LEARNED AGGREGATION IN CNN: ALL-CONV NET FOR BIRD ACTIVITY DETECTION

Technical Report

Anshul Thakur, Arjun Pankajakshan, Padmanabhan Rajan,

School of Computing and Electrical Engineering, IIT Mandi, India
anshul_thakur@students.iitmandi.ac.in, arjunp@projects.iitmandi.ac.in, padman@iitmandi.ac.in

ABSTRACT

The task 3 of DCASE 2018 i.e. bird activity detection (BAD) deals with identifying the presence or absence of bird vocalizations in a given audio recording. In this submission, we utilize an all-convolutional neural network (all-conv net) for BAD. The network is characterized by the utilization of convolutional operations to implement aggregation/pooling and dense layers. The aggregation operation implemented by convolution helps in capturing the inter feature-map correlations which are ignored in traditional max/average pooling. This helps in learning a function which aggregates the complementary information in various feature maps, leading to better bird activity detection. Building on the all-conv net, we utilize four different derivative systems which provide good validation and preview scores.

Index Terms— bird activity detection, all-convolution network, learned pooling

1. INTRODUCTION

Bird activity detection (BAD) [1] is generally the first module in any avian acoustic monitoring system which discriminates the audio recordings having bird activity from those recordings which do not contain any bird vocalizations. By discarding the recordings that do not contain any bird activity, BAD helps in reducing the audio data to be processed for various tasks such as segmentation and species identification. The task 3 of DCASE 2018 deals with BAD in the challenging field conditions. The incorporation of the flight calls in this challenge makes this task more difficult. The flight call recordings are generally far-field recordings that are characterized by low-energy flight calls with overwhelming background noise (low SNR). From the analysis of the development datasets (*Freefield, Warblr and BirdVox*), it can be inferred that *Freefield* and *Warblr* recordings mostly contain high-energy calls while *BirdVox* contains low SNR flight calls. Most of the positive recordings in *BirdVox* resemble closely with the negative recordings of *Freefield* and *Warblr*. Hence, the major challenge here is to develop a technique that could process flight calls correctly but without a loss in the ability to correctly detect the high SNR bird activity and vice-versa.

Inspired by the all-convolutional neural network for object recognition [2], we proposed an all-conv net for BAD [3] recently. The utilization of the convolution operation to implement pooling and dense layers characterize this all-conv net. The local features obtained from a convolution layer are pooled using a learned aggregation, implemented using strided convolution operations. This aggregation function is designated as learned pooling and aggregates the contemporary information present in different feature-maps. On the contrary, max-pool operations aggregates the information present in each feature-map individually. This behaviour of learned pooling helps in obtaining better discriminative features, leading to a better classification performance in comparison to the normal max-pooling. The performance of this all conv-net is comparable to the state-of-art BAD systems [4, 5, 6], while demanding significantly lesser number of trainable parameters.

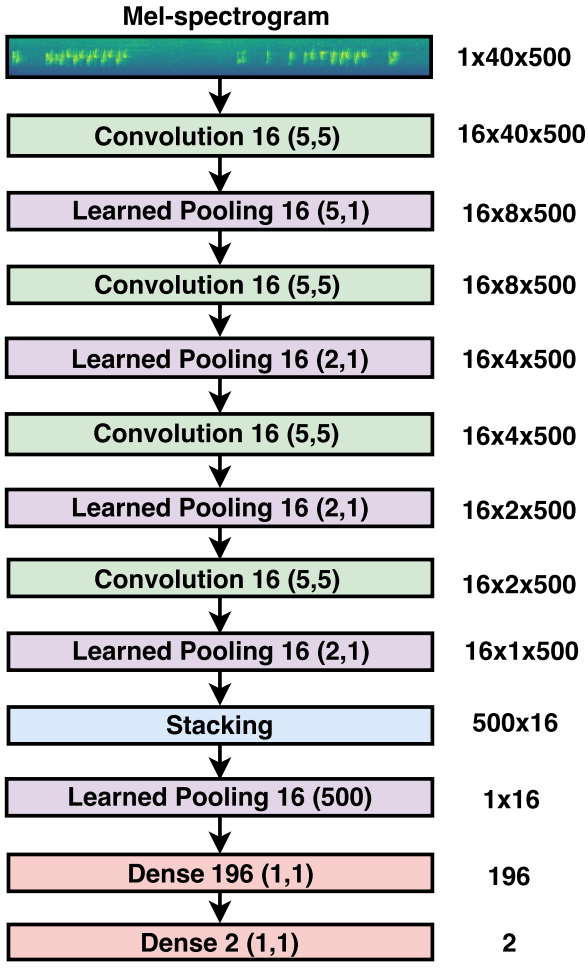

Figure 1: All-conv architecture for bird activity detection

In our submissions, we build upon this all conv-net to handle the variations present in the development data without utilizing any

pre-processing, augmentation, domain adaptation or external data. The rest of this report is organized as follows. In Section 2, the all-conv framework is described in detail. In Section 3, we describe our three submissions. The experimental results and conclusion are in Section 4 and Section 5 respectively.

2. ALL-CONV NET FOR BAD

The proposed architecture consists of four pairs of convolution and learned pooling layers followed by two $(1, 1)$ convolution layers. The input to the network is a 40×500 Mel-spectrogram. A kernel size of 5×5 with a stride of 1×1 is used in the convolutional layers. Each convolution layer has 16 filters. In order to pool the feature maps, the convolution with kernel size of 5×1 and 2×1 with a stride of 5×1 and 2×1 respectively is used at the subsequent layers. In the later part of the network, we have two 1×1 convolutional layers with 196 and 2 filters respectively. ReLU (Rectified Linear Unit) activation has been applied over all the convolutional and the learned pooling layers. While on the fully connected layer of 196 filters, we have applied sigmoid activation. Softmax is applied over the final 2-dimensional output of the network to get the probabilities of presence/absence of bird activity in any input Mel-spectrogram.

To avoid over-fitting, a dropout [7] with the probability of 0.25 and 0.5 has been used after convolutional layers and learned pooling layers respectively. The weights of the network has been initialized using random normal distribution. The network is optimized using Adam optimizer [8] with a learning rate of 0.001 and a decay of 10^{-6} and binary cross-entropy as the loss function. The network is shown in Fig. 1. These parameters are obtained empirically after an exhaustive search. More details about the network architecture can be found in [3]. The code for all-conv net is available at `http://git.io/fNgm7`. The total number of trainable parameters in all-conv net are 154,414.

Learned aggregation vs. max-pooling: The analysis of the filters learned at the first convolution layer of all-conv net confirms that the information learned by each filter can be regarded as complimentary to the other filters. This is illustrated in Fig. 2. The analysis of Fig. 2(b) illustrates that the 8th filter of the first convolution layer of the all-conv net is only concerned with learning the bird vocalizations. On the contrary, 11th filter is learning the background information (Fig. 2(c)). Thus, it can be deduced that each filter is learning a different behaviour or event. The utilization of this complementary information in the aggregation function can lead to more discriminative features. As discussed earlier, these inter feature-map correlations are ignored in max-pooling but are considered in the aggregation process implemented by the strided convolution operation. The conceptual difference in the working of max-pooling and learned aggregation is depicted in Fig. 3.

3. SUBMISSIONS

We are submitting four different systems, built upon the all-conv net, for the task 3 of DCASE 2018. In this section, we describe these three different systems. For all these submissions, LibROSA [9] is used to obtain the input Mel-spectrogram. A frame size of 40 ms with an overlap of 20 ms, 2048 fft points, Hamming window and 40 Mel bands are used to obtain the Mel spectrograms. These Mel-spectrograms are converted into decibels scale and are normalized between 0 and 1. These normalized Mel-spectrograms are given as input the all-conv net.

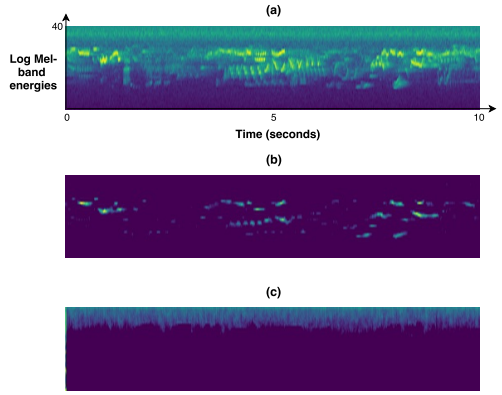

Figure 2: **(a)** Mel-spectrogram of an audio recording containing bird activity **(b)** Response of the 8th filter, learned in the first convolution layer, for the input Mel-spectrogram shown in (a) **(c)** Response of the 11th filter, learned in the first convolution layer, for the input Mel-spectrogram shown in (a).

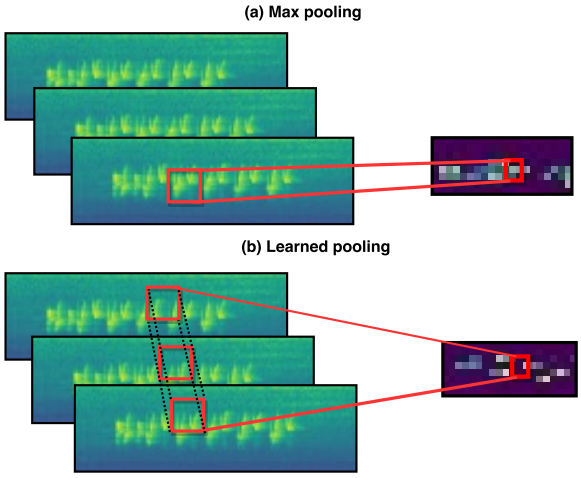

Figure 3: Illustration of the difference between (a) max pooling and (b) learned pooling.

3.1. All-conv net trained on *BirdVox, Freefield and Warblr* (ACN_1)

In our first submission (*ACN_1*), we have used all three development datasets to train the all-conv net. The 50% of audio recordings from each dataset are used for training the model, 10% are used for parameter tuning while remaining 40% are used for the performance evaluation.

3.2. Ensemble of two models (ACN_2)

In this submission, we use two separate all-conv nets having same architecture. The first model is trained on *Freefield* and *Warblr*. This trained model is used to initialize the weights of the second model. Then, the re-training is done using the *BirdVox* or flight data. An ensemble of these two models is used for detecting the bird activity. The final scores are generated by averaging the probabilities predicted by both the models. The intuition behind this approach is

to avoid the affect of flight calls on the detection of high SNR bird activity. Again, the 50% of audio recordings from each dataset are used for training the model, 10% are used for parameter tuning and 40% are used for the performance evaluation.

3.3. Three class approach (ACN_3)

In this approach, we trained an all-conv net to categorize three classes: flight calls (positive class in *BirdVox*), normal vocalizations (positive class in *Freefield* and *Warblr*) and no activity (negative class in all three development datasets). The final score i.e. probability of bird activity is obtained by adding the probabilities of first two classes. To accommodate three classes, the number of filters used in the last layer of all-conv net are changed from two to three. Also, the loss function is changed to the categorical cross-entropy. The rest of the architecture and hyper-parameters are kept same. For training, 50% of the positive examples from *BirdVox*, 50% of the positive examples from *Freefield* and *Warblr* and 35% of the negative examples from all the three evaluation datasets are used. The data distribution is done in such a way to avoid the extreme class imbalance.

3.4. Ensemble of All_conv with GRU (ACN_4)

In order to model the context information, we added a GRU layer with 16 hidden units after stacking i.e. after 9th layer of our all conv architecture. In this approach, we use an ensemble of two All_conv+GRU networks (having same architecture). First one is trained on *Freefield* and *Warblr*, while second is trained on *BirdVox*. The final probabilities are obtained by averaging the scores from both these models. 50% of the audio recordings from all development datasets are used for training in this approach. Addition of a GRU layer increases the number of parameters to 155,988.

4. RESULTS

In this section, we show the results obtained by our systems on both the evaluation and preview data. The amount of data used to train the models is already described in the previous section. The remaining audio recordings are used for the performance evaluation. Table 1 depicts the results obtained by the proposed systems on different datasets.

Table 1: AUC scores obtained by the proposed approaches on different datasets

	Freefield + Warblr	**BirdVox**	**Preview**
ACN_1	93.5	91.1	85.89
ACN_2	94	92	**90.54**
ACN_3	93.7	92.5	85.26
ACN_4	92.95	91.8	86.33

Following can be inferred from the analysis of the results tabulated in Table 1:

- The performances of all four systems are comparable on the development datasets. However, two-model ensemble approach (ACN_2) significantly outperforms the other models on the preview data. ACN_4 provides slightly better classification than ACN_1 and ACN_3. But, adding a GRU layer in two-model approach led to a decrease in the preview scores. Since we are not aware of the composition of this preview dataset, the generalization ability of these systems is still under question.
- Despite our reservations about the use of both flight calls and high SNR data in a single model, *ACN_1* and *ACN_3* provide respectable scores on the preview evaluation data as well as on all three development datasets.
- Treating flight calls as a separate class in ACN_3 has not yielded any desired result.
- Single models trained on both flight calls and high SNR calls i.e. ACN_1 and ACN_3 provide less preview score in comparison to the two-model ensemble (ACN_2). This highlights that the utilization of both flight and normal calls for training a model affect the detection ability of the model.

5. CONCLUSION

The four systems built upon the all-conv net are utilized for the task 3 of DCASE 2018. All three submissions provided good AUC scores for the development datasets. However, the experimentation shows that a two model ensemble approach yields better results that the single model approaches on the preview dataset. It will be interesting to see how well these approaches generalize on the evaluation datasets.

6. REFERENCES

[1] D. Stowell, M. Wood, Y. Stylianou, and H. Glotin, "Bird detection in audio: a survey and a challenge," in *IEEE Int. Workshop Mach. Learn. Sig. Process.*, 2016, pp. 1–6.

[2] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," in *Proc. ICLR*, 2015.

[3] A. Pankajakshan, A. Thakur, D. Thapar, P. Rajan, and A. Nigam, "All-conv net for bird activity detection: Significance of learned pooling," in *Proceedings of Interspeech*, Sept., 2018. [Online]. Available: http://faculty.iitmandi.ac.in/~padman/papers/learnt_pooling_cameraReady_interspeech2018.pdf

[4] E. Cakir, S. Adavanne, G. Parascandolo, K. Drossos, and T. Virtanen, "Convolutional recurrent neural networks for bird audio detection," in *Proc. Eusipco*, 2017, pp. 1744–1748.

[5] T. Grill and J. Schlüter, "Two convolutional neural networks for bird detection in audio signals," in *Proc. Eusipco*, 2017, pp. 1764–1768.

[6] T. Pellegrini, "Densely connected cnns for bird audio detection," in *Proceedings of Eusipco*, 2017, pp. 1734–1738.

[7] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[8] D. P. Kingma and L. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015.

[9] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, 2015, pp. 18–25.