Improving weakly supervised sound event detection with self-supervised auxiliary tasks

Soham Deshmukh\textsuperscript{1}, Bhiksha Raj\textsuperscript{2}, Rita Singh\textsuperscript{1}\\
\textsuperscript{1}Microsoft, USA \hspace{1cm} \textsuperscript{2}Carnegie Mellon University, USA

\texttt{sdeshmukh@andrew.cmu.edu, bhiksha@cs.cmu.edu, rsingh@cs.cmu.edu}

Abstract

While multitask and transfer learning has shown to improve the performance of neural networks in limited data settings, they require pretraining of the model on large datasets beforehand. In this paper, we focus on improving the performance of weakly supervised sound event detection in low data and noisy settings simultaneously without requiring any pretraining task. To that extent, we propose a shared encoder architecture with the sound event detection as a primary task and an additional secondary decoder for a self-supervised auxiliary task. We empirically evaluate the proposed framework for weakly supervised sound event detection on a remix dataset of the DCASE 2019 task 1 acoustic scene data with DCASE 2018 Task 2 sounds event data under 0, 10 and 20 dB SNR. To ensure we retain the localisation information of multiple sound events, we propose a two-step attention pooling mechanism that provides a time-frequency localisation of multiple audio events in the clip. The proposed framework with two-step attention outperforms existing benchmark models by 22.3\%, 12.8\%, 5.9\% on 0, 10 and 20 dB SNR respectively. We carry out an ablation study to determine the contribution of the auxiliary task and two-step attention pooling to the SED performance improvement.\textsuperscript{1}

Index Terms: sound event detection, self-supervised learning, pooling function

1. Introduction

Sound Event Detection (SED) aims to determine the presence, nature and temporal location of sound events in audio signals. Many SED algorithms rely on strongly labelled data \cite{1, 2, 3} for training to perform accurate event detection and localisation. However, producing strongly labelled data for SED is quite expensive in terms of the expertise, time and human resources required for the annotation. This has led to the creation of weakly labelled sound event detection dataset like Audioset \cite{4} which contains audio clip level annotations without the corresponding onset and offset times of the audio events.

The weakly supervised sound event detection was first formulated as a Multiple-Instance Learning (MIL) problem \cite{5, 6} with the recent emergence of Neural MIL. In Neural MIL, the first half of the network (segmentation network) produces temporal predictions which are then aggregated by the second half of the network (classification network) usually a pooling operator to produce audio clip level predictions. The benefit of such formulation is, along with detecting audio events in the clip, it provides insight into time level localisation of those sound events in the audio clip. Since then, recent works have focused on improving the model architecture of the segmentation network \cite{7, 8, 9} and developing better pooling methods\textsuperscript{a}.

\textsuperscript{a}work done at Carnegie Mellon University

However, few works have focused on how sound event detection models perform in either limited data or noisy settings let alone in both of them. The noisy data also affects the training of networks for sound event detection. Specifically, the deep CNN architectures \cite{15, 16} currently used to provide benchmark performance for different speech and audio tasks \cite{17} require large labelled clean datasets to train on and when considered in a noisy environment the performance is known to deteriorate \cite{10}. The two general learning strategies used as solutions are transfer learning and multitask learning which were recently utilised for sound event detection \cite{18, 19, 20}. However, in the multitask learning setup, it’s assumed you have richly annotated labels for all the tasks. We investigate a counterpart of this where only weak labels are available without any labels for the secondary task. For this setting, we propose a self-supervised auxiliary task that will be jointly trained with the primary task of sound event detection. The auxiliary task is chosen to be the reconstruction of log Mel spectrogram of audio and we show how the auxiliary task denoises internal representations and improves network performance in noisy settings.

In all, in this paper, we address the challenge of training sound event detection models in noisy (domestic or environmental) and limited data settings. To that effort, we make two-fold contributions. First, identify appropriate self-supervised auxiliary task for sound event detection in noisy settings and demonstrate performance benefits to the same. Second, develop a two-step attention pooling mechanism that improves time-frequency localisation of audio events and indirectly improves sound event detection performance in noisy settings. We perform all the experiments on a standard noisy sound event detection dataset remix \cite{10} and release the code publicly.

2. Related work

A prominent recent work \cite{10} analysed the performance of different model architectures (segmentation network and pooling functions) under different Signal to Noise Ratio (SNR) for sound event detection and localisation. The paper showed that the segmentation network of type ‘VGG-like’ CNN performed best for audio tagging and variability in performance resulted from the choice of pooling methods with not a clear winning pooling method across different SNR. Specifically, Global Attention Pooling outperformed other pooling methods on some SNR and metrics, while Global Weighted Rank Pooling (GWRP) results in the best performance on others. Still, the work on sound event detection performance in limited data and noisy settings is sparse.

Though the various type of multitask learning methods have been greatly explored for vision and natural language processing (NLP) tasks \cite{21}, it has not been utilised by the audio community. Most of the works in multitask learning for SED fo-
Figure 1: Our proposed self-supervised learning assisted framework for weakly supervised sound event detection. (A) The general architecture with shared encoder and multiple decoder branches. Shared encoder, primary decoder, auxiliary decoder is represented by \( g_1, g_2, g_3 \) respectively. (B) shows the two step attention pooling function used for primary decoder. (C) The attention mechanism used for frequency and time attention in two step attention pooling along different axis. (D) The CNN architecture used for shared encoder and auxiliary decoder. The last layer is either class or reverse convolution for encoder and decoder respectively.

3. Methodology

This section contains the details of the proposed approach for SED, segmentation mapping network \( g_1 \), classification mapping network \( g_2 \), and the auxiliary time-frequency reconstruction auxiliary task. The architecture is depicted in figure 1

3.1. Self-supervised Learning formulation for SED

Let the raw audio be represented by \( X = \{x_i \}_{i=1}^T \) where each \( x_i \in \mathbb{R} \) is a frame in the audio clip. We extract time-frequency features for each audio, let them be represented by \( \hat{X} = \{\hat{x}_i \}_{i=1}^T \) where each \( \hat{x}_i \in \mathbb{R}^d \). Each of the audio events. The time-frequency feature extracted for audio here is log Mel spectrogram as it has shown to provide close to optimal performance on speech and audio tasks [20]. However, the work uses large scale speech datasets like LibriSpeech [24] as pretasks to pretrain the networks using self-supervised learning and does not analyse the effect of noise (domestic or environmental) on sound event detection performance.

3.2. Shared encoder and auxiliary task decoder network

The segmentation mapping function (shared encoder) converts the time-frequency audio input into a T-F representation for each of the audio events. The time-frequency feature extracted for audio here is log Mel spectrogram as it has shown to provide better performance [27, 28, 17]. We choose a CNN based architecture similar to ‘VGG-like’ [10] for both shared encoder and auxiliary task decoder. The shared encoder has CNN based network consists of 8 blocks of 2D Convolution, BatchNorm and ReLU with Average Pool after every two blocks. Having a common encoder helps the network to learn a shared representation by exploiting the similarity across SED and T-F reconstruction and enables the network to generalise better on our original task. We use a hard parameter sharing framework to reduce the risk of overfitting [29] to limited samples.

The decoder of the auxiliary-task takes \( \hat{Z} = \{\hat{z}_i \}_{i=1}^T \) as input and reconstructs it to \( \{\hat{x}_i \}_{i=1}^T \). The decoder consists of CNN based network for combining the intermediate time-frequency representations obtained for each audio event to an audio clip. If the learnable parameters are \( W = \{w, w_2, w_4\} \) and \( w, w_2, w_4 \) corresponding to \( g_1, g_2, g_3 \) respectively, then the optimisation problem can be framed in terms of these weights W over all data points as:

\[
\min_W \mathcal{L}_1(P, y|w, w_2) + \alpha \mathcal{L}_2(\{\hat{x}_i \}_{i=1}^T, \{\hat{x}_i \}_{i=1}^T|w, w_2) \quad (3)
\]

The parameter alpha (\( \alpha \)) accounts for scale difference between losses L1 and L2. It helps in adjusting the contribution of auxiliary task relative to the primary task in learning weights.

597
3.3. Primary decoder

The primary decoder is not CNN based, instead, it is a pooling operator to satisfy MIL formulation. The choice of pooling operator has a significant performance effect on both the SED and each audio event intermediate time-frequency representation. Global max pooling and global average pooling results in underestimate and overestimate the audio event’s temporal presence respectively, and to overcome this problem dynamic poolings were proposed [10, 12, 14]. However, the developed pooling mechanisms still lacks the granularity in temporal predictions and does not provide frequency localisation which might be used to further disambiguate sound events. Also, the sigmoid operator has a significant performance effect on both the SED except it operates along time axis:

\[ Z_{a1} = \frac{e^{\sigma(WZ_{p1} + b_{p1})}}{\sum_{i=1}^{F} e^{\sigma(WZ_{c1,i} + b_{c1,i})}} \]

(4)

This is followed by a weighted combination of classification output \( Z_{c1} \) by attention weights \( Z_{a1} \):

\[ Z_{p1} = \sum_{i=0}^{F} Z_{c1,i} \cdot Z_{a1} \]

(5)

The time level attention is similar to frequency (first step) attention except it operates along time axis:

\[ Z_{a2} = \frac{e^{\sigma(Z_{p1}W_{a2} + b_{a2})}}{\sum_{t=1}^{T} e^{\sigma(Z_{p2}W_{a2} + b_{a2})}} \]

\[ Z_{p2} = \sum_{t=0}^{T} Z_{c2,t} \cdot Z_{a2} \]

(7)

where \( Z_{p2} \in [0, 1] \) and denotes the presence probability of each sound event in the audio clip. Figure 1 subsection c, provides an overview of a single attention step. In relation to figure, \( Z_{a}, Z_{c}, Z_{p} \) are the outputs after attention matrix, classification matrix and \( P(.) \) respectively in the first stage and second stage depending on subcript. By breaking the attention into two steps, it makes the pooling more interpretable by answering the questions of what frequency bins and what time steps contributes to which audio events by visualising normalised attention weights \( Z_{a1}, Z_{a2} \) and output \( Z_{p1}, Z_{p2} \). Also, the sigmoid (\( \sigma \)) ensures the attention output stays between 0 to 1 and avoids unstable training for multilabel training with cross-entropy in practice.

4. Experiments

4.1. Dataset

To study the effect of noise in limited data settings, we form a noisy dataset by mixing DCASE 2019 Task 1 of Acoustic scene classification [30] and DCASE 2018 Task 2 of General purpose Audio tagging [31]. The DCASE 2019 Task 1 provides background sounds (noise) recorded from a variety of real world scenes in which the sounds from DCASE 2019 Task 2 are randomly embedded [10]. To ensure the noise conditions are natural, diverse and challenging, we use the new DCASE 2019 Task 1 instead of DCASE 2018 as used in [10]. The 2019 variant extends the TUT Urban Acoustic Scenes 2018 with the other 6 cities to a total of 12 large European cities. This results in 32000 audio clips with 8000 audio clips for each 20,10,0 dB SNR where each audio clip is of 10 secs with background noise and three random audio events (out of total 41) in it.

4.2. Set up

The raw data is converted to time-frequency representation by applying FFT with a window size of 2048 and an overlap of 1024 between windows. This is followed by applying Mel filter banks with 64 bands and converting them to log scale to obtain log Mel spectromgram. The network architecture used is described in section 3.2. The entire network is trained end-to-end with a batch size of 24 and learning of 1e-3 using Adam optimiser [32]. The code and setup is publicly released\(^1\).

5. Results

5.1. Sound event detection

We evaluate our self-supervision assisted architecture and pooling method against different baselines, benchmark architectures and pooling methods [14, 10]. Table 1 shows weakly supervised sound event detection performance across different SNR of 20,10, and 0 dB. The important evaluation metric here under consideration is micro precision (micro-p), as it uses global counts of true positives, false negatives and false positives for metric computation against macro precision which does simple unweighted averaging disregarding class-imbalance. The VGGish encoder here indicates VGGish architecture but with dilated/atrous convolutions known to provide benchmark performance for sound event detection, [14]. The VGGish encoder with reconstruction based auxiliary task and two step attention pooling outperforms the existing benchmark of atrous attention pooling [14] on SNR 20, 10 and 0 dB by 5.9%, 12.8% and 22.3% respectively. Apart from improving performance, by breaking the attention into two steps, it allows for the intermediate use of sigmoid which helps in ensuring the outputs don’t overflow above 1 during training.

\(^1\)https://github.com/soham97/MTL_Weakly_labelled_audio_data

Table 1: Weakly supervised sound event detection performance across different SNR

<table>
<thead>
<tr>
<th>Network</th>
<th>encoder</th>
<th>pooling</th>
<th>aux.</th>
<th>SNR 20 dB</th>
<th>SNR 10 dB</th>
<th>SNR 0 dB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>micro-p</td>
<td>macro-p</td>
<td>AUC</td>
</tr>
<tr>
<td>VGGish</td>
<td>GAP</td>
<td>x</td>
<td></td>
<td>0.5067</td>
<td>0.6127</td>
<td>0.9338</td>
</tr>
<tr>
<td>VGGish</td>
<td>GMP</td>
<td>x</td>
<td></td>
<td>0.3590</td>
<td>0.5186</td>
<td>0.8497</td>
</tr>
<tr>
<td>VGGish</td>
<td>GWRP</td>
<td>x</td>
<td></td>
<td>0.7018</td>
<td>0.7522</td>
<td>0.9362</td>
</tr>
<tr>
<td>VGGish</td>
<td>GMP</td>
<td></td>
<td></td>
<td>0.7391</td>
<td>0.7586</td>
<td>0.9279</td>
</tr>
<tr>
<td>VGGish</td>
<td>2AP</td>
<td>✓</td>
<td></td>
<td>0.7829</td>
<td>0.7645</td>
<td>0.9390</td>
</tr>
</tbody>
</table>

598
Table 2: Ablation study to determine auxiliary task contribution

<table>
<thead>
<tr>
<th>auxiliary task</th>
<th>SNR 20 dB</th>
<th>SNR 10 dB</th>
<th>SNR 0 dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>α = 0.0</td>
<td>0.7772</td>
<td>0.7430</td>
<td>0.6937</td>
</tr>
<tr>
<td>α = 0.001</td>
<td>0.7829</td>
<td>0.7603</td>
<td>0.6986</td>
</tr>
<tr>
<td>α = 0.1</td>
<td>0.7637</td>
<td>0.7428</td>
<td>0.6792</td>
</tr>
</tbody>
</table>

5.2. Ablation study for auxiliary task contribution

We perform an ablation study to determine the contribution of reconstruction auxiliary task and two step attention pooling towards the total performance improvement. As described in Section 3.1, the total loss is:

$$L = L_1(P, y|w, w_4) + \alpha L_2(\{\hat{x}_i\}_{i=1}^T, \{\hat{x}_i\}_{i=1}^T|w, w_2)$$  \hspace{1cm} (8)

By changing the value of α before training, we can adjust the contribution of the auxiliary task to primary sound event detection. When α = 0.0, the network has no contribution from the reconstruction auxiliary task during training and it can be used to evaluate the performance of two step attention pooling. In terms of micro-precision, the two step attention pooling outperforms existing benchmark of atrous AP (row 4) from table 1 on SNR 20, 10 and 0 dB by 5.2%, 10.2% and 21.4% respectively. By adding the auxiliary task contribution with a relative weightage of α = 0.001, an additional improvement of 0.7%, 2.3% and 0.7% is observed. This indicates that two step attention has a prominent contribution in improving the performance of sound event detection in limited data and noisy settings, with additional performance gains from the auxiliary task. When α is increased to 0.01, the performance compared to α = 0.001 is decreased. This suggests that the auxiliary task’s loss contribution starts to overpower the primary SED task’s loss contribution rather than improving generalisation.

Table 3: Two top and worst performing sound events- SNR 0 dB

<table>
<thead>
<tr>
<th>model</th>
<th>aux.</th>
<th>bus</th>
<th>cowbell</th>
<th>gong</th>
<th>meow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atrous + AP</td>
<td>X</td>
<td>0.2</td>
<td>0.781</td>
<td>0.692</td>
<td>0.583</td>
</tr>
<tr>
<td>VGGish + 2AP</td>
<td>X</td>
<td>0.572</td>
<td>0.921</td>
<td>0.643</td>
<td>0.483</td>
</tr>
<tr>
<td>VGGish + 2AP</td>
<td>✓</td>
<td>0.627</td>
<td>0.94</td>
<td>0.663</td>
<td>0.532</td>
</tr>
</tbody>
</table>

5.3. SED performance on specific audio events

For almost all audio events, our proposed architectures have the best precision scores against GMP, GAP, GWRP. Atrous across all SNR = 0, 10, 20. Particularly, for audio events like ‘Bass drum’, ‘bus’, ‘double bass’, ‘cowbell’ the architecture outperforms other models by a large margin as shown in table 3. However, the proposed model struggles in audio events like ‘gong’, ‘chime’ and ‘meow’ where the attention pooling with dilated convolution encoder performs better [14]. This indicates using atrous or dilated convolutions helps in detecting audio events whose energy is spread wide in the temporal domain. This can be incorporated into our current architecture by replacing the linear convolutions in the shared encoder with dilated convolutions. Further analysis and event-specific results are available in the long version of paper 3 and skipped due to space constraints.

5.4. Interpretable visualisation of audio events

Apart from improved performance, using two step attention pooling provides a way to localise each audio event present


Figure 2: Visualisation of two step attention pooling and reconstruction decoder outputs. Subplot 1 depicts the scaled log Mel spectrogram of an audio clip. Subplot 2 is the output of the reconstruction auxiliary task. Subplots 3,4 and 5 are attention weights for the most probable audio events in the audio clip. Subplot 6 is the output of the first step attention pooling. Subplot 7 and 8 is the attention weight and output of second step attention pooling respectively. The y-axis in subplot 1-4 corresponds toMel-h bins and sound events in subplot 5-6. The x-axis in subplot 1-7 corresponds to time and sound events in subplot 8 in the audio clip along with both the time and frequency axis. To illustrate this, we pick a random example with SNR 20 dB and show the end to end visualisation of the two step attention pooling mechanism in figure 2. The audio under consideration has three events occurring in it: telephone ringing, cello playing and cat meowing, with outdoor environmental background noise. Subplot 2 in figure 2 depicts the reconstructed Mel spectrogram of the audio clip. From the subplot, we can see that the decoder is not only able to reconstruct the audio events clearly but it is also denoising the log Mel spectrogram retaining the key elements of three audio events. A future extension of work is to jointly train sound source separation along with weakly supervised SED by using the auxiliary task reconstruction output.

6. Conclusions

This paper proposes assisted self-supervised task for improving sound event detection in limited data and noisy settings. The architecture consists of sound event detection as a primary task with two-step attention pooling as a primary decoder and time-frequency representation reconstruction as an auxiliary task. We empirically evaluate the proposed framework for multi-label weakly supervised sound event detection, on a remix DCASE 2019 and 2018 dataset under 0, 10 and 20 dB SNR. The proposed self-supervised auxiliary task framework with two-step attention outperforms existing benchmark models by 22.3 %, 12.8 %, 5.9 % on 0, 10 and 20 dB SNR respectively. The ablation study carried out indicates the majority of performance improvement is associated with two step attention pooling with secondary performance improvement from self-supervised auxiliary task. Furthermore, by using two step attention, we can easily visualise the sound event presence along both time-frequency axis. The code is publicly released.
7. References


