Learning to Recognize Transient Sound Events Using Attentional Supervision

Szu-Yu Chou[1,2], **Jyh-Shing Roger Jang** [1], **Yi-Hsuan Yang**[2]
[1]Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei, Taiwan
[2]Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan
fearofchou@citi.sinica.edu.tw, jang@csie.ntu.edu.tw, yang@citi.sinica.edu.tw

Abstract

Making sense of the surrounding context and ongoing events through not only the visual inputs but also acoustic cues is critical for various AI applications. This paper presents an attempt to learn a neural network model that recognizes more than 500 different sound events from the audio part of user generated videos (UGV). Aside from the large number of categories and the diverse recording conditions found in UGV, the task is challenging because a sound event may occur only for a short period of time in a video clip. Our model specifically tackles this issue by combining a main subnet that aggregates information from the entire clip to make clip-level predictions, and a supplementary subnet that examines each short segment of the clip for segment-level predictions. As the labeled data available for model training are typically on the clip level, the latter subnet learns to pay attention to segments selectively to facilitate attentional segment-level supervision. We call our model the M&mnet, for it leverages both "M"acro (clip-level) supervision and "m"icro (segment-level) supervision derived from the macro one. Our experiments show that M&mnet works remarkably well for recognizing sound events, establishing a new state-of-the-art for DCASE17 and AudioSet data sets. Qualitative analysis suggests that our model exhibits strong gains for short events. In addition, we show that the micro subnet is computationally light and we can use multiple micro subnets to better exploit information in different temporal scales.

1 Introduction

Sound event recognition is important for many applications, including surveillance, smart cars, video indexing, etc [Mesaros *et al.*, 2016; Mesaros *et al.*, 2017]. While exciting progress has been made in the past few years, existing works on sound event recognition were usually based on data sets of moderate size and limited vocabulary, due to the difficulty in collecting labeled data. Such data sets include UrbanSound-8K [Salamon *et al.*, 2014], ESC-2K [Piczak, 2015], and DCASE16-1K [Mesaros *et al.*, 2016]. To deal with this issue, Google shared with the research community the AudioSet [Gemmeke *et al.*, 2017], a collection of over two million 10-second YouTube video clips with annotations (on the clip level) of over 500 sound categories, covering sounds of nature, animals, human and sounds of things. This data set facilitates learning from large-scale data recorded in unconstrained conditions (user generated videos) to discriminate a diverse set of sound events encountered in daily lives. A subset of it was used in DCASE2017 Task 4 [Mesaros *et al.*, 2017], with envisioned applications on smart cars.

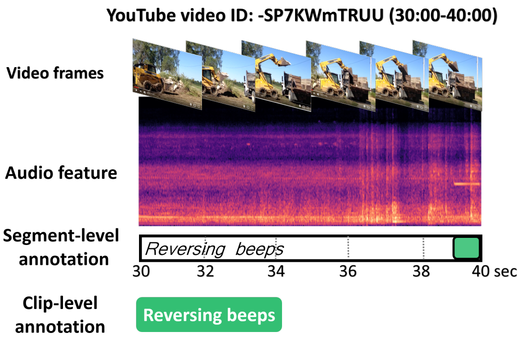

Figure 1: Illustration of a transient (less than 1 second) sound event. The clip-level annotation is available for all the two million video clips in the AudioSet, but the segment-level annotation marking the time interval the event actually took place is available only for a subset of 488 clips used by the DCASE2017 Challenge.

Naturally, such diverse sound events have fairly different characteristics. In a video clip, some events may occur persistently across the entire clip (e.g., background and environmental sounds), while some others may appear only for a short period of time, as exemplified in Figure 1 (also see Figures 4 and 5). While one can increase the depth of a network to increase model capacity and to learn fine-grained information for sound event recognition (e.g. Google demonstrated learning a 50-layer ResNet model from an in-house superset of AudioSet [Hershey *et al.*, 2017]), a performance bottleneck for higher accuracy may be related to the lack of *segment-level* annotation that marks the time interval(s) an event took place in a clip. With clip-level labels alone, standard network architectures such as ConvNet and ResNet [He *et al.*, 2016] have to aggregate information across an entire clip to make predictions, so transient events (e.g., less than 1 second) may be easily overlooked for a 10-second clip.[1]

Research has been made to address the aforementioned issue, attempting to localize sound events on the segment level by using weakly-supervised methods [Liu and Yang, 2016; Hershey *et al.*, 2017; Vu *et al.*, 2017; Lee *et al.*, 2017a]. In general, most of them assumed all the segments of a clip share the same labels as the (mother) clip, an overly strong assumption for transient events. More recent work used the so-called attention mechanism [Bahdanau *et al.*, 2014] to measure the importance of different segments while making the clip-level prediction [Huang *et al.*, 2017; Xu *et al.*, 2017a]. This method can be further extended by using both label-specific and label-agnostic attentions [Girdhar and Ramanan, 2017]. Yet, [Zhu *et al.*, 2017] showed in a computer vision problem that such a method still tends to neglect tiny objects (analogously transient sound events) when there are relatively larger (longer) ones in the same image (audio clip).

We propose in this paper a novel network architecture called *M&mnet* for large-scale sound event recognition, especially for recalling transient events. Instead of using attention for clip-level prediction only, we use attention to create *attentional segment-level supervision* and use that to learn to make segment-level predictions. The attentional supervision is derived from the clip-level labels, but we use attention to weigh the segments when calculating segment-level prediction errors. The name of M&mnet stems from the fact that it considers both "M"acro (clip-level) and "m"icro (segment-level) supervision to train a neural network in an end-to-end manner. As illustrated in Figure 2, a main subnet such as ResNet is trained in regular way based on available clip-level labels, to provide a holistic view of a clip. At the same time, a supplementary subnet (which is much thinner than the main subnet) is trained additionally to "listen" more carefully each segment of the clip. For a testing clip, we fuse the clip- and segment-level predictions to arrive at the final estimate.

The micro subnet taking care of attentional micro supervision is computationally light and can be used as an add-on to any types of main subnet. For ConvNet- or ResNet-based main net, we can learn a micro subnet from the feature map of each intermediate convolutional layers, leading to multi-scale (MS) micro supervision at a hierarchy of multiple temporal scales. We call this extension *M&mnet-MS*.

The proposed M&mnet is idea-wise a simple modification of prevalent architectures such as ResNet. Yet, extensive experiments on DCASE2017 and AudioSet data sets show that M&mnet works remarkably well for sound event recognition, outperforming ResNet and many existing methods (to the best of our knowledge) that have been evaluated on the two data sets. Moreover, qualitative analysis shows that M&mnet can indeed better recall transient events than other competing attention-based convolutional or recurrent networks (e.g., [Xu *et al.*, 2017a]).

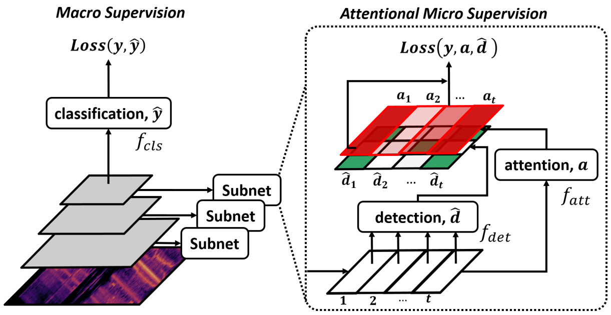

Figure 2: Schematic plot of the proposed model. The stack of layers on the left represents the main subnet for clip-level (macro) prediction. From the output of one or multiple intermediate layers of the main subnet, we train supplementary subnet(s) for segment-level (micro) prediction, based on the proposed attentional supervision.

2 Approach

2.1 Problem Formulation

We are given a training set $\mathcal{D} = \{(\chi, \mathbf{y}) \mid \chi \in \mathcal{X}, y \in \{0,1\}^C\}$, where $\mathcal{X}$ denotes a set of $N = |\mathcal{X}|$ audio clips, $\mathbf{y}$ a binary vector of clip-level labels, and C the number of possible sound events (categories). An entry y_c in $\mathbf{y}$ is 1 if the corresponding event appears in the clip, and 0 otherwise. A clip χ may be of arbitrary length, and the training set does not specify the time interval(s) an event occurred in the clip. Our goal is to learn a model from $\mathcal{D}$ (with only clip-level annotations) for predicting whether any of these C events occurred in an input (unseen) clip (i.e., making clip-level predictions). This is in principle a multi-label classification problem. The model f is parameterized by θ.

2.2 Macro (Clip-level) Supervision

M&mnet uses a main (macro) subnet for clip-level predictions, based on clip-level supervision available in $\mathcal{D}$. In this paper, we consider either standard ConvNet or ResNet as the network architecture for the macro net, for they have been shown effective for sound event recognition and detection in recent years [Eghbal-Zadeh *et al.*, 2016; Kumar *et al.*, 2018; Hershey *et al.*, 2017]. ConvNet and ResNet differ mainly in the use of the so-called residual blocks [He *et al.*, 2016]; both of them can be viewed as the cascade of a feature learning stack of convolutional layers f_{cnn} and a classification stack of fully-connected layers f_{cls}. Given the input feature of a clip χ, the *clip-level prediction* $\hat{\mathbf{y}} \in \mathbb{R}^C$ is made by

$$\mathbf{X} = f_{cnn}(g(\chi); \theta_{cnn}) \,, \tag{1}$$

$$\hat{\mathbf{y}} = \sigma(f_{cls}(\text{pool}(\mathbf{X}); \theta_{cls})) \,, \tag{2}$$

where $g(\cdot)$ denotes an optional function that extracts a feature representation (e.g., mel-spectrogram) from an input audio clip; $\mathbf{X} \in \mathbb{R}^{K \times T}$ denotes the output (a.k.a. a *feature map*) from the last (convolutional) layer of f_{cnn}, with K channels and T temporal segments; pool$(\cdot)$ is the pooling operation that aggregates the feature map $\mathbf{X}$ across the temporal dimension (e.g., by taking the mean) to get a fixed-length K-dimensional vector, allowing the model to deal with variable-length audio clips; and finally $\sigma(x) = 1/(1 + e^{-x})$ is the

[1] DCASE2017 Task 4 [Mesaros *et al.*, 2017] compiled a small set of 488 clips with segment-level annotations for 17 sound events, but used it only for evaluation. Similarly, we use such segment-level annotations for evaluation (e.g., in Figures 4 and 5), not for training.

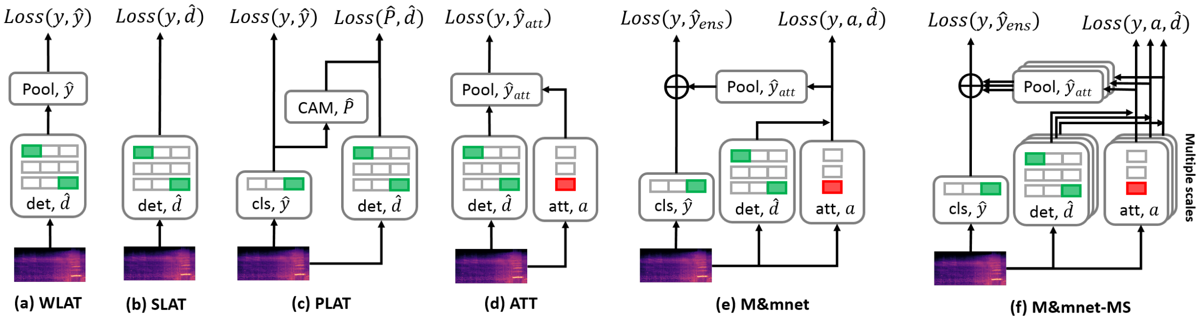

Figure 3: Six different methods for model training: (a) weak-label assumption training (WLAT) [Kumar *et al.*, 2018]; (b) strong-label assumption training (SLAT) [Hershey *et al.*, 2017]; (c) predicted-label assumption training (PLAT) [Diba *et al.*, 2016]; (d) attentional prediction model (ATT) [Xu *et al.*, 2017a]; (e) the proposed M&mnet and its (f) multi-scale extension. See Section 2.6 for explanations.

sigmoid function to scale things to $[0,1]$.[2]

The macro subnet is trained by minimizing the binary cross-entropy loss $\xi(\cdot,\cdot)$ between $\mathbf{y}$ and $\hat{\mathbf{y}}$, for each χ in $\mathcal{D}$:

$$\mathcal{L}_{\text{macro}} \equiv \xi(\mathbf{y},\hat{\mathbf{y}}) = -\sum_{c=1}^{C} \hat{y}_c \log y_c + (1-\hat{y}_c)\log(1-y_c)\,. \tag{3}$$

2.3 Attentional Micro (Segment-level) Supervision

M&mnet additionally uses a micro subnet to make segment-level predictions. Instead of using a pooling function to aggregate the feature map $\mathbf{X}$ as done in Eq. (2), we treat each temporal slice of $\mathbf{X}$ as a segment $\mathbf{x}_t \in \mathbb{R}^K$, $t = 1,\ldots,T$, and use such segments as input to a stack of fully connected layers f_{det} for making *segment-level predictions* $\hat{\mathbf{d}}_t \in \mathbb{R}^C$,

$$\hat{\mathbf{d}}_t = \sigma(f_{det}(\mathbf{x}_t;\theta_{det}))\,. \tag{4}$$

Due to the absence of segment-level labels, to measure the accuracy of segment-level prediction we can only compute the binary cross-entropy loss between $\mathbf{y}$ and $\hat{\mathbf{d}}_t$, for $t = 1,\ldots,T$. However, oftentimes not all the segments of a clip have the same labels as the mother clip. Therefore, we introduce an attention mechanism to weigh the segments while calculating the errors. Instead of computing a single attention score of each segment as done in [Huang *et al.*, 2017; Xu *et al.*, 2017a], we find it beneficial to learn multiple attention scores (as originally proposed by [Shen *et al.*, 2018] for language understanding) to leverage attentions from possibly different aspects. Specifically, for each segment, we compute an m-dimensional attentional vector $\tilde{\mathbf{v}}_t \in \mathbb{R}^m$ by feeding $\mathbf{x}_t$ to another stack of fully connected layers f_{att}:

$$\mathbf{v}_t = f_{att}(\mathbf{x}_t;\theta_{att})\,. \tag{5}$$

We normalize $\{\mathbf{v}_t\}_{t=1}^{T}$ by using the softmax function over the T segments for each m and then compute the *segment-level attention score* a_t by summing over the M dimensions:

$$a_t = \sum_{m=1}^{M} \frac{\exp(v_{t,m})}{\sum_{t=1}^{T}\exp(v_{t,m})}\,. \tag{6}$$

We use a_t to weigh the error for each segment while calculating the micro supervision loss:

$$\mathcal{L}_{\text{micro}} \equiv \sum_{t=1}^{T} \mathcal{L}(\mathbf{y},a_t,\hat{\mathbf{d}}_t) = \sum_{t=1}^{T} a_t\, \xi(\mathbf{y},\hat{\mathbf{d}}_t)\,. \tag{7}$$

M&mnet combines $\mathcal{L}_{\text{macro}}$ and $\mathcal{L}_{\text{micro}}$ and optimizes the set of parameters $\theta = \{\theta_{cnn},\theta_{cls},\theta_{det},\theta_{att}\}$ via the following objective function in an end-to-end manner,

$$\min_{\theta} \sum_{i=1}^{N} \left(\xi(\mathbf{y}^i,\hat{\mathbf{y}}^i) + \lambda \sum_{t=1}^{T} a_t^i\, \xi(\mathbf{y}^i,\hat{\mathbf{d}}_t^i)\right) + \mathcal{R}(\theta)\,, \tag{8}$$

where we use superscripts to index audio clips in the training set, and $\mathcal{R}(\theta)$ is a regularization term to avoid overfitting. λ is a hyper-parameter that weighs macro and micro supervisions; we simply set it to 1 in our implementation. We note that Eq. (8) can also be understood as performing multi-task learning [Kendall *et al.*, 2017].

2.4 Ensemble Prediction

As depicted in Figure 3(e), M&mnet further uses the attention scores a_t to fuse the result of clip- and segment-level predictions in testing time. We call this clip-level ensemble prediction, or EP for short. Specifically, the ensemble prediction $\hat{\mathbf{y}}_{ens} \in \mathbb{R}^C$ is calculated by:

$$\hat{\mathbf{y}}_{ens} = \mu\hat{\mathbf{y}} + (1-\mu)\hat{\mathbf{y}}_{att}\,, \tag{9}$$

where $\hat{\mathbf{y}}_{att} \equiv \sum_{t=1}^{T}\alpha_t\hat{\mathbf{d}}_t$; α_t is the softmax-ed version of a_t, with $\sum_{t=1}^{T}\alpha_t = 1$; and $\mu \in [0,1]$ is a hyper-parameter that is simply set to 0.5 in our implementation.

2.5 Multi-scale Attentional Micro Supervision

Instead of using micro supervision for only the last convolutional layer of f_{cnn}, we can use micro supervision for all the intermediate convolutional layers of f_{cnn} to have a feature pyramid module [Lin *et al.*, 2016] that captures information in different temporal scales. Comparing to using multi-scale audio features as input to the neural network [Dieleman and Schrauwen, 2013; Liu and Yang, 2016], the proposed design is computationally more efficient because 1) it largely reuses

[2] We note that in computer vision, a feature map is usually a tensor. However, for audio problems we use 1D convolutions along time and the resulting feature map is a matrix.

Method	Attention	Depth	Param.	mAP	AUC	F1	R	P
ConvNet (mean pooling)		8	1.54M	54.1	88.1	56.3	57.8	58.9
ResNet (max pooling)		12	3.70M	56.5	90.6	58.0	61.7	57.5
ResNet (mean pooling)		12	3.70M	57.1	90.7	58.5	58.6	61.9
ResNet-WLAT [Kumar *et al.*, 2018]		12	3.70M	56.5	90.5	57.3	64.1	54.0
ResNet-SLAT [Hershey *et al.*, 2017]		12	3.70M	58.9	91.5	58.4	58.3	62.6
ResNet-PLAT [Diba *et al.*, 2016]		12	3.72M	53.1	88.0	57.1	62.4	55.0
ResNet-ATT [Xu *et al.*, 2017a]	✓	12+1	4.14M	59.0	90.5	59.3	67.0	56.0
ResNet-SPDA [Zhang *et al.*, 2016]	✓	12+1	6.84M	59.6	91.1	60.0	64.0	59.3
M&mnet	✓	12+2	4.16M	**63.2**	91.9	**63.4**	66.1	**64.4**
M&mnet without EP	✓	12+2	4.16M	61.4	**92.1**	62.3	**68.7**	58.6

Table 1: Sound event recognition performance (in %) on DCASE17. All the methods reported here are based on our implementation. We also indicate whether a method uses any attention mechanism, the network depth, and the number of parameters to be learned.

a single f_{cnn}, and 2) each micro subnet is composed of only fully-connected layers (i.e., f_{det} and f_{att}).

The feature learning stack f_{cnn} is a ConvNet or ResNet that iteratively uses convolutional layers for pattern extraction and pooling layers for abstraction/downsampling. Assuming L convolutional layers, we denote the set of group (hierarchical) feature maps from these layers as $\{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \cdots, \mathbf{X}^{(L)}\}$. Each of them $\mathbf{X}^{(l)}$ has a different number of channels $K^{(l)}$ and temporal segments $T^{(l)}$. While we use only the last one $\mathbf{X}^{(L)} \equiv \mathbf{X}$ in Section 2.3, we want to use them all here. Moreover, following the so-called feature pyramid networks structure widely used in image segmentation [Lin *et al.*, 2016], we can learn $L-1$ transposed convolutional layers to upsample the last feature map $\mathbf{X}^{(L)}$, adding extra $L-1$ feature maps. Then, M&mnet-MS can be trained by replacing $\mathcal{L}_{\text{micro}}$ with

$$\mathcal{L}_{\text{micro-MS}} \equiv \sum_{l=1}^{2L-1} \mathcal{L}_{\text{micro@}l} = \sum_{l=1}^{2L-1} \sum_{t=1}^{T^{(l)}} a_t^{(l)} \, \xi(\mathbf{y}, \hat{\mathbf{d}}_t^{(l)}) \,, \quad (10)$$

where $\hat{\mathbf{d}}_t^{(l)} = \sigma(f_{det}(\mathbf{x}_t^{(l)}; \theta_{det@l}))$ is the segment-level prediction from $\mathbf{X}^{(l)}$, and similarly for $a_t^{(l)}$. The set of parameters becomes $\theta = \{\theta_{cnn}, \theta_{cls}, \{\theta_{det@l}\}_{l=1}^{2L-1}, \{\theta_{att@l}\}_{l=1}^{2L-1}\}$.

2.6 Related Methods

Figure 3 compares M&mnet with four existing methods. Weak-label assumption training (WLAT) [Kumar *et al.*, 2018] is a weakly-supervised method, making a prediction for each segment and then aggregating the predictions by mean pooling to make clip-level prediction. Strong-label assumption training (SLAT) [Hershey *et al.*, 2017] assumes all the segments share the same label as the mother clip and uses only such (non-attentional) micro supervision for training. Predicted-label assumption training (PLAT) [Diba *et al.*, 2016] creates a class activation map (CAM) based on macro-level predictions to extrapolate possible micro-level labels, and then uses the extrapolated labels for micro supervision. This method considers both macro and micro supervision, but the extrapolated micro supervision may be error-prone. Finally, ATT [Xu *et al.*, 2017a] can be viewed as an advanced version of WLAT that further uses an attention mechanism to weigh the segments while making clip-level prediction. A similar approach was proposed by [Huang *et al.*, 2017].

2.7 Implementation Details

Data preprocessing: For $g(\cdot)$ (see Eq. (1)), we used the librosa library [McFee *et al.*, 2015] to represent an audio clip by a 128-bin log-scale mel-spectrogram from 0 to 22 kHz for this representation has been widely used in audio tasks [van den Oord *et al.*, 2013; Liu and Yang, 2016]. The mel-spectrograms were computed by short-time Fourier transform with a 1/4-overlapping sliding window of size 46 ms, assuming 44.1 kHz sampling rate. Moreover, we standardized the extracted features by removing mean and dividing by standard deviation by values derived from the training set.

Per-class loss weight: Class imbalance is commonly seen in large-scale data sets. To address this, we followed the trick used by Google[3] and assigned predefined weights to each class while calculating the training loss. The predefined weight w_c for class c can be described as

$$w_c = \left(\frac{\overline{p}}{p_c} \cdot \frac{1-p_c}{1-\overline{p}} \right)^{\beta} \,, \quad (11)$$

where p_c is the proportion of examples for class c, $\overline{p}$ is the mean of all p_c, and β another hyper-parameter that was empirically set to 0.3.

Network design: Following [He *et al.*, 2016], we used batch normalization [Ioffe and Szegedy, 2015] for each convolutional layer and used RELU as the activation function. All network parameters were initialized following [He *et al.*, 2016] and trained without any additional data or pre-trained models. For optimization, we used SGD with a mini-batch size of 64 and initial learning rate 0.1. We divided the learning rate by 10 every 30 epochs and set the maximal number of epochs to 100. To avoid overfitting, we set the weight decay to 1e-4. For reproducibility, we will share the python source code and trained models online through a github repo.[4]

3 Experiments

The first set of experiments uses **DCASE17**, a subset of AudioSet that was used in DCASE2017 Challenge Task 4 [Mesaros *et al.*, 2017]. It contains around 50K audio clips

[3] http://www.cs.tut.fi/sgn/arg/dcase2017/documents/workshop_presentations/the_story_of_audioset.pdf

[4] https://github.com/fearofchou/mmnet

Model	SM	F1	R	P
Baseline [Mesaros *et al.*, 2017]	✓	13.1	12.2	14.1
SDCNN [Lee *et al.*, 2017b]		41.2	37.6	45.7
DenseCRNN [Vu *et al.*, 2017]		51.8	49.5	54.2
FrameCNN [Chou *et al.*, 2017]		53.8	55.4	54.0
CNN-EB [Lee *et al.*, 2017a]		57.0	47.9	**70.3**
CVSSP [Xu *et al.*, 2017b]		61.9	64.7	59.4
M&mnet	✓	63.4	66.1	64.4
M&mnet-MS	✓	**65.6**	**67.5**	65.2

Table 2: Sound event recognition performance (in %) on DCASE17 data set, comparing the two proposed methods (bottom two) against those actually participated in the challenge. CVSSP used to be the winning method. Most of the methods use a hybrid of multiple models, but the proposed ones use only a single model (SM).

Model	Att.	mAP	AUC
WLAT [Kumar *et al.*, 2018]		21.3	92.7
ConvNet (mean pooling)		20.3	93.5
ResNet (mean pooling)		21.8	93.6
ResNet-ATT [Xu *et al.*, 2017a]	✓	22.0	93.5
ResNet-SPDA [Zhang *et al.*, 2016]	✓	21.9	93.6
M&mnet	✓	22.6	93.8
M&mnet-MS	✓	**23.2**	**94.0**

Table 3: Sound event recognition performance (in %) on AudioSet-20K, using AudioSet-22K for training. The result of WLAT is cited from [Kumar *et al.*, 2018]; the others are our own implementation.

labeled with 17 sound events, all of which are related to the sound of `Warning` or `Vehicle`. The pre-defined training set has at least 30 examples (i.e., clips) per class, and the test set has around 30 examples per class. The second set of experiments uses AudioSet, containing over 2M audio clips with 527 possible sound events. Google provides a balanced training set with at least 59 examples per class, called **AudioSet-22K**, and a balanced test set with again at least 59 examples per class, called **AudioSet-20K**. As 22K clips only represent a small sample of the data, we took a much larger, yet imbalanced 1M-clip subset by removing clips that were only annotated with either `Music` or `Speech`, the two most popular classes. There is no overlap between the resulting **AudioSet-1M** and the test set AudioSet-20K.

Following common practice, we use class-level mean average precision (mAP), area under ROC curve (AUC), recall (R), precision (P) and F1-score (F1; the harmonic mean of R and P) as the evaluation metrics. And, when a binarized version of $\hat{\mathbf{y}}$ is needed in calculating R and P, we use a class-specific confidence threshold τ_c to quantize $\hat{\mathbf{y}}$, based on the same approach as [Lee *et al.*, 2017a; Xu *et al.*, 2017b].

Table 1 compares the performance of M&mnet against our own implementation of a number of existing methods on DCASE17: the first three are standard methods considering only macro supervision; the middle four have been introduced in Section 2.6 and we used ResNet as their main subnet; the remaining one ResNet-SPDA [Zhang *et al.*, 2016] is a more sophisticated (and computationally more expensive) attention-based method. We can see that, with similar network depth,[5] M&mnet outperforms the competing methods in most metrics, exhibiting around +4% improvement in mAP and F1 over ResNet-ATT and ResNet-PDN. Moreover, while ResNet involves 3.70M parameters, the additional micro subnet used by M&mnet only adds 0.46M parameters (+12% relatively), which is relatively not a big overhead. The last two rows of Table 1 show that EP (see Section 2.4) also contributes to the effectiveness of M&mnet in F1 and mAP.

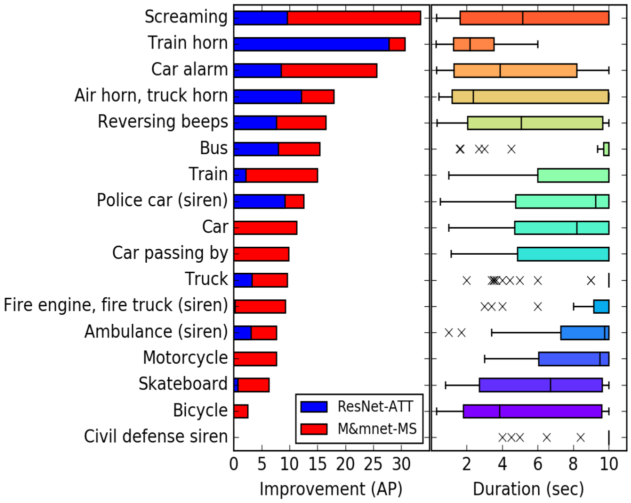

Figure 4: (Left) relative improvement in per-class average precision (AP) of M&mnet-MS (red bar) and ResNet-ATT (blue bar) over ResNet (using mean pooling) on the DCASE17 test set, and (right) box plot showing the time duration of the events per class.

Model	Att.	Depth	Param.	mAP	AUC
ResNet (mean)		28	16M	27.8	94.0
ResNet (mean)		50	20M	26.6	93.8
ResNet-ATT	✓	12+1	4M	29.8	94.0
M&mnet-MS	✓	12+10	8M	**32.7**	**95.1**

Table 4: Sound event recognition performance (in %) on AudioSet-20K, using AudioSet-1M for training. All our implementation.

Table 2 further compares M&mnet against the methods that actually took part in the DCASE2017 Challenge, citing the results reported on the DCASE website rather than implementing them by ourselves. Most of these methods used a hybrid of multiple models. As a single model alone, M&mnet already outperforms these prior arts. In particular, M&mnet-MS outperforms the winning method CVSSP [Xu *et al.*, 2017b] by +3.7% in F1, a large performance leap.

Figure 4 presents a detailed per-class performance comparison between M&mnet-MS and ResNet-ATT, showing their (positive-only) relative performance gain over ResNet. We see that M&mnet-MS outperforms ResNet-ATT consistently for all the 17 sound classes, especially for shorter ones such as `Screaming` and `Car alarm`. Salient improvement is

[5] In the 'Depth' column of Table 1 we write '+1' for ResNet-ATT and ResNet-SPDA due to the additional attention layer, and '+2' for M&mnet for the added attention and detection layers (cf. Figure 3).

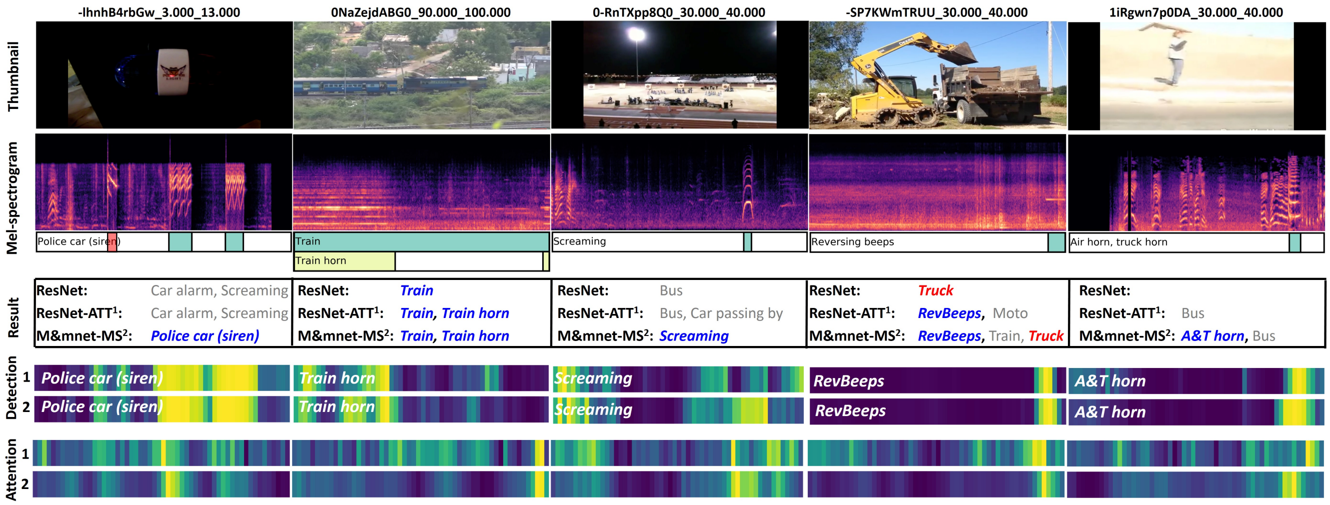
-lhnhB4rbGw_3.000_13.000
0NaZejdABG0_90.000_100.000
0-RnTXpp8Q0_30.000_40.000
-SP7KWmTRUU_30.000_40.000
1iRgwn7p0DA_30.000_40.000
Thumbnail
Mel-spectrogram
Police car (siren)
Train
Train horn
Screaming
Reversing beeps
Air horn, truck horn
Result
ResNet: Car alarm, Screaming
ResNet-ATT¹: Car alarm, Screaming
M&mnet-MS²: Police car (siren)
ResNet: Train
ResNet-ATT¹: Train, Train horn
M&mnet-MS²: Train, Train horn
ResNet: Bus
ResNet-ATT¹: Bus, Car passing by
M&mnet-MS²: Screaming
ResNet: Truck
ResNet-ATT¹: RevBeeps, Moto
M&mnet-MS²: RevBeeps, Train, Truck
ResNet:
ResNet-ATT¹: Bus
M&mnet-MS²: A&T horn, Bus
Detection
1
2
Police car (siren)
Train horn
Screaming
RevBeeps
A&T horn
Attention
1
2

Figure 5: Qualitative result of sound event recognition on DCASE17 test set. From top to bottom: 1) YouTube video ID, 2) video thumbnail, 3) mel-spectrogram and the corresponding segment-level labels, 4) the clip-level predictions (binarized) of three methods: ResNet (with mean pooling), ResNet-ATT and M&mnet-MS, 5) the segment-level predictions $\hat{d}_{t,c}$ of ResNet-ATT and M&mnet-MS for a groundtruth class, and 6) the estimated segment-level attention scores a_t of ResNet-ATT and M&mnet-MS. Those in red color are labels we think should have been added to the groundtruth set. Those in blue and gray are true positives and false positives of clip-level result. Best viewed in color.

also found in classes that are occasionally short, such as `Car` and `Car passing by` (ResNet-ATT does not have performance gain over ResNet for these two classes). The only class M&mnet-MS does not perform well is `Civil defense siren`, exhibiting –0.5% relative AP for this long event.

Tables 3 and 4 show M&mnet and M&mnet-MS outperform existing methods in recognizing 500+ sound events as well on AudioSet-20K. Using a larger data set (AudioSet-1M) for training naturally leads to better results. We also see from Table 4 that we do not necessarily need very deep networks when we use attention mechanisms.

Figure 5 offers a qualitative comparison of the result of ResNet, ResNet-ATT, and M&mnet-MS for five clips from DCASE17. From the middle row we see M&mnet-MS can recall the events in all the clips, whereas ResNet only recalls the longer event `Train` of the second clip but misses the shorter event `Train horn`. For the fourth clip, both ResNet and M&mnet-MS recall the longer event `Truck`, but ResNet-ATT only recalls the shorter event `Reversing beeps`. These examples suggest that M&mnet-MS works well for both long and short events, possibly due to the joint consideration of macro and micro supervisions.

Figure 5 also shows the segment-level predictions $\hat{d}_{t,c}$ of ResNet-ATT and M&mnet-MS for some target class, and the estimated attention scores a_t. We see that both $\hat{d}_{t,c}$ and a_t correspond nicely to the segment-level annotations (which were not used for training), especially for M&mnet-MS.

Figure 6 finally investigates the effect of the temporal length of the segments (i.e., the length of the part in an audio clip each slice $\mathbf{x}_t$ of the feature map of the last convolutional layer of the main subnet actually accounts for) on the performance of M&mnet and M&mnet-MS.[6] Intuitively, overly short segments may not contain sufficient information for discriminating different events, whereas overly long segments may miss local, transient information. Figure 6 shows that the segment length matters and a proper choice seems to be 0.18 second for both methods. Hence, all the experiments presented above set the segment length to this number.

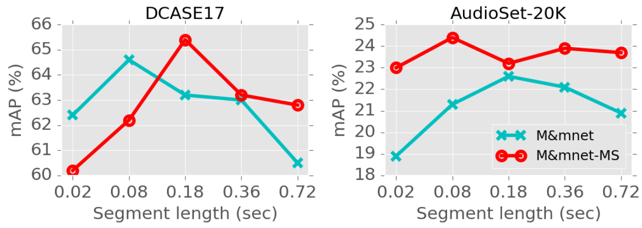

Figure 6: Effect of segment length on the performance of M&mnet and M&mnet-MS on (left) DCASE17 and (right) AudioSet-20K.

4 Conclusion

In this paper, we have demonstrated in the context of sound event recognition a novel use of attention mechanisms to create attentional segment-level supervision without actual segment-level annotations. Moreover, we presented a novel network architecture that considers both clip-level (macro) and segment-level (micro) supervisions to train a neural network in an end-to-end manner. Extensive experiments showed that our method can recognize either long or short sound events better than existing methods do. A multi-scale version of our method led to 65.6% F1 score for DCASE17 and 32.7% mAP for AudioSet-20K. An interesting future direction is to apply the idea of attentional supervision to other fields, such as image recognition.

[6] This number is related to the filter and stride sizes used in f_{cnn}.

References

[Bahdanau *et al.*, 2014] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv*, 2014.

[Chou *et al.*, 2017] Szu-Yu Chou, Jyh-Shing Jang, and Yi-Hsuan Yang. FrameCNN: A weakly-supervised learning framework for frame-wise acoustic event detection and classification. In *Proc. DCASE*, 2017.

[Diba *et al.*, 2016] Ali Diba, Vivek Sharma, Ali Mohammad Pazandeh, Hamed Pirsiavash, and Luc Van Gool. Weakly supervised cascaded convolutional networks. In *Proc. CVPR*, 2016.

[Dieleman and Schrauwen, 2013] Sander Dieleman and Benjamin Schrauwen. Multiscale approaches to music audio feature learning. In *Proc. ICASSP*, 2013.

[Eghbal-Zadeh *et al.*, 2016] Hamid Eghbal-Zadeh, Bernhard Lehner, Matthias Dorfer, and Gerhard Widmer. CP-JKU submissions for DCASE-2016: a hybrid approach using binaural i-vectors and deep convolutional neural networks. In *Proc. DCASE*, 2016.

[Gemmeke *et al.*, 2017] Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *Proc. ICASSP*, 2017.

[Girdhar and Ramanan, 2017] Rohit Girdhar and Deva Ramanan. Attentional pooling for action recognition. In *Proc. NIPS*, 2017.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016.

[Hershey *et al.*, 2017] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron Weiss, and Kevin Wilson. CNN architectures for large-scale audio classification. In *Proc. ICASSP*. 2017.

[Huang *et al.*, 2017] Yu-Siang Huang, Szu-Yu Chou, and Yi-Hsuan Yang. Music thumbnailing via neural attention modeling of music emotion. In *Proc. APSIPA ASC*, 2017.

[Ioffe and Szegedy, 2015] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. ICML*, 2015.

[Kendall *et al.*, 2017] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *arXiv*, 2017.

[Kumar *et al.*, 2018] Anurag Kumar, Maksim Khadkevich, and Christian Fügen. Knowledge transfer from weakly labeled audio using convolutional neural network for sound events and scenes. In *Proc. ICASSP*, 2018.

[Lee *et al.*, 2017a] Donmoon Lee, Subin Lee, Yoonchang Han, and Kyogu Lee. Ensemble of convolutional neural networks for weakly-supervised sound event detection using multiple scale input. In *Proc. DCASE*, 2017.

[Lee *et al.*, 2017b] Jongpil Lee, Jiyoung Park, and Juhan Nam. Combining multi-scale features using sample-level deep convolutional neural networks for weakly supervised sound event detection. In *Proc. DCASE*, 2017.

[Lin *et al.*, 2016] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *Proc. CVPR*, 2016.

[Liu and Yang, 2016] Jen-Yu Liu and Yi-Hsuan Yang. Event localization in music auto-tagging. In *Proc. MM*, 2016.

[McFee *et al.*, 2015] Brian McFee, Colin Raffel, Dawen Liang, Daniel P. W. Ellis, Matt McVicar, Eric Battenberg, and Oriol Nietok. librosa: Audio and music signal analysis in Python. In *Proc. scipy*, 2015.

[Mesaros *et al.*, 2016] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen. TUT database for acoustic scene classification and sound event detection. In *Proc. EUSIPCO*, 2016.

[Mesaros *et al.*, 2017] Annamaria Mesaros, Toni Heittola, Aleksandr Diment, Benjamin Elizalde, Ankit Shah, Emmanuel Vincent, Bhiksha Raj, and Tuomas Virtanen. DCASE 2017 Challenge setup: Tasks, datasets and baseline system. In *Proc. DCASE*, 2017.

[Piczak, 2015] Karol J. Piczak. ESC: Dataset for environmental sound classification. In *Proc. ACM MM*, 2015.

[Salamon *et al.*, 2014] J. Salamon, C. Jacoby, and J. P. Bello. A dataset and taxonomy for urban sound research. In *Proc. ACM MM*, 2014.

[Shen *et al.*, 2018] Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. DiSAN: Directional self-attention network for RNN/CNN-free language understanding. In *Proc. AAAI*, 2018.

[van den Oord *et al.*, 2013] Aaron van den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. In *Proc. NIPS*, 2013.

[Vu *et al.*, 2017] Toan Vu, An Dang, and Jia-Ching Wang. Deep learning for DCASE2017 Challenge. In *Proc. DCASE*, 2017.

[Xu *et al.*, 2017a] Yong Xu, Qiuqiang Kong, Qiang Huang, Wenwu Wang, and Mark D. Plumbley. Attention and localization based on a deep convolutional recurrent model for weakly supervised audio tagging. In *Proc. INTERSPEECH*, 2017.

[Xu *et al.*, 2017b] Yong Xu, Qiuqiang Kong, Wenwu Wang, and Mark D. Plumbley. Surrey-CVSSP system for DCASE2017 Challenge Task4. In *Proc. DCASE*, 2017.

[Zhang *et al.*, 2016] Han Zhang, Tao Xu, Mohamed Elhoseiny, Xiaolei Huang, Shaoting Zhang, Ahmed Elgammal, and Dimitris Metaxas. SPDA-CNN: Unifying semantic part detection and abstraction for fine-grained recognition. In *Proc. CVPR*, 2016.

[Zhu *et al.*, 2017] Feng Zhu, Hongsheng Li, Wanli Ouyang, Nenghai Yu, and Xiaogang Wang. Learning spatial regularization with image-level supervisions for multi-label image classification. In *Proc. CVPR*, 2017.