SIAMESE NETWORK FOR FEW-SHOT BIOACUSTIC EVENT DETECTION

Technical Report

Bartłomiej Zgórzyński, Mateusz Matuszewski

Samsung R&D Institute Poland
Artificial Intelligence
Warsaw, Poland
b.zgorzynski@samsung.com

ABSTRACT

This report presents the system and results of a submission to Task 5 (Few-shot Bioacoustics Event Detection) of the Detection and Classification of Acoustic Scenes and Events Challenge 2022 (DCASE2022). This task focuses on sound event detection in a few-shot learning setting for animal (mammal and bird) vocalisations. To address this problem, we propose a deep-learning-based system for similarity estimation between two given audio samples. The presented approach is based on a siamese network with two identical convolutional feature extractors and one mutual fully connected head, which is responsible for similarity modeling. Due to the fact that sounds occurring in nature can vary significantly in duration, the proposed system consists of three separate models, each responsible for detecting events of a different length. We achieve an F-measure score of 67.3 % on DCASE2022 Task 5 validation set, compared to 29.6 % scored by the best baseline system.

Index Terms— sound event detection, few-shot learning, siamese network, deep learning

1. INTRODUCTION

This paper presents the most important technical details of the solutions submitted for DCASE 2022 Task 5: Few-shot Bioacoustic Event Detection [1]. A few-shot learning [2, 3, 4] system must be able to make predictions based only on a few instances of the target class. The main purpose of this challenge is to find reliable algorithms that are capable of dealing with data sparsity, class imbalance and noisy environments. Sound events occurring in nature can vary significantly, especially in terms of their duration. Preparing one model or system that is able to detect both long (over 1 s) and short (under 100 ms) sound events can be challenging. To address this problem, in each of the prepared submissions we use three models responsible for the detection of events of various lengths. The models' architecture is based on a siamese network [5], with two identical convolutional feature extractors and one mutual fully connected head. Each model can be treated as a classifier whose task is to determine whether two input samples belong to the same class.

2. DATASETS AND PREPROCESSING

2.1. Dataset

During preparation of the presented solution, only the DCASE2022 Task 5 development dataset was used. It consists of training (containing 21 hours of audio recordings) and validation (containing almost 6 hours of audio recordings) splits. The sampling rate of audio samples ranges between 6 kHz and 48 kHz.

2.2. Preprocessing

Initially, all recordings were upsampled to 48 kHz, due to the input requirements of the pre-trained OpenL3 network [6]. All positive examples from the training split were cut out from recordings based on the provided onset and offset times. Additionally, parts of the audio samples that did not contain any of the target classes were cut out and labeled as negative examples.

As mentioned in section 1, each of the prepared solutions consists of three models designed for detecting sound events of different lengths. The preprocessing phase for each case is explained in subsections 2.2.1 and 2.2.2.

2.2.1. Models for short and medium sound events

During training and inference of models intended for the detection of short (under 100 ms) and medium-length (between 100 ms and 1 s) sound events, each audio sample was transformed to a mel spectrogram. In both cases, training was performed on 200 ms segments. Table 1 presents parameters used to generate the mel spectrograms.

Additionally, during the fine-tuning and inference phases (described in detail in sec. 3.2 and 3.3), the input segment length has been reduced to 100 ms and STFT hop size to 64. It made short sound events more significant on the generated spectrogram without changing the size of the input tensor.

2.2.2. Model for long events

Due to the fact, that for detection of long events we use a pre-trained OpenL3 network as the convolutional feature extractor (the whole architecture of the solution is described in detail in sec. 3.1), we do not have to compute the mel spectrogram as it is an internal part of the used implementation. The only preprocessing step that has to be done in this case is trimming input audio samples into 1 s segments.

Table 1: Mel spectrogram parameters.

Model	mels	hop size	n_fft	f_{min} [Hz]	f_{max} [Hz]
short	128	128	1024	0	16000
medium	128	128	4096	0	16000

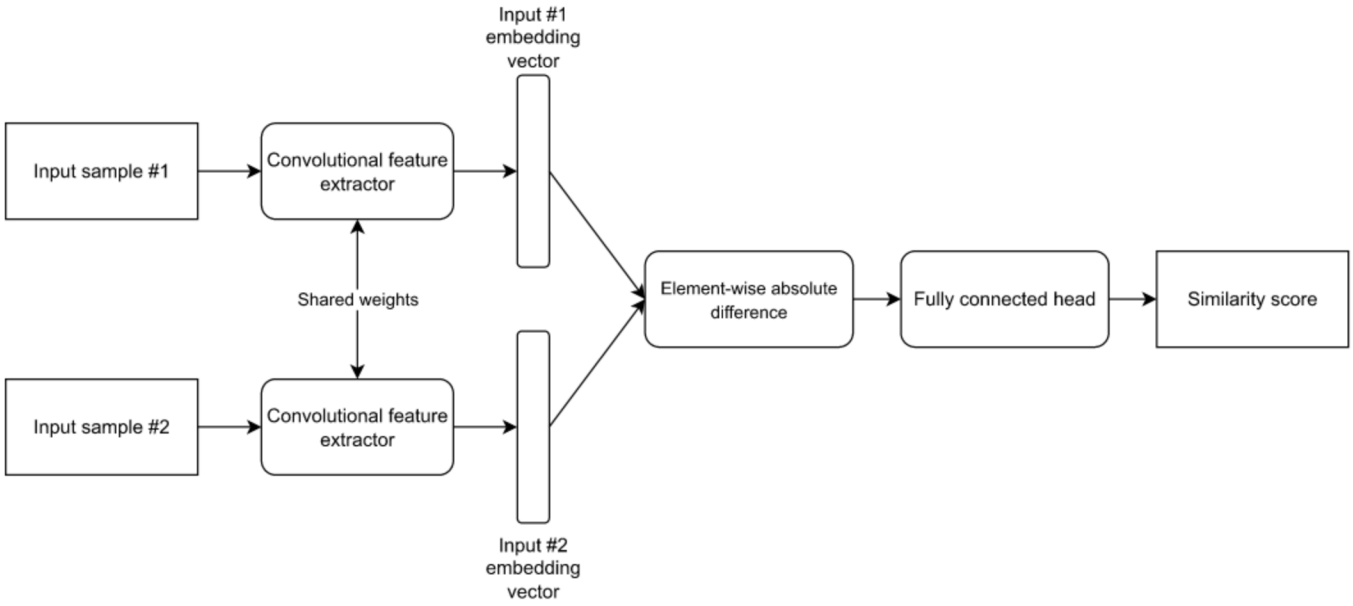

Figure 1: Model architecture.

2.3. Augmentation

During training, two methods of online data augmentations were used. Initially, the input data was randomly cropped to the appropriate length. In the following step, the input samples were mixed with artificial noise generated based on the algorithm presented in [7]. We used the implementation from [8] with the sigma parameter set to 2. After the mixing procedure we obtain samples of the same RMS energy as the input, while maintaining the appropriate signal-to-noise energy ratio. Other methods of data augmentation, including *SpecAugment* [9], were also tested, but they did not improve the quality of the model.

3. PROPOSED METHOD

3.1. Model architectures

The main goal of the proposed neural networks is to estimate the similarity between two input samples. For this purpose, we design a siamese network consisting of a convolutional feature extractor and a fully connected head. Figure 1 shows an overview of the proposed architecture. Details of the depicted components are described in sections 3.1.1 and 3.1.2.

3.1.1. Feature extraction

The first stage of each model consists of a feature extraction module that produces embedding vectors for the input audio samples. For the short and medium event models we train a custom convolutional neural network based on VGG-16 [10] from scratch (Table 2), whereas for the long event model we opt for the pre-trained OpenL3 network (Table 3).

In all proposed models, two preprocessed samples are delivered to the input. During forward propagation, the embedding vectors are determined for each of the samples separately using exactly the same feature extractor (shared weights). Finally, the element-wise

Table 2: Architecture of model prepared for short and medium events detection.

Layer type	Number of output features
ConvBlock*	64
ConvBlock*	128
ConvBlock**	256
ConvBlock**	512
ConvBlock**	512
Flatten	256
Fully connected + ReLU	256

ConvBlock* - 2 x (Conv2D + ReLU)
ConvBlock** - 3 x (Conv2D + ReLU)

Table 3: Architecture of model prepared for long events detection.

Layer type	Number of output features
CustomMelSTFT	-
ConvBlock***	64
ConvBlock***	128
ConvBlock***	256
ConvBlock***	512
Flatten	256
Fully connected + ReLU	256

ConvBlock*** - 2 x (BatchNorm + Conv2D) + ReLU

Table 4: Fully connected classifier architecture.

Layers type	short	medium	long
Linear + ReLU	2048	2048	256
Linear + ReLU	2048	2048	256
Linear + Sigmoid	1	1	1

absolute difference between these vectors is calculated and fed to the next stage of the system.

3.1.2. Fully connected head

The second part of each of the proposed models is the fully connected network, which is responsible for estimating the similarity between the input samples. Table 4 lists the sequence of the used layers.

3.2. Training

During the training phase, the models were treated as classifiers, whose task is to assess whether the two input samples belong to the same class. For this purpose, a binary cross-entropy loss was used. Adam with the initial learning rate set to 0.0001 was selected as the training optimizer. In addition, learning rate was scheduled to be multiplied by 0.99 every other epoch. All prepared models were trained for 300 epochs, however the model with the lowest validation loss was considered the best. In order to avoid overfitting to the training split, data augmentation methods described in paragraph 2.3 were used. Each training batch consisted of 40 % positive (both samples belonged to the same class) and 60 % negative pairs, which included samples containing sounds generated by two different species of animals, as well as animal-background pairs.

For each evaluation file (each new class), additional fine-tuning was performed independently. During this process we used the same loss function, optimizer and learning rate scheduler as for the main training phase. The main difference was creating positive pairs using only five examples of a given class. Each batch consisted of 80 % negative pairs. The entire fine-tuning took 100 steps and the last weights were used in evaluation.

3.3. Inference and post-processing

The aim of the inference process is to determine the average similarity score of the tested samples to each of the five labeled target events. In the case of detection of long and medium-length events, we elected to use a sliding window with a step of 100 ms. For short events, during inference, we decided to stretch the mel spectrogram by shortening the input segment length from 200 ms to 100 ms and reduce the STFT hop_size value by 50 %. In addition, the sliding window step was set to 10 ms.

In most of the prepared models, we used output score thresholding for determining onsets and offsets of events. The thresholds differ depending on the model and have been set based on the validation split. In addition, gaussian filtering with sigma parameter set to 9 was used to smooth out the predictions for the model designed to detect long audio events.

In the case of the model dedicated to the detection of short audio events, which was used in submission *ZGORZYNSKI_SRPOL_task5_1*, the post-processing module differs significantly. It uses a median filter to smooth out the predictions and peak picking for event onset and offset estimation.

Table 5: F-score values on validation split (in %).

Submission name	Overall	HB	PB	ME
ZGORZYNSKI_SRPOL_task5_1	67.3	91.3	44.4	90.0
ZGORZYNSKI_SRPOL_task5_2	59.4	91.3	35.1	90.0
ZGORZYNSKI_SRPOL_task5_3	60.0	91.7	35.2	93.5
ZGORZYNSKI_SRPOL_task5_4	57.2	89.0	32.5	96.1

4. RESULTS

We selected four systems for the DCASE2022 Task 5 final evaluation. Submissions *ZGORZYNSKI_SRPOL_task5_2*, *ZGORZYNSKI_SRPOL_task5_3* and *ZGORZYNSKI_SRPOL_task5_4* differ from each other by random selection of the validation set from the provided training split. DCASE 2022 Task 5 validation split was used only for selection of the best models. In case of submission *ZGORZYNSKI_SRPOL_task5_1*, a modified post-proccessing method was used, as described in section 3.3. Table 5 presents the per-dataset and overall F-score values for the DCASE 2022 Task 5 validation split.

5. CONCLUSIONS

In this technical report, we presented systems and achieved results of the submissions to Task 5 of the DCASE2022. Prepared architecture was based on siamese network with fully connected head, which is responisble for similarity modeling. Each of the prepared solutions consists of three deep learning models used to detect sound events of various lengths. A fine-tuning phase, was used to increase the accuracy of the models on target sound events, based only on 5 samples of each class. Additionaly, post-processing methods had a significant impact on the F-measure calculated on the validation split.

Finally, we achieve an F-measure score of 67.3 % on DCASE2022 Task 5 validation set, compared to 29.6 % scored by the best baseline system. These results indicate that siamese networks are a promising approach to audio few-shot learning tasks.

6. REFERENCES

[1] http://dcase.community/challenge2022/.

[2] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang, "A closer look at few-shot classification," 2019. [Online]. Available: https://arxiv.org/abs/1904.04232

[3] Y. Wang, Q. Yao, J. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," 2019. [Online]. Available: https://arxiv.org/abs/1904.05046

[4] A. Parnami and M. Lee, "Learning from few examples: A summary of approaches to few-shot learning," 2022. [Online]. Available: https://arxiv.org/abs/2203.04291

[5] S. Dey, A. Dutta, J. I. Toledo, S. K. Ghosh, J. Llados, and U. Pal, "Signet: Convolutional siamese network for writer independent offline signature verification," 2017. [Online]. Available: https://arxiv.org/abs/1707.02131

[6] J. Cramer, H.-H. Wu, J. Salamon, and J. P. Bello, "Look, listen, and learn more: Design choices for deep audio embeddings," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 3852–3856.

[7] J. Timmer and M. König, "On generating power law noise," *AA*, vol. 300, pp. 707–710, 1995.

[8] https://github.com/felixpatzelt/colorednoise.

[9] 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 *Interspeech 2019*. ISCA, sep 2019. [Online]. Available: https://doi.org/10.21437%2Finterspeech.2019-2680

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014. [Online]. Available: https://arxiv.org/abs/1409.1556