FEW-SHOT BIOACOUSTIC EVENT DETECTION WITH PROTOTYPICAL NETWORKS, KNOWLEDGE DISTILLATION AND ATTENTION TRANSFER LOSS

Technical Report

Radoslaw Bielecki

Warsaw, Poland
r.bielecki@samsung.com

ABSTRACT

The report presents the results of submission to Task 5 (Few-shot Bioacoustics Event Detection) of Detection and Classification of Acoustic Scenes and Events Challenge (DCASE) 2021. This task focuses on sound event detection in a few-shot learning setting for animal (mammal and bird) vocalizations. Main issue of this task is the very limited number of training instances. The presented approach is based on prototypical networks built up from the convolutional layers. Main techniques used during model development are knowledge distillation, attention transfer loss and spectrogram augmentation. The best of presented models achieved 55.5% if the F-measure on the challenge validation set. That is improvement by over 10% in comparison to baseline model.

Index Terms— acoustic event detection, convolutional neural networks, prototypical network, deep learning, few-shot learning

1. INTRODUCTION

Task 5 of Detection and Classification of Acoustic Scenes and Events Challenge (DCASE) 2021 deals with detecting bioacoustic events in the conditions of very limited samples of predicted class. Methods for the few-shot machine learning in audio domain are still actively researched. For this task, prototypical networks were chosen[1]. Additional training techniques like knowledge distillation[2, 3], data augmentation[4] and additional loss components[2, 3] significantly improved models performance. In addition, different combinations of size and amount of layers used in models gave a significant performance boost as well.

2. DATASETS AND PREPROCESSING

Task 5 development set consisted of 14.3 hours for training set and 10 hours for validation set, of audio with labeled classes instances. Despite the duration of development set, labels covered only small fraction of it. The sampling rate within development set varies between 8000 Hz to 44100 Hz. Training data labels consisted of 38 classes referring to different mammals and bird sounds. In addition to the training data provided by the organizers, the ESC50[5] dataset was used. ESC50 consisted of 4000 audio files grouped into 40 classes. Each ESC50 audio file was in 44100 Hz sampling rate. Total audio duration of ESC50 is about 5.5 hour. Important to notice is that ESC50 contained not only animal sounds like in development dataset but also natural soundscapes, human speech, domestic sounds and urban noises.

The data preprocessing consisted of the following steps. Initially, from each audio file was resampled to 22050 Hz upon loading and silence was removed. Afterwards the log-transformed melspectrograms were created and PCEN[6] (Per-Channel Energy Normalization) was applied to obtained melspectrograms. Parameters used during spectrogram creation are: window of 1024 with hop of 256 and 128 mel bins. Next, slicing into frames was applied to melspectrograms - each frame was 0.2 sec long and an overlap between adjacent frames was 0.15 sec - ultimately each frame was of shape (17, 128). After slicing was complete, frames corresponding to class instances were extracted on the basis of the provided onset-offset annotations. As for ESC50 files, whole files were considered a class instances. This methodology provided a positives set.

For validation data, first 5 events in each file were used for training a few-shot model, the rest was considered a query set. A negative set was created from all frames for a given audio file. This bases on the assumption that the target event is relatively rare. Similar approach was used in Y.Wang article[7].

All created models based on the data preprocessed in the same manner.

3. ARCHITECTURES AND TRAINING

In this section, more detailed information about architecture and training process used in the model development will be provided.

All models were created basing on convolutional neural network blocks. Each block consisted of 4 steps - Conv2D, BatchNorm2D, ReLU, MaxPool2D(2). If model consisted of more than 4 such blocks, 5th and following blocks did not contain MaxPool2D step. Also size of channels used during Conv2D was distinct between models.

During model training prototypical loss, knowledge distillation and attention transfer loss were combined. Prototypical loss (PL) calculation is adopted from baseline model of organisers and it is based on J.Snell article[1]. Knowledge distillation (KD) consisted of training several models in one run. Each subsequent model was taking into account differences between predictions of its own and its predecessor during loss calculation. Usage of attention transfer (AT) loss was inspired by an article of Y.Tian[2] and S.Zagoruyko[3]. With attention transfer efficiency of knowledge distillation from teacher to student improved most of the times.

For each complete training, 10 generations of models were created. Model created from first generation used only PL during loss calculation. Each subsequent generation combined PL, KD loss and AT loss in proportion:

$$Loss = 0.5 * PL + 0.25 * KD + 0.25 * AT \tag{1}$$

.

Training data is processed before the main training phase. Classes in training data are balanced through random oversampling. Next, both training and validation data are normalized by subtracting mean and diving by standard deviation. Both mean and standard deviation are based on balanced training data. Each batch consisted of 10 samples from 10 classes, 100 in total. During training phase mel-spectrograms were augmented by random time and frequencies masking methods[4].

Model	Layers	Conv2D channels	Best Generation
Bielecki_SMSNG_task5_1	5	250	3
Bielecki_SMSNG_task5_2	4	200	0
Bielecki_SMSNG_task5_3	5	150	3
Bielecki_SMSNG_task5_4	5	150	8

Table 1: Models architecture differences

Operation	Outputs	Kernel	Stride	Params
ConvBlock*	150	3	1	1800
ConvBlock*	150	3	1	202950
ConvBlock*	150	3	1	202950
ConvBlock*	150	3	1	202950
ConvBlock**	150	3	1	202950
Total				813600

ConvBlock*: Conv2D+BatchNorm2D+RelU+MaxPool2D(2)
ConvBlock**: Conv2D+BatchNorm2D+RelU

Table 2: Architecture of *Bielecki_SMSNG_task5_1*, *Bielecki_SMSNG_task5_3* and *Bielecki_SMSNG_task5_4 models*

Operation	Outputs	Kernel	Stride	Params
ConvBlock*	200	3	1	2400
ConvBlock*	200	3	1	360600
ConvBlock*	200	3	1	360600
ConvBlock*	200	3	1	360600
Total				1084200

ConvBlock*: Conv2D+BatchNorm2D+RelU+MaxPool2D(2)

Table 3: Architecture of *Bielecki_SMSNG_task5_2 model*

4. PREDICTIONS POST-PROCESSING

Post-processing on output predictions consisted of 2 steps. Firstly, all events which duration was shorter than 60% (for *Bielecki_SMSNG_task5_1* and *Bielecki_SMSNG_task5_2*) or 65% (for *Bielecki_SMSNG_task5_3* and *Bielecki_SMSNG_task5_4*) of the minimum duration of the shots for each prediction file.

Secondly, predictions elongation basing on the mean duration of shots from each prediction file - models predictions were elongated by 30% of the duration of that mean.

5. RESULTS

In Table 4 and Table 5 sizes and results for all models are presented.

Model	Params	Size
Bielecki_SMSNG_task5_1	813600	3196 KB
Bielecki_SMSNG_task5_2	1084200	4251 KB
Bielecki_SMSNG_task5_3	813600	3196 KB
Bielecki_SMSNG_task5_4	813600	3196 KB

Table 4: Models size summary

Model	Overall F-measure (in %)	Overall Precision (in %)	Overall Recall (in %)
Bielecki_SMSNG_task5_1	**52.505**	55.999	49.421
Bielecki_SMSNG_task5_2	51.829	**57.953**	46.875
Bielecki_SMSNG_task5_3	51.794	52.11	51.47
Bielecki_SMSNG_task5_4	51.143	54.326	48.311
Baseline	41.48	32.20	**58.27**

Table 5: Models results summary

6. CONCLUSION

In this report 4 models were presented to Task 5 of DCASE 2021. Architecture was based on convolutional neural networks with log-transformed melspectrograms and prototypical loss as few-shot technique. Main addition in comparison to baseline model was knowledge distillation with attention transfer. Also size and depth of the models had significant impact on the scores. Most of the times deeper models performed better. After including additional training data in form of ESC50 dataset the performance gap between shorter but bigger and deeper but smaller models narrowed. However, both model types benefited from more training data. Suprisingly, post-processing of predictions had a great impact on the final scores. Removal of short predictions was crucial and resulted in more than 20% F-measure improvement. On the other hand, elongation of remaining predictions improved the performance by 1.5% at best.

Finally, developed models showed an improvement in F-measure by 10-11% and in precision by 20-25%, but also a downgrade by 8-10% in recall in comparison to baseline model. These results indicate that prototypical networks used with the knowledge distillation technique are promising approach to few-shot learning audio tasks. The presented models can definitely be improved by including more training datasets beyond ESC50 and more diverse data augmentation strategies. Further refinement of the network architectures and postprocessing phase could also bring score improvement.

7. REFERENCES

[1] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," *CoRR*, vol. abs/1703.05175, 2017. [Online]. Available: http://arxiv.org/abs/1703.05175

[2] Y. Tian, Y. Wang, D. Krishnan, J. B. Tenenbaum, and P. Isola, "Rethinking few-shot image classification: a good embedding

is all you need?" *CoRR*, vol. abs/2003.11539, 2020. [Online]. Available: https://arxiv.org/abs/2003.11539

[3] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," in *ICLR*, 2017. [Online]. Available: https://arxiv.org/abs/1612.03928

[4] 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," *Interspeech 2019*, Sep 2019. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2680

[5] K. J. Piczak, "ESC: Dataset for Environmental Sound Classification," in *Proceedings of the 23rd Annual ACM Conference on Multimedia*. ACM Press, 2015, pp. 1015–1018. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2733373.2806390

[6] V. Lostanlen, J. Salamon, M. Cartwright, B. McFee, A. Farnsworth, S. Kelling, and J. P. Bello, "Per-channel energy normalization: Why and how," *IEEE Signal Processing Letters*, vol. 26, no. 1, pp. 39–43, 2019.

[7] Y. Wang, J. Salamon, M. Cartwright, N. J. Bryan, and J. P. Bello, "Few-shot drum transcription in polyphonic music," 2020.