SELF-TRAINED AUDIO TAGGING AND SOUND EVENT DETECTION IN DOMESTIC ENVIRONMENTS

Technical Report

Janek Ebbers, Reinhold Haeb-Umbach

Paderborn University, Department of Communications Engineering, Paderborn, Germany
{ebbers, haeb}@nt.upb.de

ABSTRACT

In this report we present our system for the *Detection and Classification of Acoustic Scenes and Events (DCASE) 2021 Challenge Task 4: Sound Event Detection and Separation in Domestic Environments*. Our presented solution is an advancement of our system used in the previous edition of the task.We use our previously proposed forward-backward convolutional recurrent neural network (FBCRNN) for tagging and pseudo labeling and tag-conditioned sound event detection (SED) models which are trained using the strong pseudo labels provided by the FBCRNN. Our advancement over our previous model is threefold. Firstly, we introduce a strong label loss in the objective of the FBCRNN to take advantage of the strongly labeled synthetic data during training, which leads to both better tagging and detection performance. Secondly, we perform multiple iterations of self-training for both the FBCRNN and tag-conditioned SED models. Thirdly, while we used only tag-conditioned CNNs as our SED model in the last edition we here explore sophisticated SED model architectures, namely, tag-conditioned bidirectional CRNNs and tag-conditioned bidirectional convolutional transformer neural networks (CTNNs) and combine them. With scenario and class dependent tuning of median filter lengths for post-processing, our final SED model, consisting of 6 submodels (2 of each architecture), is able to achieve validation poly-phonic sound event detection scores (PSDS) of 0.454 for scenario 1 and 0.758 for scenario 2 as well as a collar-based F1-score of 0.602 outperforming the baselines and our model from the last edition by far. Source code will be made publicly available at `https://github.com/fgnt/pb_sed`.

Index Terms— sound event detection, audio tagging, weak labels, self-training

1. FORWARD-BACKWARD CRNN

The FBCRNN [1] is illustrated in Fig. 1. It consist of a shared CNN front-end and two separate recursive classifier networks (RNN+fully connected neural network (FCN)) with one processing the audio in forward direction and the other in backward direction. Note that unlike a bidirectional RNN the two classifiers do not exchange hidden representations and, therefore, at each frame one classifier has only seen previous frames and the other only subsequent frames.

To encourage the model to output tag predictions as soon as it has seen the event in the input, as shown in Fig. 2, we compute, at each frame, the binary cross entropy (BCE) loss between the pointwise maximum of the predictions of the two classifiers and the clip-level (weak) label. Note, that this can be seen as multiple instance learning (MIL) with two instances. One instance comprises the current plus all previous frames, which has been processed by the forward classifier, and the other instance comprises the current plus all subsequent frames, which has been processed by the backward classifier. Hence, if an event is labeled positive in the clip at least one of the classifiers has to be able to classify the event as positive, given that the event is either present in previous or in subsequent frames or both. This training scheme forces the two classifiers to output predictions without having processed the whole audio, which makes it generalize to much shorter segments later on of, e.g., only a couple of hundred milliseconds and, hence, enables SED.

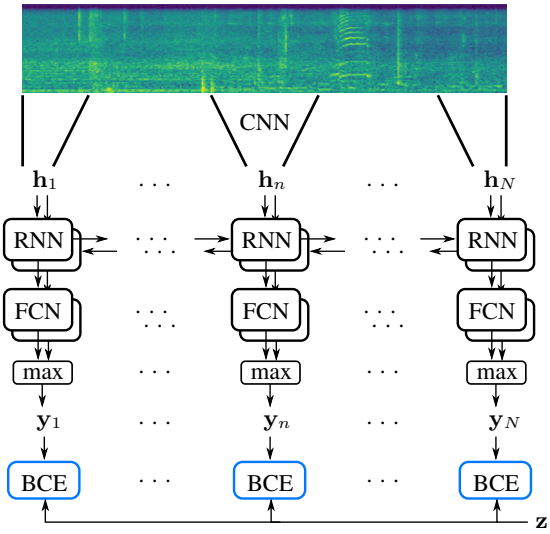

Figure 1: FBCRNN

We use the same architecture as in [1]. We only removed the last pooling layer between the Conv2d and Conv1d blocks.

1.1. Strong Label Loss

As the training data of the challenge contains synthetic data which comes with strong labels, it is desirable to make use of the strong labels in the FBCRNN training, which we previously did not do. If strong labels are given, we now, instead of the weak label loss, compute a strong label BCE for both classifiers with respect to the desired outputs shown in Fig. 2 and average the two loss terms.

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - 282835863. Computational resources were provided by the Paderborn Center for Parallel Computing.

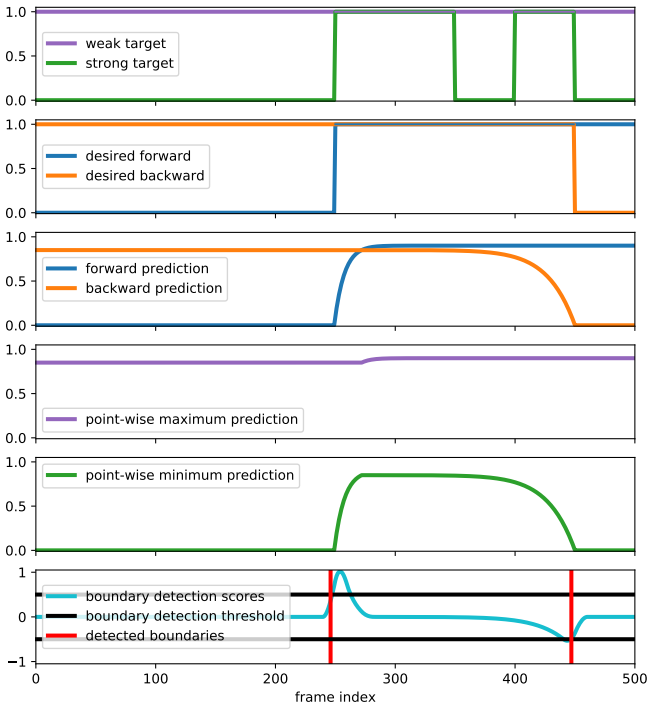

Figure 2: FBCRNN signals

1.2. Self-Training

As a large fraction of the provided data is unlabeled, we now perform more extensive self-training with training 8 initial FBCRNNs on only weakly labeled real and strongly labeled synthetic data followed by three iterations of pseudo labeling and retraining 4 FBCRNN models in each iteration.

In each iteration we generate weak pseudo labels for the complete unlabeled data, where tagging thresholds are tuned on the validation set to maximize the F1-score. Additionally, we perform an boundary detection for weakly labeled and unlabeled data by filtering the point-wise minimum of the two classifier signals with $\begin{bmatrix}-2/N & \dots & -2/N & 2/N & \dots & 2/N\end{bmatrix}$ where N is the filter size. The class-specific filter sizes and thresholds that the output or negative output has to exceed to detect an onset or offset boundary, respectively, are tuned on the validation data such that a minimum precision of $75\,\%$ is achieved, when using a detection collar of $500\,\mathrm{ms}$. For those events where onset and/or offset can be detected, the strong label loss from Sec. 1.1 is used in the following FBCRNN retraining. The different signals are illustrated in Fig. 2.

Finally, we use both the FBCRNNs after the second and third iteration to separately perform strong pseudo labeling of the weakly labeled and unlabeled data giving us two different sets of strong pseudo labels. To achieve SED with the FBCRNNs, it is applied to a small context of a couple of $100\,\mathrm{ms}$ around each frame to generate an SED score at that frame.Here, class specific context lengths, median filter lengths and detection thresholds are tuned on the validation set to maximize the frame-based F1-score. The obtained strong pseudo labels allow us to train SED systems in a strongly supervised manner as described in the following.

For tag-conditioning at test-time we use all of the 8 models jointly to perform audio tagging.

2. TAG-CONDITIONED SED

As in the previous edition our SED model uses tag-conditioning [1], which means we also input the predicted tags from the FBCRNN in addition to the audio input features. While in the last edition we only used a tag-conditioned CNN, we now also train a tag-conditioned bidirectional CRNN and tag-conditioned bidirectional CTNN.

Here, we use similar architectures as in the FBCRNN with, however, only one classifier back-end. For the pure CNN the CNN1d and RNN Blocks are removed. In the bidirectional CRNN, a bidirectional RNN instead of unidirectional RNNs as in the FBCRNN. For the CTNN a Transformer Encoder [2] is used instead of an RNN, where we use 3 Transformer blocks each with 10 heads and 32-dimensional embeddings in each head. Also an positional encoding is added at the Transformer input.

Tag-conditioning is performed by concatenating a 10-dimensional multi-hot encoding of the tags with the inputs of the CNN2d, CNN1d, RNN/Tranformer, and FCN Blocks. For the CNN1d, RNN and FCN the encoding is concatenated along channel/feature dimension at each frame. For the CNN2d the encoding is concatenated along channel dimension at each time-frequency bin.

The models are trained with standard strong label BCE loss. For each set of the 2 strong pseudo label sets we train each of the model architectures giving us 3 models for each of the 2 strong pseudo label sets. For each of the strong pseudo label sets, we perform one iteration of self-training, i.e., generating new strong pseudo labels using the 3 models of that particular set followed by retraining the 3 architectures. Finally, we combine all the models from the two sets of pseudo labels into our final ensemble, i.e., 6 models in total.

3. IMPLEMENTATION DETAILS

3.1. Data Preparation/Augmentation

Initially, waveforms are resampled to $16\,\mathrm{kHz}$ and normalized $x(t) = s(t)/\max(|s(t)|)$ to be within the range of -1 and 1. As our systems input we then extract a M=128-dimensional log-mel spectrogram using a short-time Fourier transform (STFT) with frame-length of $60\,\mathrm{ms}$ and hop-size of $20\,\mathrm{ms}$. Each mel-bin is globally normalized to zero mean and unit variance.

At training time we perform various data augmentations, which is similar to what we already used previously [3, 1] and is described in the following.

Scaling: We randomly scale the waveform with a scale weight sampled out of a Log Truncated Standard Normal distribution with truncation at $\log(3)$.

Shifted superposition: We randomly superpose two audios as $x'_i(t) = x_i(t) + x_j(t-\tau)$ with a random shift τ sampled uniformly such that the superposed signal is not longer than $15\,\mathrm{s}$, i.e., if we, e.g., superpose 2 signals each having a length of $10\,\mathrm{s}$, the shift is uniformly sampled between -5 s and $5\,\mathrm{s}$. Labels are superposed accordingly and clipped at 1 to retain binary labels. We apply superposition with a probability of $2/3$. Due to the similarity to mixup [4], we previously referred to this augmentation also as mixup. However, as we do not interpolate the signals, calling it superposition is more accurate.

Frequency warping: We randomly warp the center frequencies of the mel filter bank similar to vocal tract length perturbation (VTLP) [5]. The boundary frequency is sampled from a Truncated Exponential distribution with $\sigma = M/2$ and truncation at $5 \cdot M$.

Table 1: Results of submitted SED models on validation set.

Metric	Baseline	Single	Ensemble
collar-based F1	0.401	0.591	0.602
PSD1	0.342	0.429	0.454
PSD2	0.527	0.748	0.758

The warping factor is sampled from a Log Truncated Normal distribution with $\mu = 0$, $\sigma = 0.8$ and truncation at $\log(1.3) \approx 0.26$. Note that the boundary frequency can fall above M, in which case the whole spectrogram is stretched or squeezed and filled with zeros.

Frequency-/Time-Masking: As in SpecAugment [6], we apply 1 time- and 1 frequency mask for each input with random locations and widths. The locations are uniformly sampled along the time- and frequency axes, respectively. Widths are uniformly sampled between 0 and $\min(1.4\,\mathrm{s}, 0.2T)$ for the time mask, where T is the length of the audio, or between 0 and 20 bins for the frequency mask.

Gaussian Noise: We add Gaussian noise to the final feature map with its standard deviation being uniformly sampled between 0 and 0.2.

3.2. Training

Training is performed for $40\,\mathrm{k}$ update steps with a batch size of 16. To balance the different data sets we repeat certain data sets in one epoch multiple time. Here, one epoch consists of 20 times the weakly labeled data, 2 times pseudo labeled unlabeled data (if used), 1 time synthetic data from this edition (synthetic21) and 2 times synthetic data from last edition (synthetic20). This sums up to $\approx 31\,\mathrm{k} + 28\,\mathrm{k} + 10\,\mathrm{k} + 5\,\mathrm{k}$ audio clips in one epoch. We further ensure that each batch includes at least 6 clips from the weakly labeled data, 2 clips from synthetic21 and 1 clip from synthetic20 as well as a each event class at least 1 time. We employ Adam [7] for optimization with a learning rate of $5 \cdot 10^{-4}$, with a ramp up during the first $1\,\mathrm{k}$ update steps and a reduction to 10^{-4} after $20\,\mathrm{k}$ update steps. We perform validation every $1\,\mathrm{k}$ update steps and choose the checkpoint with best validation performance in terms of (frame-based) F1-score as the final model.

4. RESULTS

In Table 1 we report the results of our submitted systems which are our final ensemble consisting of 8 FBCRNNs for audio tagging followed by 6 tag-conditioned SED models and a single model system with only 1 FBCRNN for audio tagging followed by 1 SED model. For the single model system we have chosen the models that achieved best tagging / SED performance on the validation set, which has been one of the bidirectional CRNNs for the SED. We performed a class- and metric-specific tuning of median filter lengths for post processing yielding two submissions for each of the systems. For collar-based F1-score we also tuned the decision threshold to give best performance on the validation set.

The $5\,\mathrm{min}$ long audio files from the evaluation set have been processed in chunks of $10\,\mathrm{s}$ with an overlap of $2\,\mathrm{s}$. Predictions have then been concatenated afterwards, where the first and last second of a chunk have been dropped in case of an overlap.

Only two days after challenge submission deadline we recognized, that significantly higher performance can be achieved for scenario 2 when using FBCRNN-based SED with larger frame context lengths and median filter lengths. Table 2 shows FBCRNN-based performance with class- and metric-specific frame context lengths and median filter lengths up to $4\,\mathrm{s}$.

Table 2: Results of FBCRNN-based SED on validation set.

Metric	Single	Ensemble
collar-based F1	0.511	0.526
PSD1	0.382	0.404
PSD2	0.793	0.812

5. REFERENCES

[1] J. Ebbers and R. Haeb-Umbach, "Forward-backward convolutional recurrent neural networks and tag-conditioned convolutional neural networks for weakly labeled semi-supervised sound event detection," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*, Tokyo, Japan, November 2020, pp. 41–45.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[3] J. Ebbers and R. Hb-Umbach, "Convolutional recurrent neural network and data augmentation for audio tagging with noisy labels and minimal supervision," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, New York University, NY, USA, October 2019, pp. 64–68.

[4] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[5] N. Jaitly and G. E. Hinton, "Vocal tract length perturbation (vtlp) improves speech recognition," in *Proceedings of ICML Workshop on Deep Learning for Audio, Speech and Language*, vol. 117, 2013.

[6] 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," *arXiv preprint arXiv:1904.08779*, 2019.

[7] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.