PRE-TRAINING AND SELF-TRAINING FOR 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) 2022 Challenge Task 4: Sound Event Detection in Domestic Environments*[1]. As in previous editions of the Challenge, we use forward-backward convolutional recurrent neural networks (FBCRNNs) [1, 2] for weakly labeled and semi-supervised sound event detection (SED) and eventually generate strong pseudo labels for weakly labeled and unlabeled data. Then, (tag-conditioned) bidirectional CRNNs (BiCRNNs) [1, 2] are trained in a strongly supervised manner as our final SED models. In each of the training stages we use multiple iterations of self-training. Compared to previous editions, we improved our system performance by 1) some tweaks regarding data augmentation, pseudo labeling and inference 2) using weakly labeled AudioSet data [3] for pretraining larger networks and 3) augmenting the DESED data [4] with strongly labeled AudioSet data [5] for finetuning of the networks. Source code is publicly available at `https://github.com/fgnt/pb_sed`.

***Index Terms*—** pre-training, self-training, sound event detection, polyphonic sound detection

1. MODELS

As in [2] the input features to all models are 128-dimensional log-mel energy features. For feature extraction we resample audio clips to 16 kHz and compute the short-time Fourier transform (STFT) using a window size and hop size of 960 and 320 samples, respectively, yielding a frame rate of 50 Hz.

1.1. Forward-Backward CRNN

The FBCRNN [1, 2], which is illustrated in Fig. 1, has been developed for the training with weak labels. It consists of a shared CNN front-end and separate forward and backward classifiers. Note, that differently to bidirectional RNNs the forward and backward RNNs do not exchange hidden representations here.

The core idea is, that at each point in time a tagged sound is either active somewhere before that point in time, or somewhere after that point in time or both. Therefore, the objective of the FBCRNN is that at each input frame a tagged sound must either be recognized by the forward classifier, which has processed all previous frames plus the current frame, or by the backward classifier, which has processed all subsequent frames plus the current frame:

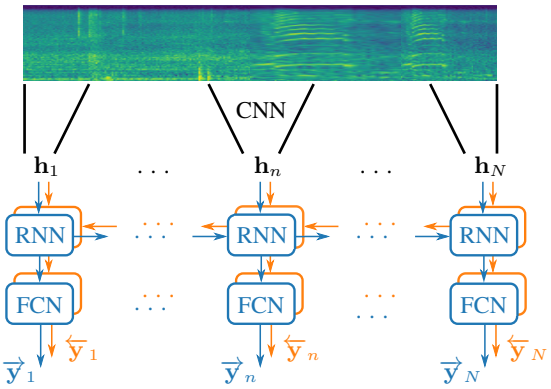

Figure 1: FBCRNN

$$y_{n,k}^{(\text{weak})} = \max\left(\overrightarrow{y}_{n,k}, \overleftarrow{y}_{n,k}\right), \tag{1}$$

$$L_{n,k} = -z_k^{(\text{weak})} \log(y_{n,k}^{(\text{weak})}) - (1 - z_k^{(\text{weak})}) \log(1 - y_{n,k}^{(\text{weak})}). \tag{2}$$

This way, the classifiers are encouraged to output high classification scores for active events as soon as possible and before having processed the whole audio clip. This enables the FBCRNN to be applied to much shorter segments at test-time than used for training. If we do have strong labels, however, we can derive desired outputs $\overrightarrow{z}_{n,k}$ and $\overleftarrow{z}_{n,k}$ and output scores are directly compared with the desired outputs using binary cross entropy (BCE):

$$\begin{aligned} L_{n,k} = &-(\overrightarrow{z}_{n,k} \log(\overrightarrow{y}_{n,k}) + (1 - \overrightarrow{z}_{n,k}) \log(1 - \overrightarrow{y}_{n,k}))/2 \\ &-(\overleftarrow{z}_{n,k} \log(\overleftarrow{y}_{n,k}) + (1 - \overleftarrow{z}_{n,k}) \log(1 - \overleftarrow{y}_{n,k}))/2\,. \end{aligned} \tag{3}$$

At test time a tag prediction is obtained as $\overline{y}_k = \overrightarrow{y}_{N,k}/2 + \overleftarrow{y}_{1,k}/2$. Also, SED can be achieved by performing tagging in small windows around each frame as illustrated in Fig. 2. For more details please refer to [1, 2].

1.2. Bidirectional CRNN

In a second training stage, and after having generated strong pseudo labels for weakly labeled and unlabeled data, we train BiCRNNs, i.e., where forward and backward RNNs do exchange hidden representations and where we only have a single fully connected classifi-

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

[1] `https://dcase.community/challenge2022/task-sound-event-detection-in-domestic-environments`

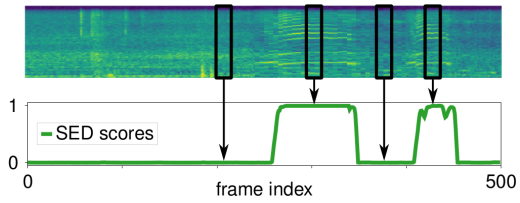

Figure 2: FBCRNN-based SED as tagging in small windows.

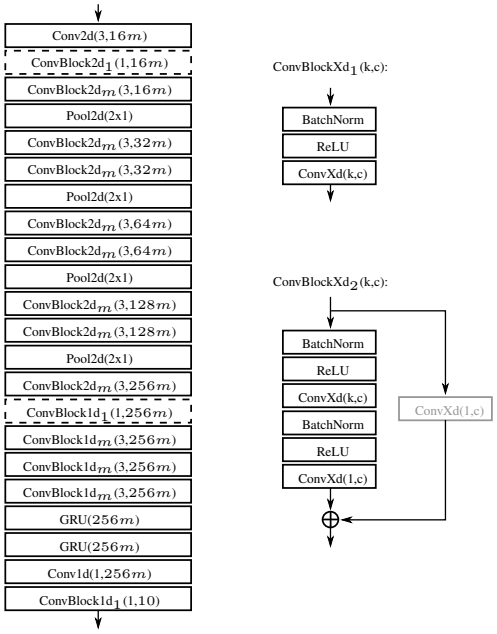

Figure 3: Neural Network Architectures with $m \in \{1, 2\}$. Dashed blocks are only used when $m = 2$. ConvXd(k,c) represent X-dimensional convolutional layers with kernels size k, stride 1 and c output channels. For 2-dimensional convolutions the kernel is always quadratic ($k \times k$). The grey convolution is only performed if output channels do not match input channels. The number of units in the GRU blocks refer to the number of units in each of the forward and backward GRUs for both FBCRNNs and BiCRNNs.

cation network outputting strong label predictions. Here, a standard strong label BCE loss is used for training.

In previous editions of the Challenge we exclusively used tag-conditioned networks [1, 2], i.e., where a multi-hot tag prediction vector $\hat{\mathbf{z}}^{(\text{weak})}$ is concatenated to the inputs of the convolutional and recurrent networks with $\hat{z}_k^{(\text{weak})} = [\overline{y}_k > \alpha_k]$ being one if $\overline{y}_k > \alpha_k$ else zero. While tag-conditioning has shown in [1] to improve performance in terms of collar-based F_1-score [6], we recently found that it harms performance in terms of PSDS [7]. An intuitive explanation for this is the following. The poly-phonic sound event detection score (PSDS), which is the area under the PSD-ROC curve, evaluates performance over various operating points whereas the thresholds α_k for tag prediction are tuned towards a single operating point (F_1-score in our case). Therefore, it should be harder to put the system in a different operating mode when being conditioned on tag predictions from another operating mode.

Therefore, we use tag-conditioned BiCRNNs for pseudo labeling within the self-training procedure (and when targeting collar-based F_1-score performance), but finally we train unconditioned BiCRNNs when targeting PSDS performance.

1.3. Neural Network Architectures

Our employed neural network architecture is shown in Fig. 3. We consider the two settings $m \in \{1, 2\}$ here. While $m = 1$ refers to the model architecture used in our previous work [2], $m = 2$ gives a deeper and wider architecture, which, however, does not affect the receptive field of the network. We set $m = 2$ when using external data for model training whereas we set $m = 1$ without external data usage.

2. TRAINING PROCEDURE

2.1. Data Preparation/Augmentation

In all trainings we use the same data preparation/augmentation as described in [2] with the following adjustments. Inspired by SpecAugment [8], we add a time warping augmentation, where, for each audio clip, we sample a random anchor from a uniform distribution $\mathcal{U}(0.4N, 0.6N)$, where N is the number of frames in the clip, and warp the anchor left or right by a random shift sampled from a uniform distribution $\mathcal{U}(-0.1N, 0.1N)$ without changing the length of the audio clip. The time warping is implemented by using different STFT hop sizes left and right to the anchor.

Further, we perform an unshifted superposition instead of a shifted superposition here, which keeps the lengths of the superposed audio clips similar to the lengths of the original clips.

2.2. Pre-Training

When using external training data, we pre-train an FBCRNN model on the whole AudioSet with $\approx$1.8 M audio clips that we were able to download. Training is performed for 500 k update steps using a batch size of 32 and Adam optimizer [9] with a learning rate of $5 \cdot 10^{-4}$. During training we perform a balancing of the 527 event classes by repeating clips with rare event classes so that there are for each event class at least 10 k clips per epoch. Our final checkpoint achieves a mean average precision of 42.66% on AudioSet's eval set.

2.3. Self-Training

Our self-training procedure is illustrated in Fig. 4. It is mostly the same as in [2] with minor changes in the number of training steps, larger ensemble sizes and different strong pseudo labeling hyper parameters. In all trainings Adam [9] is used for optimization with a learning rate of $5 \cdot 10^{-4}$, if not stated otherwise, and checkpoints are saved every 1000 update steps.

2.3.1. With External Data

Here, we combine data from the DESED dataset [4] and from AudioSet [5]. As before, we repeat certain subsets multiple times within a training epoch to balance the training data. The training data is composed of 10 times the weakly labeled real data (10·1578 examples), 10 times the strongly labeled external real data from the baseline system (10·3470 examples), 2 times pseudo labeled unlabeled real data if used (2·14412) and 2 and 1 times strongly labeled synthetic data from the 2020 (2·2576) and 2021 (1·10000) editions of the Challenge, respectively.

FBCRNNs are initialized with the pre-trained FBCRNN model up to the output layer, whereas for BiCRNNs only the convolutional layers are initialized with those from the pre-trained FBCRNN

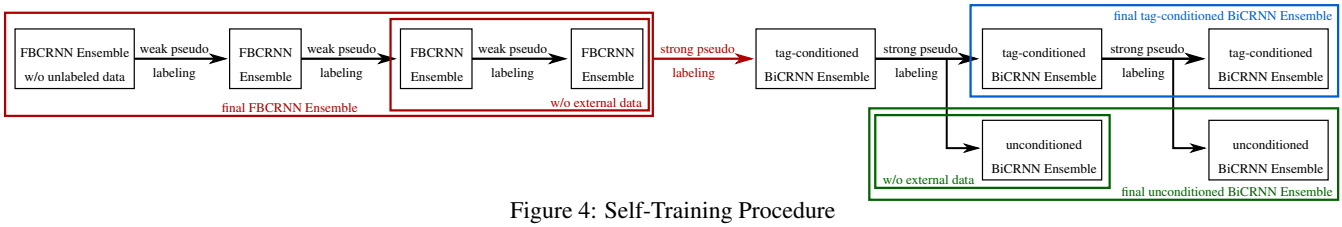

Figure 4: Self-Training Procedure

model. In all trainings all the Conv2d layers and the first Conv1d layer are frozen. The rest of the network is fine-tuned/trained for 20 k update steps if not stated otherwise. Note, that for tag-conditioned BiCRNNs, the tag predictions are only concatenated to the RNN input, as the CNN is to be initialized with the pre-trained weights.

Initially, an ensemble of 10 FBCRNNs is fine-tuned without using the unlabeled data for 10 k update steps. For each FBCRNN we choose the checkpoint that achieved the best tagging F_1-score on the validation set and combine them in an ensemble where the ensemble output scores are given as an average over the individual output scores. The ensemble is used to generate weak pseudo labels for all the examples in the unlabeled data set. Further, outer event boundaries are detected for some of the tags in the weakly and pseudo labeled unlabeled data as described in [2]. We now perform 3 iterations of self-training including the pseudo labeled data. In each iteration 10 new FBCRNNs are fine-tuned, with the resulting ensemble being used to rerun the pseudo labeling.

However, the performance of the FBCRNN ensembles do not improve a lot over the self-training iterations anymore when using the pre-trained weight initialization. Therefore, we simply combine all the FBCRNNs from the 4 training iterations into a large final FBCRNN ensemble consisting of 40 sub-models. The final ensemble is used for SED, as illustrated in Fig. 2, to strongly pseudo label the weakly and unlabeled data. Here, event-specific window sizes, median filter lengths for post-processing and decision thresholds are tuned to give best collar-based F_1-score on the validation set (rather than best frame-based F_1-score as in previous editions). Further, the SED scores are masked according to the (predicted) tags here, i.e., SED scores for a certain event class can only be >0 if the event class has also been tagged in the audio clip, which we refer to as tag-masking in the following.

Next, we perform 3 iterations of strongly labeled self-training of tag-conditioned BiCRNN ensembles, each consisting of 10 BiCRNNs. Here, those training checkpoints are chosen as final models which achieve highest frame-based F_1-score on the validation set when not using median filtering as post-processing. For the final ensemble, event-specific median filter lengths and decision thresholds are tuned to give best collar-based F_1-score on the validation set. Each iteration ends with a strong pseudo re-labeling of the weakly and unlabeled data.

While validation performance significantly improves after the first strong pseudo re-labeling, it does not further improve afterwards. Therefore, we train 10 unconditioned BiCRNNs using the pseudo labels provided by the first tag-conditioned BiCRNN ensemble and another 10 using the pseudo labels provided by the second tag-conditioned BiCRNN ensemble, which are combined into our final unconditioned BiCRNN ensemble consisting of 20 BiCRNNs. Similarly, we combine the second tag-conditioned BiCRNN ensemble (which as been trained using the pseudo labels provided by the first tag-conditioned BiCRNN ensemble) and the third tag-conditioned BiCRNN ensemble into our final tag-conditioned BiCRNN ensemble.

2.3.2. *Without External Data*

When not using the external data, models are not initialized with the pre-trained weights and we use 20 times the weakly labeled real data in one epoch and no strongly labeled external real data. Further, a learning rate rampup to $5 \cdot 10^{-4}$ over the first 1000 update steps is used. Training of the initial FBCRNN is performed for 30 k update steps, where the learning rate is reduced to 10^{-4} after 20 k update steps. Following trainings including pseudo labeled data run for 45 k update steps, where the learning rate is reduced to 10^{-4} after 30 k update steps.

Here, without the pre-training, the performance increases over the self-training iterations and we only combine the 20 FBCRNNs from self-training iterations 2 and 3 into our final FBCRNN ensemble. Also we have not performed a third iteration of strongly labeled self-training here, but simply use the 10 unconditioned BiCRNNs trained using the pseudo labels provided by the first tag-conditioned BiCRNN ensemble as our final unconditioned BiCRNN ensemble.

3. RESULTS

We report results on the validation portion of the DESED dataset [4] in terms of

- approx. PSDS1/PSDS2: PSDSs [7] computed with the psds_eval[2] package using 50 linearly spaced thresholds with two different sets of metric parameters which are the official metrics used in the challenge[1].
- true PSDS1/PSDS2: In [10] it has been shown that the PSDS approximation using a limited set of thresholds may significantly underestimate the true PSDS which is obtained when taking all possible thresholds into account. Therefore, we here also report the true PSDSs, which we compute using the sed_scores_eval[3] package, as a reference.
- $F_1^{(\text{collar})}$: macro-average collar-based F_1-score [6] with a 200 ms collar on onsets and a 200 ms / 20% of the event length collar on offsets.

We report results for the following systems that we submitted to the Challenge:

- Ensemble1: final unconditioned BiCRNN ensemble (40 FBCRNNs for tagging plus 20 BiCRNNs for SED) with hyper-parameters tuned towards good PSDS1 performance on validation set.

[2] https://github.com/audioanalytic/psds_eval
[3] https://github.com/fgnt/sed_scores_eval

Table 1: Submitted systems' performances on validation set. The column "ext" indicates whether external data has been used. Bold values indicate best performance.

		PSDS1		PSDS2		
System	ext	approx	true	approx	true	$F_1^{(collar)}$
Baseline1		.336		.536		40.1%
Baseline2	✓	.351		.552		42.9%
Ensemble1	✓	**.512**	**.521**	.772	.817	65.4%
Ensemble2	✓	.093	.080	**.868**	**.877**	9.1%
Ensemble3	✓	.483	.498	.713	.778	**68.0%**
Ensemble4		.492	.504	.721	.765	60.5%

- Ensemble2: final FBCRNN ensemble (40 FBCRNNs) performing tagging in segments with a segment length of 5 s. This model aims at high PSDS2 performance. No further hyper-parameters need to be tuned.
- Ensemble3: final tag-conditioned BiCRNN ensemble (40 FBCRNNs for tagging plus 20 BiCRNNs for SED) with hyper-parameters tuned towards good collar-based F_1-score performance on validation set.
- Ensemble4: final unconditioned BiCRNN ensemble without external data usage (20 FBCRNNs for tagging plus 10 BiCRNNs for SED) with hyper-parameters tuned towards good PSDS1 performance on validation set.

The following hyper-parameters are tuned individually for each event class

- Median filter length
- Decision threshold for collar-based F_1-score evaluation
- Whether to use tag-masking or not

Note, that in the previous edition our model tended to output either very high or low scores which is why we tuned a non-linear score transformation on the validation set to bring the approximate PSDSs closer to the true PSDSs. For the current models, however, such score transformation did not bring the targeted approximate PSDS significantly closer to the true PSDS, which is why we do not use a score transformation in the submitted systems. Also note, that previously we used tag-masking for all events which turned out to significantly degrade PSDS performance (probably for the same reason why tag-conditioning degrades PSDS performance as discussed above). Therefore, we now only use tag-masking for those event classes for which it increases validation performance.

Results are shown in Table 1. It can be seen that our systems significantly outperform the baselines w.r.t. all metrics.

Due to the limitation in the number of systems that can be submitted, we only submitted large ensembles to achieve best performance. Here, however, we also report in Table 2 the validation performance, when using at most a single FBCRNN and a single BiCRNN. For this we choose those models from the ensembles which perform best according to the corresponding checkpoint selection criterion (for FBCRNN: best tagging F_1-score on validation set, for BiCRNN: best frame-based F_1-score on validation set without post-processing) and refer to the systems as Single1-Single4. It can be seen that the single model system performances do not lie far behind the ensemble performances, while they only require a small fraction of the memory and computational resources at test-time.

Table 2: Non-submitted single model systems' performances on validation set. The column "ext" indicates whether external data has been used. Bold values indicate best performance.

		PSDS1		PSDS2		
System	ext	approx	true	approx	true	$F_1^{(collar)}$
Baseline1		.336		.536		40.1%
Baseline2	✓	.351		.552		42.9%
Single1	✓	**.505**	**.515**	.757	.807	57.3%
Single2	✓	.077	.089	**.858**	**.866**	9.1%
Single3	✓	.466	.481	.685	.769	**66.8%**
Single4		.481	.497	.703	.769	58.2%

4. 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 *Proc. Workshop on Detection and Classification of Acoustic Scenes and Events*, Tokyo, Japan, November 2020, pp. 41–45.

[2] ——, "Self-trained audio tagging and sound event detection in domestic environments," in *Proc. Workshop on Detection and Classification of Acoustic Scenes and Events*, Barcelona, Spain, November 2021, pp. 226–230.

[3] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2017, pp. 776–780.

[4] N. Turpault, R. Serizel, A. P. Shah, and J. Salamon, "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis," in *Proc. Workshop on Detection and Classification of Acoustic Scenes and Events*, 2019.

[5] S. Hershey, D. P. Ellis, E. Fonseca, A. Jansen, C. Liu, R. C. Moore, and M. Plakal, "The benefit of temporally-strong labels in audio event classification," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2021, pp. 366–370.

[6] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, p. 162, 2016.

[7] Ç. Bilen, G. Ferroni, F. Tuveri, J. Azcarreta, and S. Krstulović, "A framework for the robust evaluation of sound event detection," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2020, pp. 61–65.

[8] 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," in *Proc. Interspeech*, 2019, pp. 2613–2617.

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

[10] J. Ebbers, R. Haeb-Umbach, and R. Serizel, "Threshold independent evaluation of sound event detection scores," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2022, pp. 1021–1025.