TASK AWARE SOUND EVENT DETECTION BASED ON SEMI-SUPERVISED CRNN WITH SKIP CONNECTIONS: DCASE 2021 CHALLENGE, TASK 4

Technical Report

Mohammed Hafsati

Tuito
943 Voie Antiope
France, 13600, La Ciotat
mohammed.hafsati@tuito.fr

Kamil Bentounes

Tuito
943 Voie Antiope
France, 13600, La Ciotat
kamil.bentounes@tuito.fr

ABSTRACT

Sound Event Detection (SED) is the task of classifying different sounds occurring in a recorded environment and their onset and offset times. This assignment is the primary goal of the fourth task of the DCASE challenge using some strongly labeled, partially labeled, and unlabeled datasets. In this paper, we describe our submitted approach for this challenge. Our neural network is based on sequential convolutional neural networks with skipping some layers and a recurrent neural network. To overcome the challenge of using unlabeled data, we used semi-supervised learning, and to improve the performance further, we propose to use data augmentation techniques. With our model, we can slightly outperform the baseline with fewer filters and therefore fewer parameters. Moreover, similar amount of parameters as the baseline, we significantly outperform it.

Index Terms— SED, Semi-supervised learning, CRNN, Skip connections, Data augmentation, DCASE 2021.

1. INTRODUCTION

The fourth task of the DCASE challenge consists of providing a framework that can classify ten different sound events (cat, Alarm/ringing, Running water, vacuum cleaner, dishes, frying, dog, speech, electric shaver/toothbrush, and blender) and estimate their onset and offset times from an audio recording.[1] Such a task can be used in several applications such as speaker diarization, sound source separation, sound source localization, voice activity detection to improve ASR modules, *etc*.

The most challenging part of the assignment is that a large amount of the provided dataset is unlabeled, which means that the participants do not have any idea about the events or their onset and offset times. To overcome, this limitation several contributions exist, especially in the previous editions of the DCASE challenge such as in [1, 2, 3, 4]. The idea is to train a neural network using a semi-supervised approach such as the mean-teacher approach [5], which the baseline model adopts for the training process [1]. Our contribution to the challenge is also based on the mean-teacher approach. We changed the architecture of the model and used data augmentation to overthrow the lack of labeled data. We outperform the baseline with a smaller number of trainable parameters and, therefore, a shorter training time with our approach. In this technical report, we begin by describing the provided dataset and the encoding of the inputs and outputs of the model, followed by a brief description of the training process, we describe our model architecture and our data augmentation afterward, and finally, we present our results and discuss them.

2. DATA SET INPUTS AND OUTPUTS ENCODING

The DCASE 2021 data set is composed as follows:

- Weakly labeled training set.
- Unlabeled in domain training set.
- Synthetic set with strong annotations.

The weakly labeled training set contains 1578 clips, the Unlabeled in domain training set has 14412 clips, and finally, the Synthetic strongly labeled set contains 10000 generated clips for the training and 1500 for the validation. The audio clips are sampled at 44,100 Hz for the weakly labeled and labeled, 22,050 Hz for the strongly labeled, and with a maximum duration of 10 seconds. Each audio clip contains at least one sound corresponding to one of the ten possible classes.

All audio clips are resampled at 16,000 Hz and converted to mono-channel if it isn't the case. We afterward extract the log mel-spectrogram from the audio clips using an analysis window of 2048, a hop length of 365, and 128 as the number of mels. Thus, we have an input size of (628,128). We standardize the input by computing the mean over all the training data and standard deviation of each mel bin. The outputs for the training, validation, and test were encoded from the provided tsv files (the same way as the baseline) into a time map of $(\#temporal\_frames, classes)$, the number of temporal frames in our case was equal to 157, and the number of classes was equivalent to 10.

3. TRAINING PROCESS

As mentioned in the introduction, the most challenging part of this task is considering unlabeled data during the training. Indeed, supervised learning, in this case, can not be used since the output is not always provided. To overcome this problem, as adopted by several contributions in previous editions of the DCASE challenge [1, 3, **?**]

[1] http://dcase.community/challenge2021/task-sound-event-detection-and-separation-in-domestic-environmentsDcase website

semi-supervised learning can be considered a solution that can effectively exploit a large amount of unlabeled data.

As in [1], we adopt the mean teacher strategy [5]. This technique is based on training a student model, which its weights are used to update the mean teacher parameters as an exponential moving average of the student weights. This helps to produce a more accurate model than using the final weights directly. The backpropagation involves computing two types of loss functions classification cost and consistency loss. For more information about the training process, please refer to the baseline.

4. THE PROPOSED MODEL

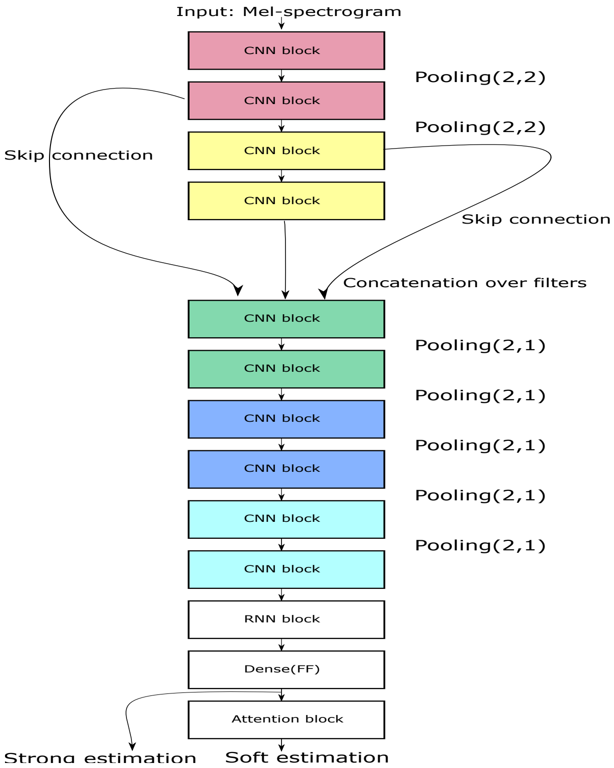

Figure 1: The proposed model to solve the SED problem. Same color represents the same number of filters

As shown in Fig. 1, our model consists of ten consecutive convolutional neural network (CNN) blocks, followed by an RNN block, which its output is fed to a dense layer activated with sigmoid function. A CNN block consists of a convolutional layer followed by batch normalization and a Relu activation. The RNN block consists of two GLU layers. We added some skip connections to prevent the vanishing gradient problem, which seems beneficial to our model. These skip connections connect the third, the fourth, and the sixth CNN blocks by concatenating the filters. This technique allows also the model to take in consideration some missing information from previous the CNN block. The size of our inputs and outputs for each block is given in Tab. 1.

We concluded some dropout layers between each block besides between the RNN blocks, which were activated during the training. We decided a drop out of 20% between the CNN blocks and 50% otherwise.

	Model with 8 filters		Model with 16 filters	
Block	Input size	# of filters	Input size	# of filters
Cnn1	(bs,1,628,128)	8	(bs,1,628,128)	16
Cnn2	(bs,8,314,64)	8	(bs,16,314,64)	16
Cnn3	(bs,8,157,32)	16	(bs,16,157,32)	32
Cnn4	(bs,16,157,32)	16	(bs,32,157,32)	32
Cnn5	(bs,16,157,32)	32	(bs,32,157,32)	64
Cnn6	(bs,64,157,16)	32	(bs,128,157,16)	64
Cnn7	(bs,32,628,8)	64	(bs,64,628,8)	128
Cnn8	(bs,64,628,4)	64	(bs,128,628,4)	128
Cnn9	(bs,128,628,2)	128	(bs,128,628,2)	128
Cnn10	(bs,128,628,1)	128	(bs,128,628,1)	128
GRU (2 layers)	(bs,128,157)	NULL	(bs,128,157)	NULL
Dense layer	(bs,128,157)	NULL	(bs,128,157)	NULL

Table 1: Our model's parameters

5. DATA AUGMENTATION

We chose to augment the strongly labeled and weakly labeled data with three different techniques, and they are given as follows:

- Mixing the provided audio.
- Shifting the frequency.
- Changing the magnitude.

Mixing the audios will help to have new inputs with different events which can overlap each other. This should help the training to better generalize the classification and time events prediction. From each audio, we created three (x3) new inputs (mixtures) by simply adding the treated audio to a randomly chosen sound from the same folder which doesn't have the same events.

Shifting the frequency can be beneficial to events such as vacuum cleaner, electric shaver, and toothbrushes, *etc*. Indeed, objects from the same listed classes can produce similar sounds but with different frequencies, and shifting the frequency can simulate this effect. For each audio, we applied two (x2) different pitch factors randomly picked between [0.5, 1] and [-1, -0.5], respectively.

Changing the magnitude can help our model to become invariant to the overall volume of the input audio. For each audio, we generated two inputs (x2), one with a gain randomly picked between [10dB, 20dB] and the other randomly picked between [-20dB, -10dB].

We finally multiplied the number of inputs for the strong and weak labeled data by a factor of 7.

6. EVALUATION

To evaluate our trained models and compare them to other contributions, we use as recommended the following metrics:

- F1-score [6].
- Polyphonic Sound Detection Score (PSDS) [7].

Three evaluation data sets were provided:

- Validation.
- Public evaluation.
- Official evaluation. The results corresponding to the evaluation on this data set will be announced by the challenge organizers on the official website.

We inferred several versions of our proposed neural network on these three evaluation data sets. The results are reported for the two first data sets in Sec. 6.1, Tab. 2 and Tab. 3, and the predictions on the third data set were sent to be evaluated by the organizers of the challenge.

First, we trained our neural network starting with eight filters for the first CNN blocks. Then, we trained the same model but starting this time with 16 filters (more informations about the number filters are given in Tab. 1). Finally, we applied data augmentation techniques mentioned in Sec. 5 starting with eight filters and using a batch size of 54 ([18, 12, 24] for labeled, weakly labeled, and unlabelled dataset, respectively). We finally discussed these results in Sec. 6.2.

6.1. Results

In Tab. 2 and Tab. 3, we report the scores to compare the performances between the baseline and our proposed model trained in different ways. The higher the scores are, the better the model is. For more information about the F1-score, and the PSDS score please refer to [6] and [7], respectively.

6.2. Discussion

We wanted first to reproduce the Baseline results reported on the challenge Website.[2] For this, we trained the given model on the provided data set. However, we weren't able to get precisely the same baseline results. In the meantime, we trained our neural network on the same database.

We can see that our neural network outperforms the SED baseline results in terms of all evaluation metrics on the two first evaluation data sets.

Data-augmentation techniques allow us to gain some precision. Indeed, augmenting the labeled and weakly labeled data helps the model to be more accurate. However, since the evaluation is done on real recorded data, and the augmented labeled part of the trained data is synthetic, the model became more efficient on synthetic data, and therefore struggles on real recordings.

7. CONCLUSION

This paper proposed a sound event detector based on a CRNN architecture with some skip connections, trained with the mean teacher semi-supervised technique. Our model significantly outperformed the baseline in terms of all scores on both scenarios. These results can be improved in several combinations: augmenting the real data, using more filters, and changing the batch size. We can also train for a first time the model, inferred it into the unlabelled subset, then train it for a second time with new real data generated from the first trained model predictions on real unlabelled data. The time context processing part can also be improved using another recurrent neural network instead of using the baseline RNN (GRU). The encoding part can be replaced by others architectures (transformers, ResNet, *etc*).

	PSDS scenario 1	PSDS scenario 2	Collar-based F1	Intersection-based F1
Baseline	33.5	52.7	40.0	64.4
Proposed model: start filters = 8	32.7	52.8	40.9	63.2
Proposed model: start filters = 16	**34.5**	55.5	**41.3**	65.1
Proposed model with augmented data (start filters = 8)	32.5	**56.1**	40.0	**66.2**

Table 2: Results on the validation data set.

	PSDS scenario 1	PSDS scenario 2	Collar-based F1	Intersection-based F1
Baseline	36.3	58.1	42.4	67.2
Proposed model: start filters = 8	32.6	55.8	43.0	65.3
Proposed model: start filters = 16	**36.9**	**59.4**	**44.8**	**70.9**
Proposed model with augmented data (start filters = 8)	34.2	56.7	41.5	65.6

Table 3: Results on the public evaluation data set.

8. REFERENCES

[1] L. Delphin-Poulat and C. Plapous, "Mean teacher with data augmentation for dcase 2019 task 4," *Orange Labs Lannion, France, Tech. Rep*, 2019.

[2] K. Miyazaki, T. Komatsu, T. Hayashi, S. Watanabe, T. Toda, and K. Takeda, "Convolution-augmented transformer for semi-supervised sound event detection," DCASE2020 Challenge, Tech. Rep., June 2020.

[3] S. Cornell, M. Olvera, M. Pariente, G. Pepe, E. Principi, L. Gabrielli, and S. Squartini, "Task-aware separation for the dcase 2020 task 4 sound event detection and separation challenge," in *DCASE 2020-5th Workshop on Detection and Classification of Acoustic Scenes and Events*, 2020.

[4] N. K. Kim and H. K. Kim, "Polyphonic sound event detection based on convolutional recurrent neural networks with semi-supervised loss function for dcase challenge 2020 task 4," *arXiv preprint arXiv:2007.00947*, 2020.

[5] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," *arXiv preprint arXiv:1703.01780*, 2017.

[6] C. Goutte and E. Gaussier, "A probabilistic interpretation of precision, recall and f-score, with implication for evaluation," in *European conference on information retrieval*. Springer, 2005, pp. 345–359.

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

[2] http://dcase.community/challenge2021/task-sound-event-detection-and-separation-in-domestic-environmentsDcase website