_t(e) = 1$) or absent ($\bar{y}_t(e) = 0$). It can help smoothen or enhance model performance by removing noisy outputs (e.g., single frame outputs).

We categorize post-processing algorithms into two branches: 1) Probability-level post-processing such as double and triple thresholding; 2) Hard label post-processing such as median filtering. Most post-processing methods, such as median filtering or double thresholding, only consider frame-level outputs for post-processing. In our work, we propose a novel triple thresholding technique that considers clip-level and frame-level outputs. Though this work exclusively utilizes double and triple thresholding, we compare it with median filtering since it is the most common post-processing algorithm for WSSED.

Double Threshold: Double threshold [5], [14] is a probability smoothing technique defined via two thresholds $\phi_{\text{hi}}, \phi_{\text{low}}$. The algorithm first sweeps over an output probability sequence and marks all values larger than ϕ_{hi} as being valid predictions. Then it enlarges the marked frames by searching for all adjacent, continuous predictions (clusters) being larger than ϕ_{low}. An arbitrary point in time t belongs to a cluster between two time steps t_1, t_2 if the following holds:

$$\text{cluster}(t) = \begin{cases} 1, & \exists (t_1, t_2) \text{ s.t., } t_1 \leq t \leq t_2 \\ & \text{and } \forall t_0 \in [t_1, t_2], \phi_{\text{low}} \leq y_{t_0} \leq \phi_{\text{hi}} \\ 0, & \text{otherwise.} \end{cases}$$

The double threshold algorithm ($dt(\cdot)$) can be seen in Equation (2).

$$\bar{y}_t(e) = dt(y_t(e)) = \begin{cases} 1, & \text{if } y_t(e) > \phi_{\text{hi}} \\ 1, & \text{if } y_t(e) > \phi_{\text{low}} \\ & \text{and cluster}(t(e)) = 1 \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

One potential benefit of double thresholding is that its parameters are less susceptible to duration variations and can effectively remove additional post-training optimization (e.g., window size for median filter) [5]. However, double thresholding relies on robust predictions from the underlying model, since it cannot, contrary to median filtering, remove erroneous predictions (e.g., $y_t(e) > \phi_{\text{hi}}$, see Section VI).

Triple Threshold: One possible disadvantage of double thresholding is that it does not consider the clip-level output probability. We propose triple thresholding, which extends towards double thresholding by incorporating an additional threshold on clip-level ϕ_{clip}. This threshold firstly removes all frame-level ($y_t(e)$) probabilities which were not predicted on clip-level ($y(e)$). Triple thresholding is defined as Equation (3).

$$\begin{aligned} y_t(e) &= \begin{cases} y_t(e), & \text{if } y(e) > \phi_{\text{clip}} \\ 0, & \text{otherwise} \end{cases} \\ \bar{y}_t(e) &= dt(y_t(e)) \end{aligned} \quad (3)$$

Therefore, we first remove events not predicted by the model on clip level before proceeding with double threshold post-processing. The default ϕ_{clip} threshold is set to be 0.5, since this is also the most reasonable choice when assessing tagging performance. Experiments denoted as *+Triple*, utilize $\phi_{\text{clip}} = 0.5, \phi_{\text{low}} = 0.2, \phi_{\text{hi}} = 0.75$.

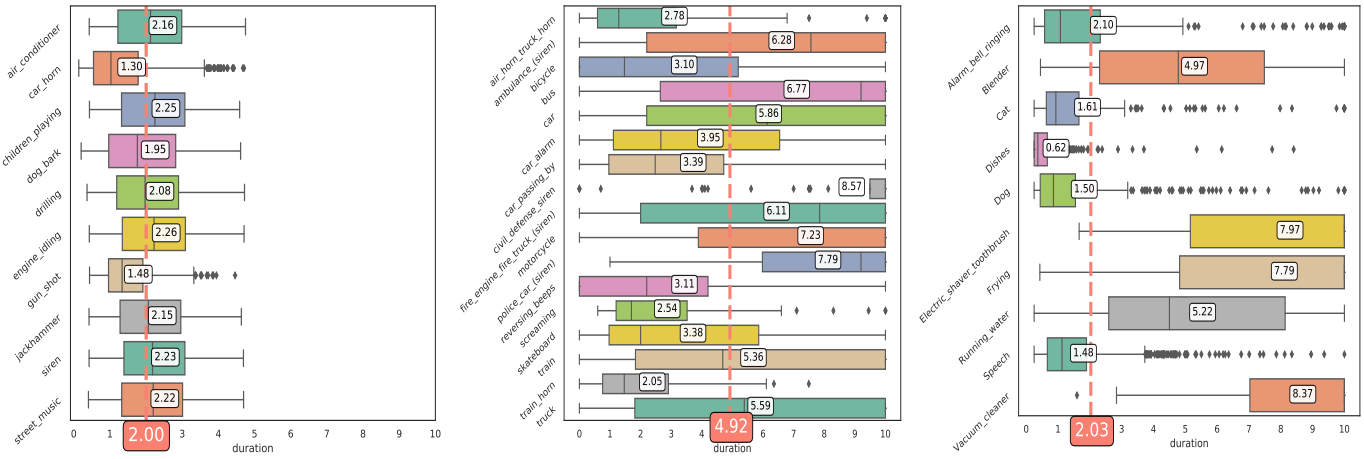
Median Filtering: Median filtering is a conventional hard label post-processing method, meaning it is applied after a thresholding operation. Let $Y = (y_1(e), \ldots, y_t(e), \ldots, y_T(e))$ be a probability sequence for event e, ϕ_{bin} be a threshold and let its corresponding binary sequence be $\bar{Y} = (\bar{y}_1(e), \ldots, \bar{y}_t(e), \ldots, \bar{y}_T(e))$ with the relation:

$$\bar{y}_t(e) = \begin{cases} 1, & \text{if } y_t(e) > \phi_{\text{bin}} \\ 0, & \text{otherwise.} \end{cases}$$

A median filter then acts on the sequence $\bar{Y}$ by computing the median value within a window of size ω. In practice, a median filter will remove any event segments with frame duration $\leq \lfloor \frac{\omega}{2} \rfloor$ and merge two segments with distance $\leq \lfloor \frac{\omega}{2} \rfloor$. In our view, median filtering skews model performance since it can delete or insert non-existing model predictions, thus making fair model comparisons unfeasible. Many successful models utilize an event-specific median filter, where $\omega(e)$ is estimated relative to each event's duration within the labeled development set.

IV. Experimental Setup

All deep neural networks were implemented in PyTorch [21], front-end feature extraction utilized librosa [22] and data pre-processing used gnu-parallel [23]. Even though a plethora of front-features exist, log-Mel spectrograms are most commonly used by the SED community due to their low memory footprint (compared to spectrograms) and excellent performance [24]. If not further specified, all our experiments use 64-dimensional log-Mel power spectrograms (LMS) front-end features. Each LMS sample was extracted by a 2048 point Fourier transform every 20 ms with a Hann window size of 40 ms using the librosa library [22]. During training, zero padding to the longest sample-length within a batch is applied, whereas a batch-size of 1 is utilized during inference, meaning no padding. A batch-size of 64 is utilized during training for all experiments. Moreover, training uses AdamW [25], [26] optimization with a starting learning rate of 1e-4, and successive learning rate reduction if the cross-validation loss did not improve for three epochs. Training



(a) URBAN-SED (b) DCASE2017 Task 4 (c) DCASE2018 Task 4

Fig. 2. Evaluation data duration boxplot distributions for the URBAN-SED and DCASE2017,18 datasets. Each whisker represents the minimum and maximum duration to lie within 99.3% of a normal distribution. Each dot represents an outlier, while boxes represent the median and likely range of standard deviation within the first and third quantile. The mean duration for each event is highlighted individually (white boxes) as well as the global average (red box).

was terminated if no loss improvement has been seen for seven epochs. The available training data was split into a label-balanced 90% training and a 10% held-out validation set for model training using stratification [27]. Furthermore, we utilized a custom sampling strategy such that each batch contained at least a single instance of each event. Note that for all experiments, we neglected the development subset of each dataset. All pseudo-random seeds for each experiment (model initialization, train/cv split, the order of batches) were fixed and, therefore, reproducible. The model contained overall 681,068 trainable parameters, having a size of 2.7 megabytes on disk, making it lightweight and possible to be deployed on embedded systems. The source code is available.[1]

A. Dataset

This paper uses three widely researched datasets: URBAN-SED, DCASE2017 Task 4 and DCASE2018 Task 4. Each evaluation data length distribution can be seen in Figure 2. Please note that estimating events with a long duration (e.g., 10 s) is equal to audio tagging. We additionally provide the number of events per clip for each evaluation dataset in Figure 3.

URBAN-SED: URBAN-SED [28] is a sound event detection dataset within an urban setting, having $E = 10$ event labels (see Figure 2a). This dataset's source material is the UrbanSound8k dataset [29] containing 27.8 hours of data split into 10-second clip segments. The URBAN-SED dataset encompasses 10,000 soundscapes generated using the Scaper soundscape synthesis library [28], being split into 6000 training, 2000 validation and 2000 evaluation clips. The training set contains mostly 10-second excerpts, which are weakly-labeled, whereas each clip contains between one and nine events. The evaluation dataset is strongly labeled, containing up to nine events per clip, as indicated in Figure 3. In our work, we only utilize the training and evaluation dataset and neglect the validation one. An essential characteristic of URBAN-SED is that since both the audio and annotations were generated computationally, the annotations are guaranteed to be correct and unbiased from human perception. As it can be seen in Figure 2a, the evaluation set of the URBAN-SED dataset seems to be artificially truncated due to the soundscape composition.

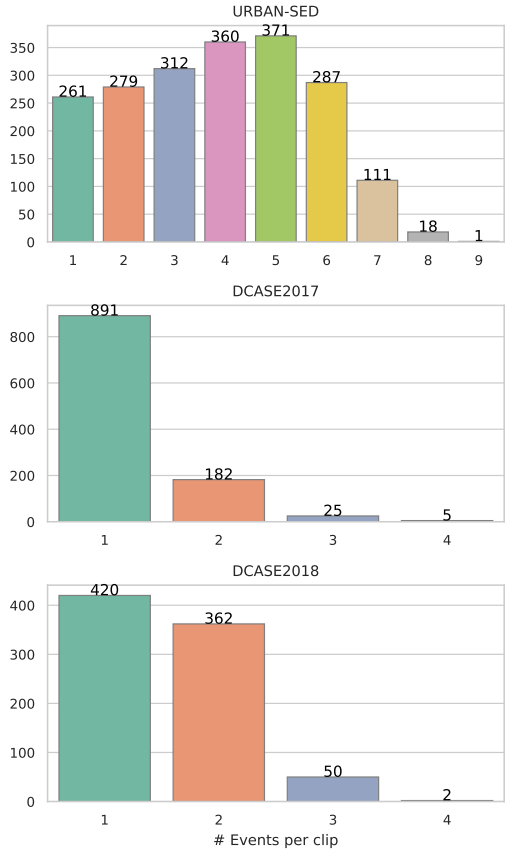


Fig. 3. Number of events in a clip for each evaluation dataset.

[1]Source code is available https://github.com/RicherMans/CDur

DCASE2017: The DCASE2017 task 4 – Large-scale weakly supervised sound event detection for smart cars dataset is also utilized. The dataset consists of 10-second clips split into training (51,172 clips), development (488 clips), and evaluation subsets (1103 clips). Development and evaluation sets are strongly labeled, whereas the training set is weakly-labeled. For our work, we only utilize the training and evaluation subsets and neglect the development one. Different from URBAN-SED, DCASE2017 is subsampled from AudioSet [30], whereas $E = 17$ different car-related events need to be estimated. However, as seen from the evaluation distribution in Figure 2b, each event's duration is distributed much more naturally (between 2.5 s and 8.5 s). In particular, the event "civil defense siren" seems to be the event with the most substantial duration variance, having, on average, a duration of 8.57 s, yet some samples have a duration of < 1 s. In our view, the difficulty within this dataset is the ambiguous labels, e.g., "car" and "car passing by" or the four types of sirens. Please note that due to this dataset's naturalness, it might contain non-target events, which URBAN-SED might not. Most clips in this dataset contain one distinct event, as indicated in Figure 3.

DCASE2018: This dataset was utilized during the DCASE2018 Task4 challenge – Large-scale weakly labeled semi-supervised sound event detection in domestic environments and contains $E = 10$ unique events. Different from the other datasets used in our work, the training data is partially unlabelled/incomplete. While the entire training dataset consists of 55,990 clips, being similarly sized as the DCASE2017 dataset, only 1578 clips contain labels. The rest 54,412 clips are split into in-domain data (14,413 clips), where it is guaranteed that training events occur within the subset, and out-domain data (39,999 clips), which might contain unknown labels. Even though the data is drawn from AudioSet, the ten events have been manually annotated. The evaluation data duration distribution can be seen in Figure 2c. Most clips in this dataset contain one or two distinct events, as indicated in Figure 3. Compared to other datasets, DCASE2018 has the largest average difference duration-wise between the shortest (dishes, 0.62 s) and longest (vacuum cleaner, 8.37 s) events, respectively. Moreover, it also hosts the largest variance within an event, e.g., Speech can be shorter than 1 s, but also 10 s long, indicated by a large number of outliers (Figure 2c). Our work utilizes the training (weak) subset of the dataset for model estimation if not otherwise specified. It should be noted that the evaluation data is identical to the DCASE2019 challenge, which, in addition to DCASE2018, adds synthetic hard labeled data for training.

B. Data Augmentation

A popular method in combating model overfitting and data sparsity is data augmentation. This work investigates two different data augmentation methods, namely SpecAugment (SpecAug) and Time Shifting, in addition to a modified training criterion (Label Smoothing). Each additional augmentation method is individually provided. Further, *+All* refers to the utilization of all three methods described below (*+SpecAug*, *+LS*, *+Time*).

SpecAug: Recently, SpecAug, a cheap yet effective data augmentation method for spectrograms, has been introduced [31]. SpecAug randomly sets time-frequency regions to zero within an input log-Mel spectrogram with D (here 64) number of frequency bins and T frames. Time modification is applied by masking γ_t times η_t consecutive time frames, where ηt is chosen to be uniformly distributed between $[t_0, t_0 + \eta_{t0}]$ and t_0 is uniformly distributed within $[0, T - \eta_t)$. Frequency modification is applied by masking γ_f times η_f consecutive frequency bins $[f_0, f_0 + \eta_f)$, where η_f is randomly chosen from a uniform distribution in the range of $[0, \eta_{f0}]$ and f_0 is uniformly chosen from the range $[0, D - \eta_f)$. For all experiments labeled *+SpecAug* we set $\gamma_t = 2, \eta_{t0} = 60, \gamma_f = 2, \eta_{f0} = 12$.

Time Shifting: Another beneficial augmentation method for WSSED is time-shifting. The goal is to encourage the model to learn coherent predictions. Effectively, given a clip of multiple audio frames $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_T], \mathbf{X} \in \mathbb{R}^{T \times D}$, time rolling of length η_{sh} will shift (and wrap around) the entire sequence by η_{sh} frames to $\mathbf{X}' = [\mathbf{x}_{\eta_{sh}}, \ldots, \mathbf{x}_T, \ldots, \mathbf{x}_1, \ldots, \mathbf{x}_{\eta_{sh}-1}]$. For each audio-clip, we draw η_{sh} from a normal distribution $\mathcal{N}(0, 10)$, meaning that we randomly either shift the audio clip forward or backward by η_{sh} frames. Time shifting is labeled as *+Time* in the experiments.

Label Smoothing: As our default training criterion binary cross entropy (BCE), defined as:

$$\mathcal{L}(y, \hat{y}) = -\hat{y} \log(y) + (1 - \hat{y}) \log(1 - y), \tag{4}$$

between the clip-level prediction $y(e)$ and the ground truth $\hat{y}(e)$ is utilized. BCE encourages the model to learn the provided training ground truth labels. However, in many real-world WSSED datasets, the ground truth labels are not necessarily correct and open to interpretation (e.g., is child babbling considered Speech?). Since labels in WSSED are often noisy and thus unreliable, the BCE criterion (Equation (4)) wrongly encourages overfitting towards the training ground truth labels, which might, in turn, decrease inference performance. Label smoothing [32] (LS) is a commonly used technique for regularization, which relaxes the BCE criterion to assume the ground truth itself is noisy. Label smoothing modifies the ground truth label ($\hat{y}(e)$) for each event $e \in [1, \ldots, E]$ by introducing a smoothing constant ϵ, as seen in Equation (5).

$$\hat{y}_{\text{LS}}(e) = LS(\hat{y}(e)) = (1 - \epsilon)\hat{y}(e) + \epsilon \frac{1}{E} \tag{5}$$

In our work, we exclusively set $\epsilon = 0.1$ for all experiments labeled *+LS*.

C. Evaluation Protocol

Our work utilizes three evaluation metrics:

1) Audio Tagging F1 score (Tagging-F1). This metric measures the models' capability to correctly identify the presence of an event within an audio clip.
2) Segment-F1 (Seg-F1) score. This metric is an objective measure of a given model's sound localization capability, measured by the segment-level (adjustable) overlap between ground truth and prediction. Seg-F1 cuts an audio

clip into multiple fixed sized segments [33]. Seg-F1 can be seen as a coarse localization metric since precise time-stamps are not required.

3) Event-F1 score. This metric measures on- and off-set overlap between prediction and ground truth thus is not bound to a time-resolution (like Seg-F1). The Event-F1 specifically describes a model's capability to estimate a duration (i.e., predict on- and off-set).

Audio tagging is done by thresholding the clip-level output (after LinSoft) $y(e)$ of each event with a fixed threshold of $\phi_{\text{tag}} = 0.5$ in order to obtain a many-hot vector, which is then evaluated. The threshold ϕ_{tag} is fixed since this work focuses on improving Seg- and Event-F1 performance. Segment and event F1-scores are tagging dependent, meaning that they require at least the correct prediction of an event to be assessed. Thus, we perceive audio tagging as the least difficult metric to improve since its optimization directly correlates with the observed clip-level training criterion. Since our work mainly focuses on duration robust estimation, Event-F1 [33] is used as our primary evaluation metric, which requires predictions to be smooth (contiguous) and penalizes irregular or disjoint predictions. To loosen the strictness of this measure, a flexible time onset (time collar, t-collar) of 200ms, as well as an offset of at most 20% of the events' duration, is considered valid. Furthermore, we use Seg-F1 [33] with a segment size of 1 second. Lastly, in theory, two Tagging-F1 scores exist. One can be retrieved from the frame-level prediction output (i.e., $\max_{1:T} y_t(e)$), while the other can be calculated after temporal pooling (i.e., $y(e)$). We exclusively report the Tagging-F1 score from the clip-level output ($y(e)$) of the model in this work. Note that the clip-level output is unaffected by post-processing.

Since each proposed F1-score metric is a summarization of individual scores, two main averaging approaches exist. Micro scores are averaged across the number of instances (e.g., number of samples), whereas macro scores are averaged across each event (e.g., first compute F1 score on sample basis per event, then compute the average of all scores).

The sed_eval toolbox [33] is utilized for score calculation (Seg-F1 and Event-F1 scores). Each respective dataset is evaluated using the following default evaluation metrics:

DCASE2017: The DCASE2017 challenge originally consisted of two subtasks (tagging and localization). The default evaluation metric on the DCASE2017 dataset is 1 second segment-level micro F1-score. We, therefore, report all our metrics on the micro-level (Event, Segment, Tagging).

DCASE2018: All metrics for the DCASE2018 dataset are macro-averaged, and the primary metric during the challenge was Event-F1.

Urban-SED: The default evaluation metric on this dataset is 1 second segment-level macro F1-scores, thus on average, only two segments need to be estimated for each event (see Figure 2a).

V. Results

In this section, we report and compare our results on the three publicly available datasets (see Section IV-A). If not otherwise specficied, all our reported results in this section utilize by default double thresholding with $\phi_{\text{hi}} = 0.75, \phi_{\text{low}} = 0.2$ and BCE (Equation (4)) as the criterion.

TABLE II
URBAN-SED results using our proposed CDur model. *+All* refers to utilizing all augmentations. F1 scores are macro-averaged. R represents the output time resolution for a respective approach in Hz.

Approach	R	Tagging-F1	Seg-F1	Event-F1
Base-CNN [28]	1	-	56.00	-
SoftPool [7]	2.69	63.00	49.20	-
MaxPool [7]	2.69	74.30	46.30	-
AutoPool [7]	2.69	75.70	50.40	-
Multi-Branch [13]	50	-	61.60	-
Supervised SED [34]	43.1	-	64.70	-
Ours	50	76.09	62.83	19.92
+LS		75.00	61.69	18.74
+Time		76.13	62.61	19.69
+SpecAug		76.49	64.19	20.58
+All		**77.13**	**64.75**	21.73
+All +Triple		**77.13**	**64.75**	**22.54**

A. URBAN-SED

In Table II, we compare previous approaches on the URBAN-SED corpus to our CDur approach.

Our baseline CDur result can be seen to outperform all compared approaches in terms of Tagging-F1. Moreover, its Seg-F1 score is also significantly higher (62.83%) than all other previous WSSED approaches. Even though CDur is capable of estimating clip- and segment-level events, it falls short of providing a competitive Event-F1 performance. We believe that the comparatively low Event-F1 performance stems from the nature of urban auditory scenes, where most events occur in a much more random fashion compared to, e.g., domestic ones. Moreover, the performance discrepancy between Seg-F1 and Event-F1 further exemplifies the difficulty in WSSED to obtain fine-scale onset and offset estimates. When adding additional augmentation methods, the performance further improves (77.13 Tagging-F1, 64.75 Seg-F1). Most notably, our augmented CDur also outperforms the fully-supervised SED system in [34] in terms of Seg-F1 (64.70%). Lastly, by further replacing double thresholding with triple thresholding as the post-processing method, the Event-F1 score improves from 21.73 to 22.54%. We provide a per event breakdown of our model performance in Figure 4 and compare CDur to the next best (supervised SED) model from [34]. As seen, CDur performs evenly across all ten events in terms of Tagging-F1, averaging $\approx$ 70% across most events. A similar observation can also be made regarding the Seg-F1 score, where most events obtain a score of $\approx$ 65%. Even though CDur only slightly outperforms the supervised approach from [34], analyzing the per event Seg-F1 scores reveals that CDur achieves an F1 score of 71.9 compared to 66.0 on the shortest event within the dataset (car_horn, here car). Event-F1 results are also shown to be evenly distributed, reaching from the lowest 12.0% for "dog" to 41.5% for "jackhammer". Future work is still required to exclusively estimate very short events effectively, such as in this dataset.

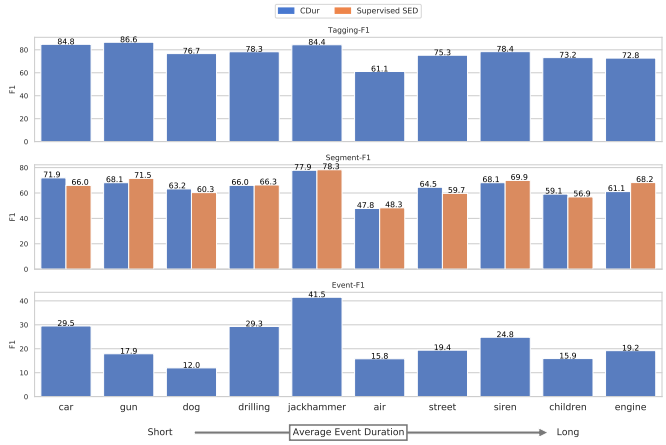


Fig. 4. CDur per event F1 score results on the URBAN-SED dataset. Segment-F1 scores are compared to the next best (supervised) approach [34].The events are sorted from (left) short to (right) long average duration.

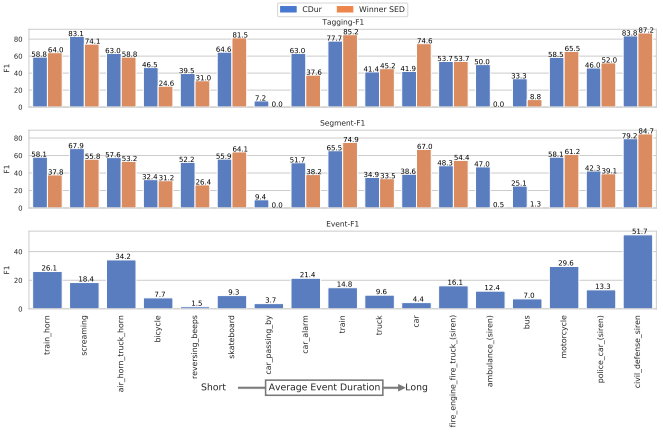


Fig. 5. CDur per event evaluation F1-scores on DCASE2017. The scores are compared to the winning SED model [18] from the DCASE2017 challenge. The events are sorted from (left) short to (right) long average duration.

TABLE III
DCASE2017 RESULTS COMPARED TO OUR APPROACH. F1-SCORES ARE MICRO AVERAGED. BEST RESULTS HIGHLIGHTED IN BOLD. RESULTS UNDERLINED ARE FUSION SYSTEMS. R REPRESENTS THE OUTPUT TIME RESOLUTION FOR A RESPECTIVE APPROACH IN HZ.

Approach	R	Tagging-F1	Seg-F1	Event-F1
MaxPool [7]	2.69	25.70	25.20	-
AutoPool [7]	2.69	45.40	42.50	-
Stacked CRNN [35]	50	43.30	48.90	-
Fusion GCRNN [36]	24	55.60	51.80	-
GCRNN [36]	24	54.20	47.50	-
GCCaps [37]	24	**58.60**	46.30	-
Winner SED [18]	1	52.60	**55.50**	-
Ours	50	52.39	46.12	15.12
+LS		52.07	46.73	15.60
+Time		49.83	46.60	16.15
+SpecAug		55.07	49.94	15.46
+All		55.29	50.79	15.26
+All +Triple		55.29	49.93	15.73

B. DCASE2017

We here provide the results on the DCASE2017 dataset, where all reported metrics are micro averaged. Regarding the DCASE2017 dataset results in Table III, it can be observed that even though the dataset contains the most available training data, Tagging- (52.39%), Seg- (46.12%), and Event-F1 (15.12%) performance is sub-optimal. After adding augmentation to CDur training, Tagging- (55.29%), Seg- (50.79%) and Event-F1 (15.26%) performance significantly increases. However, by replacing double threshold with triple threshold, no gains can be observed. We believe that our performance on this dataset could improve by increasing the number of trainable parameters since the most comparative approach to our in [36] utilized at least six convolutional layers. Moreover, note that our approach is a single model for both Tagging- and Seg-F1 evaluation, while the best comparable model [36] trained two networks with different time resolutions for each respective task. Other approaches are specialized for one individual task (tagging, localization), e.g., the winning localization system [18], a large CNN fusion model, outperforms our method only regarding Seg-F1. This is to be expected, since [18] outputs their predictions on a coarse time resolution of at least 1 s (1 Hz), matching the segment-level criterion. Besides, CDur is the only approach preserving acceptable performance, yet with a high time-resolution of 50 Hz. A closer look at the per event F1 scores in Figure 5 reveals that CDur has difficulties predicting ambiguous events such as "car passing by" and "car". Regarding joint tagging and localization performance, CDur shows promising results for the events "air horn" (2.78 s), "civil defense siren" (8.57 s) and "motorcycle" (7.23 s), achieving $\approx$ 68% Tagging-F1, $\approx$ 65% Seg-F1 and $\approx$ 38.5% Event-F1 average scores. Notably, the best comparable SED model [18] can be seen to miss some events in terms of Seg-F1. Those events are "car_passing_by", "ambulance" and "bus". In stark contrast, CDur's worst Seg-F1 performance is also on "car_passing_by", but CDur still manages a Seg-F1 score of 9.4% (against 0.0%). Compared with other approaches, one distinct feature of CDur is that it does not miss any event, regarding all utilized metrics. Moreover, performance for the shortest duration events "air horn" (2.78 s, Event-F1 34.2%), "train horn" (2.05 s, Event-F1 26.1%), "screaming" (2.54 s, Event-F1 18.4%) further exemplifies our model's capability in detecting short duration events.

C. DCASE2018

For these experiments, we utilize two given DCASE2018 subsets for model training, being the common training subset (weak) and the unlabeled in-domain subset. Weak labels were estimated on the in-domain dataset for further training using our best performing CDur model (*+All*, 70.56 Tagging-F1) by thresholding the clip-level output with a conservative value of $\phi_{\text{tag}} = 0.75$. Since not all clips obtained a label (no event has a probability higher than 0.75), we report that our predicted in-domain subset consists of 9266/14412 clips. Note that other approaches estimating labels for the in-domain data are likely to be different from ours (such as [6], [38], [17], [10]). We verify that our predicted labels are usable by training CDur exclusively on the generated in-domain labels (Table IV). Further, we refer to "Weak+" as the merged dataset of "Weak"

and the predicted "In-domain" data, containing 10844 semi-noisy clips.

TABLE IV
DCASE2018 EVALUATION RESULTS GROUPED BY TRAINING DATA. THE BEST RESULT IS HIGHLIGHTED IN BOLD, AND FUSION APPROACHES ARE UNDERLINED. R REPRESENTS THE OUTPUT TIME RESOLUTION FOR A RESPECTIVE APPROACH IN HZ.

Approach	Data	R	Tagging-F1	Seg-F1	Event-F1
Hybrid-CRNN [10]	Weak+	50	-	-	25.40
Second'18 [38]	Weak+	50	-	-	29.90
Winner'18 [17]	Weak+	16	-	-	32.40
CRNN [5]	Weak	25	-	-	32.50
Multi-Branch [13]	Weak	50	-	-	34.60
cATP-SDS [6]	Weak+	50	65.20	-	38.60
Ours	Weak	50	69.20	59.89	31.70
+All			**70.56**	63.17	35.71
+All +Triple			70.56	62.84	36.23
Ours	In-domain	50	64.68	59.43	31.12
Ours	Weak+	50	67.19	59.85	36.49
+LS			69.63	62.96	36.87
+Time			68.67	62.63	38.03
+SpecAug			69.93	62.94	36.28
+All			69.11	**63.53**	39.18
+All +Triple			69.11	63.03	**39.42**

The results in Table IV indicate our models' superior performance regarding the three evaluation metrics. CDur trained only with weak data (Event-F1 31.70) approaches performance near the winning model of 2018, which utilized additional (weak+) data. Moreover, training our model only on its estimated, noisy labels (in-domain) leads to a similar performance when trained on the weak dataset in terms of Seg- and Event-F1. This result shows that CDur is capable of handling noisy labels to estimate an events' duration successfully. However, Tagging-F1 performance deteriorates from 69.20 to 64.68% when training only on noisy labels. Training CDur on the merged weak+ data further enhances performance in terms of Event-F1 (36.49%) but worsens the Tagging-F1 (67.19%) and Seg-F1 (58.12%) performance. We believe that the additional, noisy data-enhanced onset and offset estimation accuracy, leading to an Event-F1 improvement. However, the inconsistency between clean (weak) and noisy (in-domain) labels possibly confused CDur's internal belief state, resulting in a Tagging-F1 performance downfall. Further, by explicitly modeling all labels as being noisy via label smoothing (weak+ and *+All*), the Tagging-F1 performance significantly improves from 67.19 to 69.11% and returns to original, clean label levels of only using weak data (69.20%). After incorporating our proposed augmentation methods as well as triple thresholding, our best Event-F1 result 39.42% is achieved. This result is remarkable since it would perform at eighth place on the DCASE19 challenge, which has access to hard labels during training, yet unavailable in our case.

It is notable that previously well-performing models [6], [13], [17], [38] all require a fully-labeled development dataset in order to estimate a per event median filter. By contrast, our approach does not rely on such labels yet still achieves competitive results. Note that it would be possible to further enhance our performance by estimating per event thresholds, similar to [17], [6]. However, we refrain from doing so since

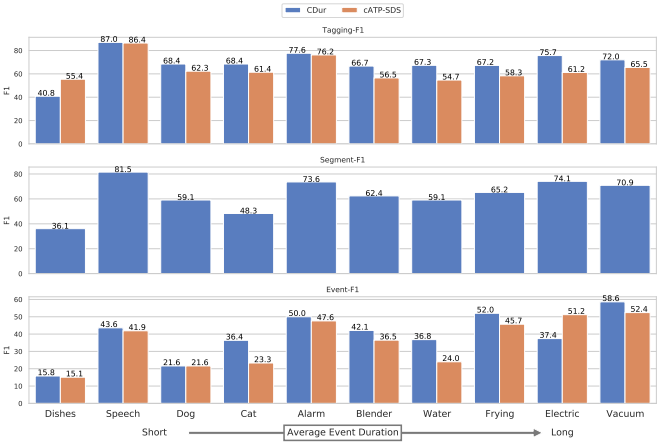
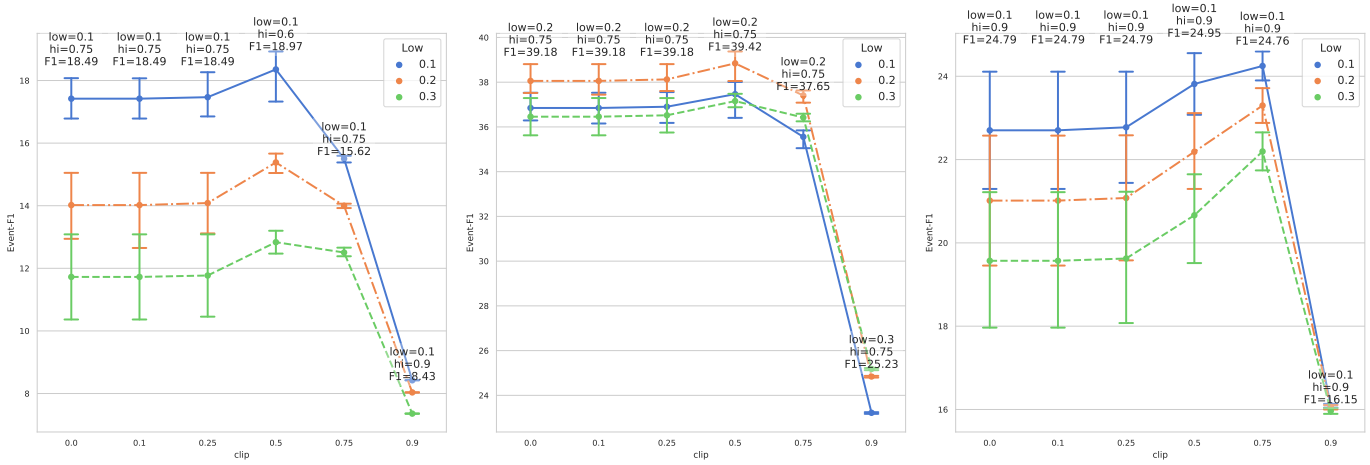


Fig. 6. CDur event evaluation F1-scores on DCASE2018. The provided F1 scores are compared against the competitor (cATP-SDS) from [6]. The events are sorted from (left) short to (right) long average duration.

one of our goals is to propose an inherently well-performing model without post-training hyperparameter tuning. In Section VI we will further discuss this post-processing issue. Furthermore, the per-event F1-scores for our best performing model is shown in Figure 6. Results reveal that CDur is capable of excellent performance across all ten events, excelling at predicting long events such as "vacuum cleaner" (average duration 8.37 s, Event-F1 58.6%) as well as short ones such as "alarm" (average duration 2.1 s, Event-F1 50.0%). Naturally, shorter events (cat, dog, dishes) are more challenging to predict due to their bursty nature, therefore overall, the worst-performing events for our model (1.61 s - 36.4%,1.50 s - 21.6%, 0.62 s - 15.8%). In particular, our model struggles at predicting "dishes" on a clip level (Tagging-F1 40.8%), while all other labels are estimated with $\approx$ 70% Tagging-F1 score. When comparing CDur against cATP-SDS some interesting observations are seen. First, only on the event "Electric", cATP-SDS outperforms CDur in terms of Event-F1 (37.4% vs. 51.2%), whereas for most other events, CDur is better. Second, estimating short events is hard for cATP-SDS. This can be seen by comparing Tagging-F1 with Event-F1 scores. Specifically, even though cATP-SDS outperforms CDur in terms of Tagging-F1 estimating "dishes" (40.8% vs. 55.4%), their Event-F1 scores on this event are near identical (15.8% vs. 15.1%). This means that even though cATP-SDS is more capable of detecting the presence of this short event than CDur, it more often fails to accurately predict on- and off-sets.

D. Performance influence of triple threshold

Another essential question to ask is how the triple threshold behaves when utilizing different thresholds and if it indeed improves performance. Recall that triple thresholding with $\phi_{\text{clip}} = 0$ reduces to double thresholding (Equation (2)). Here we investigate the following thresholds: $\phi_{\text{low}} \in [0.1, 0.2, 0.3], \phi_{\text{hi}} \in [0.5, 0.6, 0.75, 0.9], \phi_{\text{clip}} \in [0.0, 0.1, 0.25, 0.5, 0.75, 0.9]$. The results for each respective dataset in terms of Event-F1 can be seen in Figure 7.



(a) DCASE2017 Task 4 (b) DCASE2018 Task 4 (c) URBAN-SED

Fig. 7. Triple threshold performance in regards to different thresholds for all three datasets. The x-axis represents the clip threshold ϕ_{clip}, the y-axis the Event-F1 score achieved. Each color represents a lower threshold, whereas each bar represents the mean distribution of a result regarding the four utilized high thresholds $\phi_{\text{hi}} \in [0.5, 0.6, 0.75, 0.9]$. The best result for each clip-threshold is displayed. Best viewed in color.

Our proposed setting with $\phi_{\text{clip}} = 0.5$ and $\phi_{\text{hi}} = 0.75, \phi_{\text{low}} = 0.2$ can be seen to indeed perform best on the DCASE2018 dataset (see Figure 7b). However, for the other two datasets, a lower threshold of $\phi_{\text{low}} = 0.1$ seem to be favorable on the DCASE2017 and URBAN-SED datasets, culminating in Event-F1 scores of 18.97 (DCASE2017, micro) and 24.95 (URBAN-SED, macro) respectively. Again, please note that considerable gains can be obtained by optimizing our thresholds if we choose to optimize towards a specific dataset. For example, our best reported DCASE2017 model (micro Event-F1 15.73) can be improved to 18.97, by modifying $\phi_{\text{low}} = 0.1$. The same observation can be made for URBAN-SED (22.54 $\rightarrow$ 24.95).

A clip threshold of $\phi_{\text{clip}} = 0.5$ seems to be a valid choice within our investigated thresholds since that threshold performs best on the DCASE2017/2018 datasets. On the URBAN-SED dataset, larger thresholds with a high variance should be preferred ($\phi_{\text{low}} = 0.1, \phi_{\text{hi}} = 0.9, \phi_{\text{clip}} = 0.75$). However, still the best performing approach (24.95 Event-F1) on the URBAN-SED dataset utilizes $\phi_{\text{clip}} = 0.5$. Correctly, by observing the trend of ϕ_{clip} from left to right (left = double threshold $\phi_{\text{clip}} = 0.0$), one can observe that triple thresholding is effective on all three datasets and improves the average performance up until $\phi_{\text{clip}} = 0.9$ (see straight lines in Figure 7).

VI. Ablation

A. Temporal pooling alternatives to LinSoft

This ablation study investigates the influence of CDur's performance in regards to its temporal pooling layer. The default temporal pooling layer (LinSoft) is replaced by four commonly utilized pooling functions, described in Table V. Soft- and Auto pooling have been introduced in [7]. Note that Auto pooling [7] learns a non-constrained weight parameter $\alpha(e) \in \mathbb{R}$ for each respective event e (initialized as one), whereas Attention pooling [11] uses a per timestep, per event weight $w_t(e) \in [0, 1]$.

TABLE V
Temporal pooling alternatives to LinSoft investigated in this work. Soft- and Max pooling functions are parameter-free, while Auto and Attention pooling are parameterized.

Temporal pooling	Formulation
Soft	$y(e) = \sum_t^T y_t(e) \frac{\exp y_t(e)}{\sum_j^T \exp y_j(e)}$
Max	$y(e) = \max_t y_t(e)$
Auto	$y(e) = \sum_t^T y_t(e) \frac{\exp(\alpha(e) y_t(e))}{\sum_j^T \exp(\alpha(e) y_j(e))}$
Attention	$y(e) = \frac{\sum_t^T w_t(e) y_t(e)}{\sum_j^T w_j(e)}$

The ablation study results can be seen in Table VI. The results show that, indeed, LinSoft is to be seen as the best performing pooling method for CDur. In the case of DCASE2017, Auto and Soft pooling are both the closest competitors to LinSoft. However, on the URBAN-SED dataset, Soft and Auto pooling both completely fail to generate accurate on and offsets, indicated by an Event-F1 of 0.0. Further, while Attention shows competitive performance across all datasets in terms of all metrics, it is still mostly inferior to LinSoft, specifically in terms of Event-F1. Most importantly, across all three proposed datasets, LinSoft is the only temporal pooling method that provides excellent performance, e.g., Auto and Soft pooling outperform Attention on the DCASE2017 dataset but are mainly inferior to Attention on the DCASE2018 and URBAN-SED datasets.

B. Subsampling factor influence

Another ablation study in our work focuses on the subsampling factor v (default $v = 4$) and its implications towards duration robustness and Event-F1 performance. Here we compare $v = 4 \mapsto (2, 2, 1)$, against three other factors ($v = 1 \mapsto (1, 1, 1), v = 2 \mapsto (2, 1, 1), v = 8 \mapsto (2, 2, 2)$).

TABLE VI
COMPARISON OF DIFFERENT TEMPORAL POOLING FUNCTIONS COMPARED TO LINSOFT. DCASE2018 RESULTS ARE TRAINED ON THE WEAK DATASET. POST-PROCESSING IS SET TO THE DEFAULT DOUBLE THRESHOLD, AND NO AUGMENTATION IS APPLIED.

Task	Pooling	Tagging-F1	Seg-F1	Event-F1
DCASE2017	LinSoft	**52.39**	**46.12**	**15.12**
	Auto	47.53	44.25	13.06
	Soft	50.45	44.18	13.19
	Attention	51.04	41.63	8.99
	Max	47.63	32.34	7.39
DCASE2018	LinSoft	**69.20**	**59.89**	**31.70**
	Auto	67.13	58.58	19.90
	Soft	68.01	56.76	17.79
	Attention	66.52	55.85	21.95
	Max	66.01	45.80	17.85
URBAN-SED	LinSoft	**76.09**	**62.83**	**19.92**
	Auto	75.14	38.67	0.00
	Soft	73.87	37.47	0.00
	Attention	74.61	59.89	19.33
	Max	74.50	56.70	16.19

TABLE VII
INFLUENCE OF DIFFERENT SUBSAMPLING FACTORS v ON ALL THREE DATASETS. OUR DEFAULT SUBSAMPLING FACTOR IN OUR WORK IS 4. THE DCASE2018 MODEL IS TRAINED ON THE WEAK TRAINING SET. BEST RESULT FOR EACH RESPECTIVE METRIC IS HIGHLIGHTED IN BOLD.

Task	Factor v	Tagging-F1	Seg-F1	Event-F1
DCASE2017	1	50.33	45.95	9.05
	2	44.50	40.74	11.82
	4	52.39	46.12	**15.12**
	8	**52.72**	**47.22**	14.79
DCASE2018	1	**69.95**	**60.80**	26.71
	2	69.75	60.16	26.88
	4	69.20	59.89	31.70
	8	66.47	58.15	**33.18**
URBAN-SED	1	74.29	60.26	11.86
	2	74.41	60.13	14.70
	4	**76.09**	**62.83**	19.92
	8	75.00	61.18	**20.83**

The temporal subsampling function is the default L4-sub (see Table I). The result of this experiment can be seen in Table VII. For all experiments, a clear trend can be observed that a subsampling factor of one or two perform worse than factors 4 and 8. A larger subsampling factor of 8 is seen to benefit Tagging- and Seg-F1 performance on DCASE2017/8 datasets. These results are in line with our previous observations in [5]. The default subsampling factor of 4, while not always performing better than a factor of 8, can be seen to be a trade-off between tagging and localization performance. Also, even though a subsampling factor of 8 improves Event-F1 performance on the DCASE2018 and URBAN-SED datasets, we believe that a subsampling factor of 4 should be preferred on any unknown dataset due to its robustness to possibly unknown, short events. When comparing the shortest events for each dataset according to the subsampling factor in Table VIII, it can be seen that a factor of 4 is overall the most robust choice on all three datasets. Optimizing the subsampling factor can yield significant performance gains in terms of Event-F1. However, this optimization requires prior knowledge about each event's duration, which might not be available.

TABLE VIII
EVENT-F1 SCORES FOR THE SHORTEST EVENT IN EACH DATASET, BEING DISHES (DCASE2018), CAR_HORN (URBAN-SED) AND TRAIN_HORN (DCASE2017).

Subsampling factor v	Dishes	Car_horn	Train_horn
Avg. Duration	0.62 s	1.30 s	2.05 s
1	13.3	21.8	23.1
2	16.0	31.8	24.9
4	**23.2**	32.0	**28.7**
8	16.2	**33.9**	21.7

C. Model comparison without post-processing

Another critical question is whether our model is inherently capable of producing duration robust (i.e., high Event-F1) predictions or if this is solely due to the applied post-processing approach. Thus, this ablation study focuses on removing any post-processing from a trained model (further denoted as *-Post*) and using binary thresholding $y_t(e) > \phi_{\text{bin}}, \phi_{\text{bin}} = 0.5$ to re-evaluate frame-level predictions. Since most works optimize their post-processing methods towards the given dataset, performance and generalization capability can be skewed towards that specific dataset. We provide this ablation study for two successful models on the URBAN-SED and DCASE2018 datasets, respectively. Note that we do not compare on the DCASE2017 dataset since our proposed model does not outperform contemporary approaches. Moreover, the main objective in the DCASE2017 challenge was to predict on a 1-second scale, meaning that the ablation would compare rough-scale estimates (1 s) with our fine-scale CDur (20 ms), which we believe does not provide further insights into CDur. Thus, we reimplemented the currently best performing model [6] on the DCASE2018 dataset as well as as [7] on the URBAN-SED for comparison of duration robustness.

DCASE2018: As can be seen in Table IX, our reimplemented cATP-SDS model (34.07 Event-F1) is within the scope of their reported average performance [6] as well as better performing in terms of Tagging-F1. Our early stopping and balanced batch sampling training schedule, as well as possibly better noisy in-domain labels, could be the explanation behind this performance improvement. It is also indicated that the choice of post-processing approach is crucial to event-level performance. However, post-processing has little effect on segment-level performance. CDur's performance is affected by the choice of post-processing method (here double/triple thresholding). In absolute, our model drops as much as 12%

TABLE IX
POST-PROCESSING ABLATION (*-Post*) RESULTS COMPARING CATP-SDS WITH CDUR ON THE DCASE2018 EVALUATION DATASET.

Approach	Data	Tagging-F1	Seg-F1	Event-F1
cATP-SDS [6]	Weak+	65.20	-	38.60
Our cATP-SDS [6]	Weak	58.98	55.20	28.65
-Post	Weak	58.98	17.83	3.77
Our cATP-SDS [6]	Weak+	66.11	60.39	34.07
-Post	Weak+	66.11	17.74	2.74
Ours	Weak	69.20	59.89	31.70
-Post	Weak	69.20	62.50	23.57
Ours (Best)	Weak+	69.11	63.03	39.42
-Post	Weak+	69.11	63.97	27.05

in Event-F1 score when removing double/triple thresholding. However, even though event-level performance is negatively affected, segment-level performance is enhanced for both our approaches. This is likely due to default post-processing method using a conservative threshold choice of $\phi_{\text{hi}} = 0.75$ (Seg-Precision 74.51%, Seg-Recall 56.80%), while $\phi_{\text{bin}} = 0.5$ enhances recall performance (Seg-Precision 73.70%, Seg-Recall 58.35%), thus improving Seg-F1. Conservative threshold values have also been reported to impact Event-F1 performance [39] positively. Therefore, we can observe an absolute increase from 0.9 to 3.0% Seg-F1 for our models. In contrast to our approach, cATP-SDS [6] seems to be strongly dependent on the post-processing method. Its event and segment level performance plummets for both weak (28.65 $\rightarrow$ 3.77) and weak+ (34.07 $\rightarrow$ 2.74) training sets. The reason for this behavior is their dependence on clip-level post-processing, which effectively filters out false-positives. Different from cATP-SDS, CDur does not require post-processing in order to effectively localize sounds and estimate each sound's respective duration.

URBAN-SED: Since most previous work, such as [7] only utilized segment-level micro F1 scores. We reimplemented those models to compare the Event-F1 performance to ours. We also utilized the same training schema (Pitch augmentation, binarization post-processing) as in [7]. These results show that our reimplementations perform worse in terms of Tagging-F1 performance, but the Seg-F1 score improves compared to the original publication (see Table II). It can be expected that an improved Seg-F1 can correlate with an improved Event-F1.

TABLE X
POST-PROCESSING ABLATION (*-Post*) RESULTS COMPARING [7] (NO POST-PROCESSING) APPROACHES WITH CDUR ON URBAN-SED. MODELS WITH A "OUR" PREFIX ARE REIMPLEMENTATIONS FOR METRIC COMPARISON PURPOSES.

Approach	Tagging-F1	Seg-F1	Event-F1
SoftPool CNN [7]	63.00	49.20	-
MaxPool CNN [7]	74.30	46.30	-
AutoPool CNN [7]	75.70	50.40	-
Our SoftPool CNN [7]	59.86	45.80	2.45
Our MaxPool CNN [7]	74.34	58.76	13.04
Our AutoPool CNN [7]	72.74	56.68	6.89
Ours (Best)	77.13	64.75	21.73
-Post	77.13	63.86	14.54

Please note that all models reported in [7] only utilize a binary threshold strategy, identical to our (-Post) result, as post-processing. By removing our post-processing, we can compare both approaches from a common point of view. As it can be seen in Table X, CDur is capable of outperforming the best localization model (MaxPool CNN), in terms of segment and Event-F1, as well as capable of outperforming the best tagging model (AutoPool CNN).

D. Quantitative results

Since our metrics are threshold-dependent, meaning that only hard labels were evaluated, we provide a quantitative analysis to strengthen the point of our models' duration robustness. Two distinct clips containing at least three events from DCASE 2018 and URBAN-SED datasets are randomly sampled. Regarding DCASE2018, our sampled clip (bUDfTkJCg0o) contains three different events (speech [green], frying [blue], dishes [red]). We compare the probability outputs $y_t(e)$ of CDur (Event-F1 39.42) against cATP-SDS (Event-F1 34.07). The comparison can be seen in Figure 8. At first glance, it can be seen that both methods are incapable of producing perfect results. Therefore we focus on the errors made by each individual approach. In particular, for the speech event (green), one can notice that cATP-SDS predictions exhibit a peaking behavior with no apparent notion of the speech duration. Therefore, cATP-SDS requires median filtering to remove false-positives and connect (or remove) its disjoint predictions. In contrast, CDur seems to predict onset and offsets accurately without the need for post-processing. This behavior is in accord with our previous observation in Table IX. Additionally, the "dishes" event (a sharp cling sound of a fork hitting porcelain) is hard to estimate for both models. However, cATP-SDS predicts dishes as an always present background noise, meaning that it failed to learn the characteristics of the short and sharp "dishes" sound. Lastly, we also provide five distinct samples for the shortest duration events within DCASE2018 ("dishes", "cat","dog","speech","alarm bell ringing") in Figure 9. These samples further demonstrate that CDur is capable of detecting and accurately predicting short and sporadic events, such as "alarm bell ringing", "cat", and "dishes". Our second sampled clip (soundscape_test_unimodal585) is from the URBAN-SED dataset (see Figure 10). We compare CDur to a reimplementation of [7], using our best performing Event-F1 MaxPool CNN model (Table II). Note that our reimplementation uses our LMS features with 50 Hz (20 ms/frame) frame rate, whereas the original work used 43 Hz (23 ms/frame). At first glance, one can observe that this dataset is indeed artificial since the spectrogram appears to contain only the target events, without any other natural background noises. As this sample shows, the

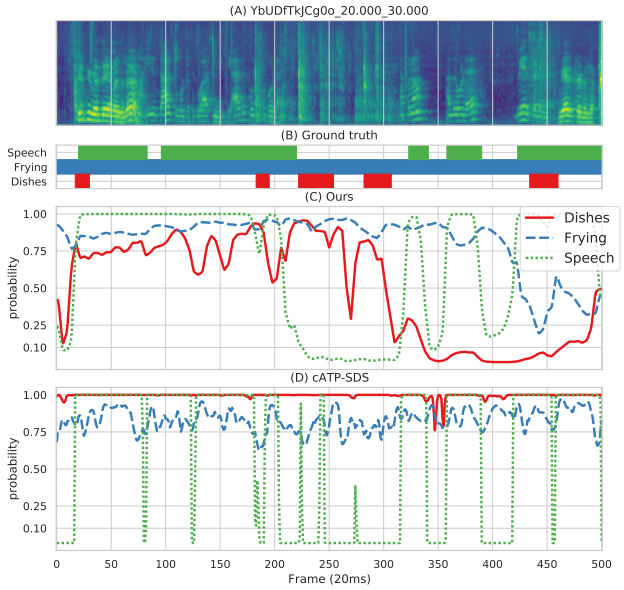


Fig. 8. (A) A sample clip comparison on the (B) DCASE2018 evaluation dataset between (C) CDur and (D) cATP-SDS. Best viewed in color.

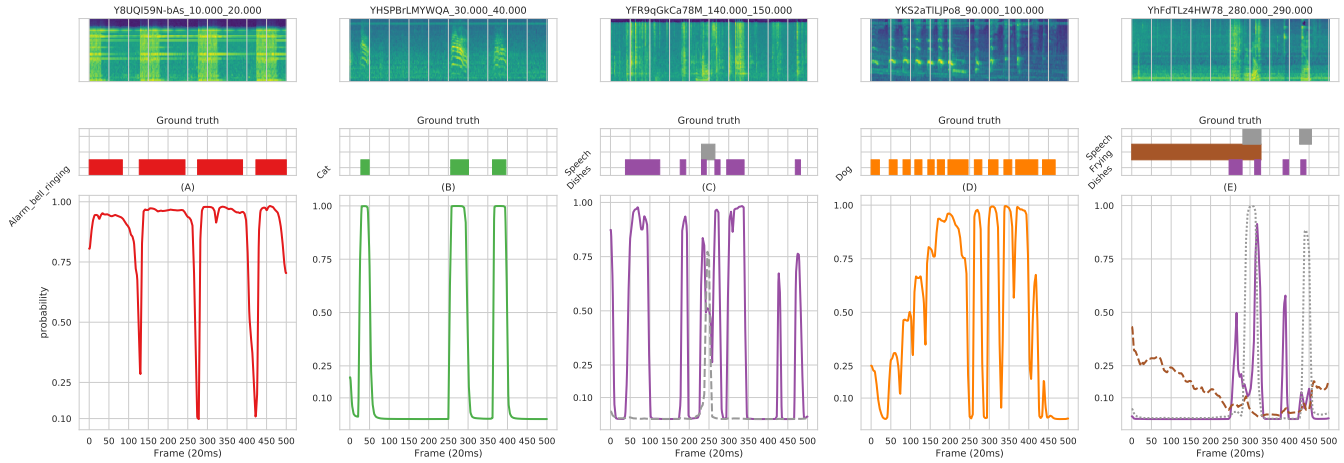


Fig. 9. Predictions for five samples for each of the shortest duration events (“dishes”, “cat”,“dog”,“speech”,“alarm bell ringing”) in DCASE2018. Best viewed in color.

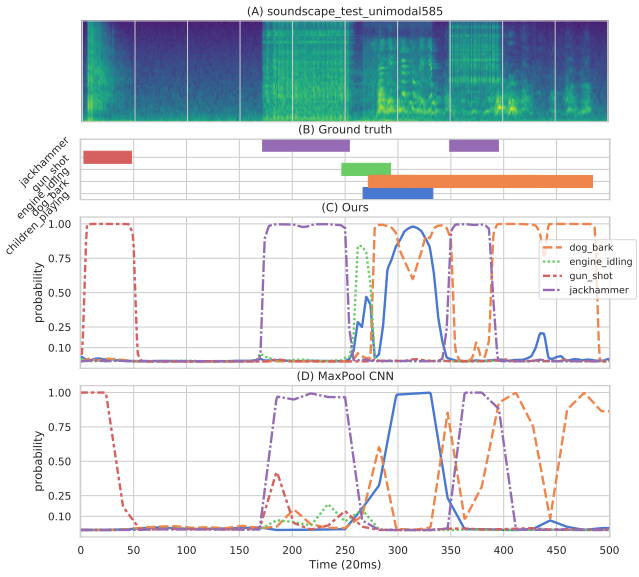


Fig. 10. (A) A sample clip comparison on the (B) URBAN-SED evaluation dataset between (C) CDur and (D) MaxPool CNN. Best viewed in color.

MaxPool CNN model is indeed capable of sound localization, specifically for events with a duration of $\approx$ 1.5 s, such as “jackhammer”, “gun shot” and “children playing”. However, it seems to struggle with longer events, such as “dog bark”, at which it exhibits a peaking behavior, chunking the event into small pieces (see orange line). On the contrary, CDur can predict and localize both short and long events for this sample. Specifically, MaxPool CNN was unable to notice the short “engine idling” event (around 800 ms), yet CDur predicted its presence. We believe that this is due to the low time-resolution of the MaxPool CNN model (320 ms/frame, 3.125 Hz), which could skip over the presence of short events.

VII. Conclusion

This work proposed CDur, a duration robust sound event detection CRNN model. CDur aims to be as flexible as possible in order to be applied across different datasets and scenarios. Further, we propose a new post-processing method called triple thresholding, which not only considers frame-level outputs but also utilizes the clip-level probabilities. Triple thresholding can be seen to improve Event-F1 performance regarding Event-F1 performance on the DCASE2018 and URBAN-SED datasets.

CDur is then compared to other approaches in terms of Segment, Event, and Tagging performance. Experiments conducted on the DCASE2017,18 and URBAN-SED datasets imply promising performance. The DCASE2018 results show that CDur can outperform previous SOTA models in terms of Tagging- (69.11), Event- (39.42), and Seg-F1 (63.53). Besides, on the URBAN-SED dataset CDur outperforms supervised methods in terms of Seg- (64.75) and Tagging-F1 (77.13). A series of ablation experiments reveal the models’ inherent capability to correctly localize sounds with effective onset and offset estimation. In our future work, we would like to investigate the new polyphonic scene detection score (psds) [39] as a metric, which seems to provide promising insights towards a model’s duration robustness and overall performance given a specific scenario.

Acknowledgment

This work has been supported by the Major Program of National Social Science Foundation of China (No.18ZDA293). Experiments have been carried out on the PI supercomputer at Shanghai Jiao Tong University.

References

[1] F. Font, G. Roma, and X. Serra, *Sound Sharing and Retrieval*. Springer International Publishing, 2018, pp. 279–301. [Online]. Available: https://doi.org/10.1007/978-3-319-63450-0{_}10

[2] J. P. Bello, C. Mydlarz, and J. Salamon, *Sound Analysis in Smart Cities*. Cham: Springer International Publishing, 2018, pp. 373–397. [Online]. Available: https://doi.org/10.1007/978-3-319-63450-0{_}13

[3] S. Krstulović, *Audio Event Recognition in the Smart Home*. Cham: Springer International Publishing, 2018, pp. 335–371. [Online]. Available: https://doi.org/10.1007/978-3-319-63450-0{_}12

[4] H. Dinkel, Y. Chen, M. Wu, and K. Yu, “Voice activity detection in the wild via weakly supervised sound event detection,” mar 2020. [Online]. Available: http://arxiv.org/abs/2003.12222

[5] H. Dinkel and K. Yu, "Duration Robust Weakly Supervised Sound Event Detection," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, may 2020, pp. 311–315. [Online]. Available: https://ieeexplore.ieee.org/document/9053459/

[6] L. Lin, X. Wang, H. Liu, and Y. Qian, "Specialized Decision Surface and Disentangled Feature for Weakly-Supervised Polyphonic Sound Event Detection," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 28, pp. 1466–1478, may 2020. [Online]. Available: http://arxiv.org/abs/1905.10091

[7] B. McFee, J. Salamon, and J. P. Bello, "Adaptive Pooling Operators for Weakly Labeled Sound Event Detection," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 26, no. 11, pp. 2180–2193, nov 2018.

[8] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "PANNs: Large-Scale Pretrained Audio Neural Networks for Audio Pattern Recognition," dec 2019. [Online]. Available: http://arxiv.org/abs/1912.10211

[9] Q. Kong, C. Yu, Y. Xu, T. Iqbal, W. Wang, and M. D. Plumbley, "Weakly labelled audioset tagging with attention neural networks," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 27, no. 11, pp. 1791–1802, mar 2019. [Online]. Available: http://arxiv.org/abs/1903.00765

[10] S. Kothinti, K. Imoto, D. Chakrabarty, G. Sell, S. Watanabe, and M. Elhilali, "Joint Acoustic and Class Inference for Weakly Supervised Sound Event Detection," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2019-May, pp. 36–40, nov 2019. [Online]. Available: https://arxiv.org/abs/1811.04048

[11] Y. Wang, J. Li, and F. Metze, "A Comparison of Five Multiple Instance Learning Pooling Functions for Sound Event Detection with Weak Labeling," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2019-May, pp. 31–35, oct 2019. [Online]. Available: http://arxiv.org/abs/1810.09050

[12] C.-C. Kao, M. Sun, W. Wang, and C. Wang, "A Comparison of Pooling Methods on LSTM Models for Rare Acoustic Event Classification," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 316–320, feb 2020. [Online]. Available: http://arxiv.org/abs/2002.06279

[13] Y. Huang, X. Wang, L. Lin, H. Liu, and Y. Qian, "Multi-Branch Learning for Weakly-Labeled Sound Event Detection," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 641–645, feb 2020. [Online]. Available: http://arxiv.org/abs/2002.09661

[14] Q. Kong, Y. Xu, I. Sobieraj, W. Wang, and M. D. Plumbley, "Sound Event Detection and Time-Frequency Segmentation from Weakly Labelled Data," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 27, no. 4, pp. 777–787, apr 2019. [Online]. Available: http://arxiv.org/abs/1804.04715

[15] T. Pellegrini and L. Cances, "Cosine-similarity penalty to discriminate sound classes in weakly-supervised sound event detection," *Proceedings of the International Joint Conference on Neural Networks*, vol. 2019-July, jan 2019. [Online]. Available: http://arxiv.org/abs/1901.03146

[16] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks." in *ICML*, vol. 2, no. 3, 2016, p. 7.

[17] L. JiaKai, "Mean teacher convolution system for dcase 2018 task 4," DCASE2018 Challenge, Tech. Rep., September 2018.

[18] D. Lee, S. Lee, Y. Han, and K. Lee, "Ensemble of convolutional neural networks for weakly-supervised sound event detection using multiple scale input," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*, 2017, pp. 74–79.

[19] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artificial Intelligence*, vol. 89, no. 1-2, pp. 31–71, jan 1997.

[20] F. Zhai, S. Potdar, B. Xiang, and B. Zhou, "Neural models for sequence chunking," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, ser. AAAI'17. AAAI Press, 2017, p. 3365–3371.

[21] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019, vol. 32, pp. 8026–8037. [Online]. Available: http://arxiv.org/abs/1912.01703

[22] B. McFee, V. Lostanlen, M. McVicar, A. Metsai, S. Balke, C. Thomé, C. Raffel, A. Malek, D. Lee, F. Zalkow, K. Lee, O. Nieto, J. Mason, D. Ellis, R. Yamamoto, S. Seyfarth, E. Battenberg, V. Morozov, R. Bittner, K. Choi, J. Moore, Z. Wei, S. Hidaka, Nullmightybofo, P. Friesch, F.-R. Stöter, D. Hereñú, T. Kim, M. Vollrath, and A. Weiss, "librosa/librosa: 0.7.2," jan 2020. [Online]. Available: https://zenodo.org/record/3606573

[23] O. Tange, "Gnu parallel - the command-line power tool," *;login: The USENIX Magazine*, vol. 36, no. 1, pp. 42–47, Feb. 2011. [Online]. Available: http://www.gnu.org/s/parallel

[24] E. Cakir and T. Virtanen, "End-to-End Polyphonic Sound Event Detection Using Convolutional Recurrent Neural Networks with Learned Time-Frequency Representation Input," *Proceedings of the International Joint Conference on Neural Networks*, vol. 2018-July, 2018. [Online]. Available: https://arxiv.org/pdf/1805.03647.pdf

[25] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *7th International Conference on Learning Representations, ICLR 2019*, p. 13, Feb 2019. [Online]. Available: https://arxiv.org/pdf/1711.05101.pdf

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[27] K. Sechidis, G. Tsoumakas, and I. Vlahavas, "On the stratification of multi-label data," *Machine Learning and Knowledge Discovery in Databases*, pp. 145–158, 2011.

[28] 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*, vol. 2017-October. Institute of Electrical and Electronics Engineers Inc., dec 2017, pp. 344–348.

[29] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proceedings of the 22nd ACM International Conference on Multimedia*, ser. MM '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 1041–1044. [Online]. Available: https://doi.org/10.1145/2647868.2655045

[30] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio Set: An ontology and human-labeled dataset for audio events," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. Institute of Electrical and Electronics Engineers Inc., jun 2017, pp. 776–780.

[31] D. S. Park, W. Chan, Y. Zhang, C. C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. 2019-September, pp. 2613–2617, apr 2019. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2680

[32] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[33] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, p. 162, 2016. [Online]. Available: http://www.mdpi.com/2076-3417/6/6/162

[34] I. Martin-Morato, A. Mesaros, T. Heittola, T. Virtanen, M. Cobos, and F. J. Ferri, "Sound Event Envelope Estimation in Polyphonic Mixtures," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2019-May. Institute of Electrical and Electronics Engineers Inc., may 2019, pp. 935–939.

[35] S. Adavanne and T. Virtanen, "Sound event detection using weakly labeled dataset with stacked convolutional and recurrent neural network," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*. Tampere University of Technology. Laboratory of Signal Processing, 2017, pp. 12–16.

[36] Y. Xu, Q. Kong, W. Wang, and M. D. Plumbley, "Surrey-CVSSP system for DCASE2017 challenge task4," DCASE2017 Challenge, Tech. Rep., September 2017.

[37] T. Iqbal, Y. Xu, Q. Kong, and W. Wang, "Capsule routing for sound event detection," *European Signal Processing Conference*, vol. 2018-Septe, pp. 2255–2259, 2018. [Online]. Available: https://arxiv.org/pdf/1806.04699.pdfhttp://arxiv.org/abs/1806.04699

[38] Y. L. Liu, J. Yan, Y. Song, and J. Du, "Ustc-nelslip system for dcase 2018 challenge task 4," DCASE2018 Challenge, Tech. Rep., September 2018.

[39] C. Bilen, G. Ferroni, F. Tuveri, J. Azcarreta, and S. Krstulovic, "A Framework for the Robust Evaluation of Sound Event Detection," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics,*

Speech and Signal Processing (ICASSP). IEEE, may 2019, pp. 61–65. [Online]. Available: http://arxiv.org/abs/1910.08440