INTEGRATING ISOLATED EXAMPLES WITH WEAKLY-SUPERVISED SOUND EVENT DETECTION: A DIRECT APPROACH

Mohammad Abdollahi, Romain Serizel, Alain Rakotomamonjy, Gilles Gasso

Universite de Rouen

ABSTRACT

In an attempt to mitigate the need for high quality strong annotations for Sound Event Detection (SED), an approach has been to resort to a mix of weakly-labelled, unlabelled and a small set of representative (isolated) examples. The common approach to integrate the set of representative examples into the training process is to use them for creating synthetic soundscapes. The process of synthesizing soundscapes however could come with its own artefacts and mismatch to real recordings and harm the overall performance. Alternatively, a rather direct way would be to use the isolated examples in a form of template matching. To this end in this paper we propose to train an isolated event classifier using the representative examples. By sliding the classifier across a recording, we use its output as an auxiliary feature vector concatenated with intermediate spectro-temporal representations extracted by the SED system. Experimental results on DESED dataset demonstrate improvements in segmentation performance when using auxiliary features and comparable results to the baseline when using them without synthetic soundscapes. Furthermore we show that this auxiliary feature vector block could act as a gateway to integrate external annotated datasets in order to further boost SED system's performance.

Index Terms— sound event detection, deep learning, posteriorgrams, weakly-supervised learning

1. INTRODUCTION

Sound Event Detection (SED) is a machine listening task that addresses the questions of *What* and *When* of occurring audio events, the responses to which would be the perceptual class of an event and its location in time respectively. The necessity of such SED systems manifests itself in applications such as audio information retrieval [1], surveillance [2], bioacoustic monitoring [3] and self-driving cars [4] to a name a few. One main challenge with SED is the inherent difficulty and cost of acquiring annotated data of high quality/resolution. Hence there has been a growing body of research in development of models using lower quality annotations (lower temporal resolution, noisy labels, etc.) and un-annotated data examples potentially augmented with a small set of (isolated) representative examples of each target event class.

Isolated examples are generally used indirectly through synthesizing soundscapes [5], to generate *strongly-labelled* examples. The relative contribution of each level of annotations in this heterogeneous dataset has been studied by Turpault et al [6]. Surprisingly, their findings indicate the relative equal contribution of both sets of weak and strongly (synthetic) labelled examples to system's segmentation performance. In pre-deep learning speech recognition, *Phoneme posteriorgrams* output by pre-trained phoneme classifiers were used as features fed to the downstream blocks[7]. In spoken keyword detection, Chen et al. [8] trained a deep neural network (DNN) to classify among a number of keywords and by sliding the DNN across an audio recording used the output scores (followed by post-processing) to mark the keyword boundaries. Similarly, isolated examples could be used for *template matching* by sliding them across recordings to generate similarity scores and using these scores for detection of event boundaries.

In this paper we propose to integrate a similar approach into training the SED system. Rather than using the similarity scores directly for decision making, we will use them as auxiliary features to enhance the performance of the Weakly-Supervised SED (WSSED) system. Beyond the isolated examples, we also investigate the utility of using classifiers trained on other (non-target) external datasets (e.g. ESC-50) to generate the auxiliary features and report performance gains.

2. BASELINE SED SYSTEM

In this work, we have chosen a CNN-Transformer architecture proposed by Myazaki et al. [9] as our baseline model. Similar to a Convolutional Recurrent Neural Network (CRNN), CNN-Transformer uses a Convolutional Neural Network (CNN) for extraction of intermediate representations, but instead of a RNN, it is followed by

Thanks to ANR agency for funding the project LEAUDS.

few layers of transformer encoder for temporal dependency modelling.

Given a time-frequency representation of an audio signal as input (e.g. Mel-spectrogram) $\boldsymbol{X}_{[F^i \times T^i]}$ where F^i and T^i represent the dimensions of the input along frequency and time respectively, a CNN extracts a representation $\boldsymbol{H}_{[C \times F^e \times T^e]}$ with F^e and T^e being the resulting (downsampled) frequency and time dimensions and C the number of extracted feature maps. Merging the F^e and C dimensions, would result a T^e-long sequence of $d = C \times F^e$ dimensional feature vectors $\boldsymbol{h}_i$:

$$\boldsymbol{H} = \{\boldsymbol{h}_1, \dots, \boldsymbol{h}_{T^e}\} \tag{1}$$

Every element of this sequence is first linearly projected to a lower dimensional space using an *Embedding layer*. The resulting sequence of embedding vectors, after being added to the respective positional encoding vectors are passed through N layers of transformer encoders.

A final single-layer Multi-layer Perceptron (MLP) is applied on all elements of the output sequence of the transformer to predict the segment-based labels. The clip-level label is obtained by performing attention-pooling [10] over all elements in the output sequence.

3. PROPOSED APPROACH

In this work we take a more direct approach in integrating isolated examples into training a WSSED system. Rather than synthesizing soundscapes with them, we propose to use them to build an isolated event classifier. The idea is that, given a large enough set of isolated examples of the target events, we could build a classifier operating on a certain time scale and slide it across test audio examples in order to generate similarity/likelihood scores. These scores could then be used (in addition to some post-processing) to detect target event boundaries. Such an approach would be very similar to the recent line of work on Keyword Spotting (KWS) [8, 11]. Unlike in the case of KWS however, our set of examples is much smaller and usually suffers from more intra-class acoustic variability. In this paper we propose to integrate a similar approach into training the SED system by using the output of an isolated event classifier as an additional feature. We leverage transfer learning to train the classifier in order to mitigate the effects of low sample-size training dataset. In fact, we use the same pre-trained backbone CNN of the CNN-transformer baseline, and train a single-layer MLP on top of it.

In our proposed system, each segment of the temporal sequence $\boldsymbol{H}$ at the CNN's output is passed to the MLP and the classifier's output scores are appended to the spectro-temporal features of that segment. The overall schematic of the proposed system is illustrated in figure 1 in which we have introduced an *Auxiliary Feature generator* block.

3.1. Event Classifier

Since the goal is to use the classifier for detection and to produce an output probability vector at a specified temporal resolution, we train a point-wise MLP operating on each $\boldsymbol{h}_t$ independently. This means we broadcast the event label to all of it sub-segments across time. Nevertheless, in order to further expand the effective receptive field input to the MLP when classifying $\boldsymbol{h}_t$, we concatenate a set of $2k+1$ dilated neighbouring feature slices centred around each $\boldsymbol{h}_t$ to be passed to the MLP:

$$\boldsymbol{o}_t = MLP(\begin{bmatrix} \boldsymbol{h}_{t-(k.d)} \\ \vdots \\ \boldsymbol{h}_t \\ \vdots \\ \boldsymbol{h}_{t+(k.d)} \end{bmatrix}) \tag{2}$$

with o_t being the classifier's output at time instance t and d the dilation factor. Similar to the SED system's output, in order for the classifier to account for event polyphony and non-target sound events, we train the classifier with a *Binary Cross-Entropy (BCE)* loss on sigmoid-activated outputs over the 10 classes of target events. We refer to the output vector in this case as a *posteriorgram.*

3.2. Auxiliary Feature Generator Block

The above trained MLP classifier is integrated into the SED system in the form of an *Auxiliary Feature Generator* (AFG) block where the trained MLP is moved across the extracted representations $\boldsymbol{h}_t$ of a recording and generates a posteriorgram $\boldsymbol{o}_t$ for each corresponding $\boldsymbol{h}_t$. We further take the logarithm of these probability scores before using them as features in the rest of the system.

Additionally, in order to emphasize the points in time where the classifiers output experience sharp changes (potential event boundaries) we propose to augment each log-probability vector with its first order dynamics $\boldsymbol{d}_t$:

$$\boldsymbol{d}_t = \sum_{n=1}^{N} n \times (\log(\boldsymbol{o}_{t+n}) - \log(\boldsymbol{o}_{t-n})) \tag{3}$$

where N is the width of the window used for the calculation of the derivative. The resulting output feature of the AFG block is the concatenation of each $\log(\boldsymbol{o}_t)$ and $\boldsymbol{d}_t$ yielding the auxiliary feature sequence $\boldsymbol{P}$:

$$\boldsymbol{P} = \{\boldsymbol{p}_1, \dots, \boldsymbol{p}_{T^e}\} = \{\begin{bmatrix} \log(\boldsymbol{o}_1) \\ \boldsymbol{d}_1 \end{bmatrix}, \dots, \begin{bmatrix} \log(\boldsymbol{o}_{T^e}) \\ \boldsymbol{d}_{T^e} \end{bmatrix}\} \tag{4}$$

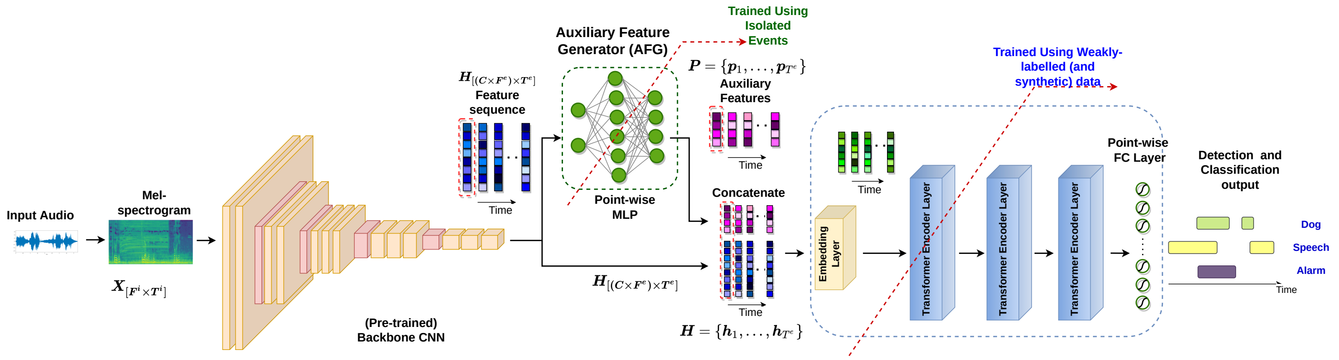
Input Audio
Mel-spectrogram
$X_{[F^i \times T^i]}$
(Pre-trained) Backbone CNN
$H_{[(C\times F^e)\times T^e]}$
Feature sequence
Time
Auxiliary Feature Generator (AFG)
Point-wise MLP
Trained Using Isolated Events
$P = \{p_1, \ldots, p_{T^e}\}$
Auxiliary Features
Time
Concatenate
$H_{[(C\times F^e)\times T^e]}$
Time
$H = \{h_1, \ldots, h_{T^e}\}$
Time
Embedding Layer
Transformer Encoder Layer
Transformer Encoder Layer
Transformer Encoder Layer
Trained Using Weakly-labelled (and synthetic) data
Point-wise FC Layer
Detection and Classification output
Dog
Speech
Alarm
Time

Fig. 1: General schematic of the proposed approach with the AFG block

As illustrated in figure 1, the AFG output *features* $\boldsymbol{p}_t$ are concatenated with the $\boldsymbol{h}_t$ and the sequence of tandem feature slices are batch-normalized before being passed to the embedding layer and the succeeding transformer encoder layers which will be trained using the weakly-labelled (and synthetic) examples.

3.3. Training Procedure

In this work, while training the SED system, we keep the parameters of the classifier (MLP and the backbone CNN) frozen and use the rest of weakly- and strongly-annotated examples only to train the embedding layer and the transformer layers. Given that the CNN is already pre-trained using a large dataset, this should not be a limiting factor on the system's learning capacity. Nevertheless training the entire system end-to-end, which given two tasks and two datasets would be framed as a multitask learning problem, seems more appealing and we intend to address that in our future works.

4. EXPERIMENTS

4.1. Model Parameters

In this work rather than using a shallow CNN as in the original baseline CRNN model of DCASE task 4, we use a deeper Efficientnet-B2 model [12] used in a work by by Gong et al. [13] pre-trained on Audioset [14].

Each 10-second audio clip is resampled at the rate of 16kHz and is converted to log Mel-spectrogram with 25ms window length, 10ms window hop and 128 Mel bands. The resulting 2D array is passed through the backbone CNN of the system.

The Efficientnet-B2 final convolutional layer yields a feature map of 1408 × 4 × 33 dimension as its output. After average pooling along the frequency axis we end up with a temporal sequence of 33 1408-dimensional feature vectors. This sequence is passed through an embedding layer to produce a sequence of 256-dimensional embedding vectors, which in turn is passed through the transformer encoder layers. We use 3 layers of transformer encoder blocks, with 512 as the hidden dimension of the positional feed-forward network and 4 attention heads. In the proposed approach the features in this sequence are concatenated with the AFG's output features, and passed to the embedding layer to be projected to a sequence of the 256-dimensional vectors similar to the baseline model. The same pre-trained CNN is used for training the MLP in the AFG block. We chose width $k = 1$ and dilation $d = 2$ to define the temporal extent of information resulting in a 4224-dimensional input to the MLP. The MLP consists of a 512 unit hidden layer with batch-normalization followed by ReLU activation function. With the 10 output classes of the isolated classifier, the auxiliary features will be 20-dimensional vectors. Since the number and duration of examples per event class are highly variable in the set of isolated examples we adopt a weighted sampling strategy in each batch to train the classifier. For training both the SED system and the classifier we use data augmentation pipeline consisting of mixup [15], frequency and time masking [16] and random additive gaussian noise.

4.2. Datasets

4.2.1. SED Datasets

The datasets that were provided as part of DCASE 2021 task4 challenge (DESED) are listed below.

Weakly-annotated data: 1578 10-second long audio clips of domestic recordings taken from Audioset [14] with verified clip-level labels.

Unlabelled Data: 14412 clips of 10-second clips selected from Audioset. We did not use the unlabelled data in our experiments.

Soundbank of isolated events: 1009 isolated examples of target events taken from FSD50k [17] with clip durations ranging from 55ms to 83.1s. The median

	w. Synth	F1	PSDS1	PSDS2
Baseline	✗	23.21	0.097	0.336
	✓	38.92	0.251	0.512
Ours	✗	35.25	0.183	0.491
	✓	40.20	0.263	0.523

Table 1: A comparative performance evaluation of proposed AFG block on WSSED system

number of examples per class is 62 and the median total duration of examples per class is 310s.

Synthetic Soundscapes: a collection of 10000 synthetically generated soundcapes using the Scaper [18] library using the soundbank of isolated events and a selected set of background sounds from SINS dataset [19].

4.2.2. External Datasets

We also used an external dataset for auxiliary feature generation in our experiments.

ESC-50: a collection of 2000 event clips all of 5s durations across 50 sound event classes each containing 40 examples [20].

4.3. Results

Table 1 summarizes the performance of our proposed approach against the baseline CNN-Transformer system. The results are reported on the provided validation set. The performance is reported using intersection-based F1 macro score and PSDS measures. PSDS1 and PSDS2 are two specific hyperparameter settings of the PSDS measure where the former emphasizes more accurate localization while the focus of latter is on distinguishing classes from one another. In order to evaluate the potential of AFG block to complement or even replace the use of synthetic soundscapes, we report the performance of the baseline system when trained only with weakly-labelled examples alone (w/o synth) and when including the synthetic soundscapes (w. synth).

4.4. AFG, a Gateway for External Datasets

Given the generality of the AFG block, we further investigate the utility of it as a gateway to integrate external annotated datasets. The idea is that nothing restricts the MLP to be trained on the provided soundbank of isolated examples. Even if the target classes of an external dataset does not overlap with the SED task at hand, projecting slices of a recording into a classifier trained on a externally-annotated dataset could provide relevant discriminative features and hence reduce the uncertainty inherent in the weakly-labelled examples.

	w. Synth	F1	PSDS1	PSDS2
Baseline	✗	23.21	0.097	0.336
	✓	38.92	0.251	0.512
AFG: ESC-50	✗	28.33	0.131	0.451
	✓	39.56	0.227	0.536
AFG: ESC-50 + isolated examples	✗	38.22	0.202	0.553
	✓	41.31	0.251	0.574

Table 2: Effect of using external datasets for feature generation on WSSED system's performance

Table 2 lists the results of using the ESC-50 dataset to generate auxiliary features and results are reported for both when used alone and when combined with features from isolated examples. When combined, we concatenate the auxiliary features from both trained classifiers before projecting them into the transformer's embedding dimension. The results are shown for both with and without the use of synthetic dataset in training the SED system. Our results suggest this approach as a rather simple way to distil discriminative information inherent in external datasets into WSSED with minimal effort and without requiring any further annotations with respect to the target task.

5. CONCLUSIONS

In this work we have presented a different way of incorporating a small set of isolated examples into training a WSSED system. Rather than using them to create synthetic soundscapes, we proposed to use the output of a classifier built using them as an auxiliary feature for each sub-segment of an input recording. We have evaluated our approach using DCASE2021 task-4 setup and dataset. Our results suggest that by means of introducing this feature augmentation we could improve the performance of the baseline system with or without synthetic soundscapes. In addition we showed that the auxiliary features could be generated using external datasets to encode extra knowledge into the system and improve the performance.

In our future work we intend to explore other ways of auxiliary feature generation particularly memory-based methods using the advances in deep metric learning and few-shot learning. Moreover, as the auxiliary feature block and the main SED system share the backbone network, the system could be end-to-end optimized through framing it as a multi-task learning problem.

6. REFERENCES

[1] E. Wold, T. Blum, D. Keislar, and J. Wheaten, "Content-based classification, search, and retrieval

of audio," *IEEE MultiMedia*, vol. 3, no. 3, pp. 27–36, 1996.

[2] R. Radhakrishnan, A. Divakaran, and A. Smaragdis, "Audio analysis for surveillance applications," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2005.*, 2005, pp. 158–161.

[3] Justin Salamon, Juan Pablo Bello, Andrew Farnsworth, Matt Robbins, Sara Keen, Holger Klinck, and Steve Kelling, "Towards the automatic classification of avian flight calls for bioacoustic monitoring," *PLoS One*, vol. 11, no. 11, Nov. 2016.

[4] Marco Cristani, Manuele Bicego, and Vittorio Murino, "Audio-visual event recognition in surveillance video sequences," *IEEE Transactions on Multimedia*, vol. 9, no. 2, pp. 257–267, 2007.

[5] Romain Serizel, Nicolas Turpault, Ankit Shah, and Justin Salamon, "Sound event detection in synthetic domestic environments," in *Proc. ICASSP*, 2020, pp. 86–90.

[6] Nicolas Turpault and Romain Serizel, "Training sound event detection on a heterogeneous dataset," in *DCASE Workshop*, 2020.

[7] Gethin Williams and Daniel P. W. Ellis, "Speech/music discrimination based on posterior probability features," in *EUROSPEECH*, 1999.

[8] Guoguo Chen, Carolina Parada, and Georg Heigold, "Small-footprint keyword spotting using deep neural networks," in *Proc. ICASSP*, 2014, pp. 4087–4091.

[9] Koichi Miyazaki, Tatsuya Komatsu, Tomoki Hayashi, Shinji Watanabe, Tomoki Toda, and Kazuya Takeda, "Weakly-supervised sound event detection with self-attention," in *Proc. ICASSP*, 2020, pp. 66–70.

[10] Yun Wang, Juncheng Li, and Florian Metze, "A comparison of five multiple instance learning pooling functions for sound event detection with weak labeling," in *Proc. ICASSP*, 2019, pp. 31–35.

[11] Tara N. Sainath and Carolina Parada, "Convolutional neural networks for small-footprint keyword spotting," in *Proc. Interspeech 2015*, 2015, pp. 1478–1482.

[12] Mingxing Tan and Quoc Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[13] Yuan Gong, Yu-An Chung, and James Glass, "PSLA: Improving audio tagging with pretraining, sampling, labeling, and aggregation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3292–3306, 2021.

[14] 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. IEEE ICASSP 2017*, New Orleans, LA, 2017.

[15] Yuji Tokozume, Yoshitaka Ushiku, and Tatsuya Harada, "Learning from between-class examples for deep sound recognition," in *International Conference on Learning Representations*, 2018.

[16] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *Proc. Interspeech 2019*, 2019, pp. 2613–2617.

[17] Eduardo Fonseca, Xavier Favory, Jordi Pons, Frederic Font, and Xavier Serra, "Fsd50k: An open dataset of human-labeled sound events," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 30, pp. 829–852, jan 2022.

[18] Justin Salamon, Duncan MacConnell, Mark Cartwright, Peter Li, and Juan Pablo Bello, "Scaper: A library for soundscape synthesis and augmentation," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2017, pp. 344–348.

[19] Gert Dekkers, Steven Lauwereins, Bart Thoen, Mulu Weldegebreal Adhana, Henk Brouckxon, Bertold Van den Bergh, Toon van Waterschoot, Bart Vanrumste, Marian Verhelst, and Peter Karsmakers, "The SINS database for detection of daily activities in a home environment using an acoustic sensor network," 2017, pp. 1–5, DCASE Workshop.

[20] Karol J. Piczak, "ESC: Dataset for Environmental Sound Classification," in *Proceedings of the 23rd Annual ACM Conference on Multimedia*. pp. 1015–1018, ACM Press.