Bird Audio Detection - DCASE 2018

Franz Berger, William Freillinger, Paul Primus, and Wolfgang Reisinger

Special Topics - Machine Learning and Audio: a challenge
Johannes Kepler University, Linz, Austria

***Abstract*— In this paper we explore three approaches on bird audio detection. We establish a simple baseline, experiment with handcrafted features and finally move to Convolutional Neural Networks.**

I. Introduction

For wildlife monitoring and other environmental projects, detecting bird calls in audio recordings is a common challenge, and can be a first step before classifying the bird species [1], [2].

This paper deals with the bird audio detection task from DCASE 2018 [1]. For this challenge, we were provided with 3 data sets of audio files (each set recorded under different condition e.g. time of day, type of microphones, length of recording), as well as 0/1 labels to encode whether there was at least one bird in the corresponding audio file. The task was to develop a method that determines for new audio files whether they included bird sounds, with a special focus on generalisation to new recording conditions. For evaluation we were provided with three different data sets (two of them had new data sources, one had the same source as the training data set, but different files), where data set membership was known for the evaluation files. The evaluation itself was done by calculating the harmonic mean of the Area under Receiver Operator Characteristic Curves (AUROCC) for the predictions for each single data set - because of this, the methods were encouraged not only to predict 0 or 1 for each file, but scores in the range [0, 1].

To tackle this problem, we initially made our own baseline with simple features and classifiers with no or few hyperparameters, such as Random Forest and Logistic Regression.

Our more sophisticated approaches were based on calculating the Mel-frequency spectrograms for each file (trimming and looping audio as necessary to get uniform audio duration), then classifying those spectrograms with CNNs.

The first CNN architecture we implemented was *bulbul*, which was a modification of the strongest-performing method of a similar bird audio detection challenge from 2017, suggested as a baseline by the organizers of DCASE.

puffin was another classification method, with which we tried to address *bulbul*'s lack of consideration given to the generalisation problem for different recording environments. Here, we trained in total six CNNs: One per data set to classify the files, and three pairwise discriminators. The dataset discriminators are used to weigh the predictions of the classifiers.

We also tried to extract some more features ourselves, to make learning easier for the neural networks or other classification methods. This approach was based on searching the spectrograms for patterns which might indicate the bird call (connected high-energy areas), then extracting windows around these patterns, for which then some summary statistics are computed. Furthermore we experimented with domain adaptation methods. Since the results were too poor, we didn't include this approaches in our final submission.

II. Related work

Stowell et al. [1] discuss different concrete bird audio detection tasks, and approaches to solve them, which include energy, spectrogram cross-correlation and hidden Markov models. A simple energy-based approach could be using a thresholding function, which is applied to a short time-windows and band-limited frequencies, which are expected to contain bird calls. Spectrogram cross-correlation uses sound templates which are supposed to represent the different classes, for which the cross correlation is calculated over subwindows of the spectrogram - these correlations can then be used as features [1], [3]. Hidden Markov models, which use a sequence of spectral vectors as input, are more powerful than simple template-matching techniques and are also

[1] `http://dcase.community/challenge2018/task-bird-audio-detection`

often used in bioacoustics, for example in speech recognition [1], [4].

A CNN-based approach (taking Mel-spectrograms of uniform size as input) is described by Grill et al. [2], where they compare different network architectures and input transformations.

Shen et al. [5] show how to do domain adaptation - in this case Wasserstein Distance Guided Representation Learning - to overcome the difficulty of training and test set originating from different distributions.

III. Data sets

The DCASE2018 challenge provides three datasets for development: freefield1010, BirdVox-DCASE-20k and warblrb10k. Furthermore, three evaluation datasets are provided: Chernobyl, PolandNFC and held out subset from warblrb10k. Datasets mostly contain 10 second long WAV files (44.1 kHz mono PCM). Although stated differently in the challenge description, the warblrb10k dataset contains recordings of lengths between approximately two and forty seconds. The files are manually labelled with 0 or 1, indicating the absence/presence of a bird anywhere within the file. The labelling accuracy of the ground truth is estimated as 96.7% or better.

Since the main goal of this challenge is to develop a model which generalises to different recording conditions, the provided datasets are quite heterogeneous. From the dataset descriptions these conditions are roughly known for all development and evaluation sets:

- **freefield1010:** excerpts from field recordings around the world and standardised for research.
- **warblrb10k:** smart phone recordings from around the UK. Recordings include weather noise, traffic noise, humans speech and even human imitations of birds.
- **BirdVox-DCASE-20k:** collected from remote monitoring units placed near Ithaca, NY, USA during autumn of 2015.
- **Chernobyl:** collected with unattended remote monitoring equipment in the Chernobyl Exclusion Zone. Recordings cover different weather conditions, mammal and insect noise and are sampled across various environments such as abandoned villages, grassland and forest areas.
- **PolandNFC:** recordings from monitoring autumn nocturnal bird migration collected on 15 days from September to November of 2016 on the Baltic Sea coast, Poland. Clips contain different weather conditions and background noises including wind, rain, sea noise, insect noise, human voice and deer calls.

	Dataset	#s	#p	#n
Development	BirdVox-DCASE-20k	2.000	10.017	9.983
	freefield1010bird	7.690	1.935	5.755
	warblrb10k	8.000	6.045	1.955
Evaluation	warblrb10k	2.000	?	?
	Chernobyl	6.620	?	?
	PolandNFC	4.000	?	?

TABLE I: Size and class distribution of development and evaluation data sets. #s denotes the number of samples, #p is the number of positive and #n is the number of negative samples

Beside differences in recording conditions, development datasets differ also in their size and class distribution (Table I). For evaluation datasets no information about class distribution is given.

IV. Methods

A. Baseline

To gain more insight into the difficulty of the given task we start by creating a simple, yet stable baseline. From the audio files we extract mel-filtered spectrograms with 128 bins, in the frequency range of 25 Hz to 11025 Hz. The spectrograms are further normalized to zero mean and unit variance. Two approaches are considered for our baseline.

1) Simple Survey Statistics: For the first approach, features based on the spectrogram representation of an input file were calculated. The statistics include mean, median, minimum, maximum of each bin. Classification was done by a random forest with 10000 estimators and logistic regression.

2) Framewise: For our second baseline approach labels are predicted on a per frame level. The input vector consists of the 128 bins of the Mel-spectrogram and we used a random forest classifier. We considered the provided, file based, ground truth as weak labels and every frame of a file was labeled with the file based label. The final prediction was done with logistic regression on the predicted frame labels.

B. Convolutional Neural Network

Most promising results in previous bird detection tasks where obtained by learning feature representations from raw (Mel-)spectograms with deep feed forward convolutional neural networks (CNN).

Layer	# nodes	Output Dimension	Activation
Conv(3×3)	16	$16 \times 998 \times 78$	LeakyReLu
MaxPool(3×3)	-	$16 \times 332 \times 26$	
Conv(3×3))	16	$16 \times 330 \times 24$	LeakyReLu
MaxPool(3×3)	-	$16 \times 110 \times 8$	
Conv(1×3)	16	$16 \times 108 \times 8$	LeakyReLu
MaxPool(1×3)	-	$16 \times 36 \times 8$	
Conv(1×3)	16	$16 \times 34 \times 8$	LeakyReLu
MaxPool(1×3)	-	$16 \times 11 \times 8$	
Faltten		1408	
Dense	256	256	LeakyReLu
Dense	32	32	LeakyReLu
Dense	1	1	Sigmoid

TABLE II: Architecture of *bulbul*

1) Input Features: For all three architectures the same input features are used. Audio files are transformed into a two dimensional spectral representation by Short Time Fourier Transformation (STFT). Sampling rate of provided audio files is 44.1 kHz. Window size for Fast Fourier Transformation (FFT) is set to $W = 2048$ which corresponds to 46.44 ms. For the FFT, Windows are multiplied with a Hann function. STFT hop size is set to 512 samples, one frame in the spectrogram therefore corresponds to a time step of 11.61 ms. The linear frequency scale is transformed to the Mel scale by applying 80 triangular filters. Frequencies below 25 Hz and above 15025 Hz are ignored. We apply dynamic range compression to the obtained power spectrogram by adding one and taking the logarithm. Afterwards Mel bands are normalized to zero mean and unit variance.
All architectures expect a fixed input size of 80×1000 (bands $\times$ frames), therefore samples shorter than 11.61 s are looped to the desired size.

2) Bulbul: The *bulbul* model is a rebuild of [2], originally created for BAD Challenge 2016/2017 hosted by Queen Mary University of London. [2]. The architecture is described in Table II.

3) Puffin: A possible disadvantage of the *bulbul* model is that it might not be able to generalize to data recorded under different conditions. Results from training and evaluating specialized models for individual datasets indeed suggest that this is the case. In our experiments specialized models perform between 5 and 20 percentage points (pp) better than models trained on datasets recorded under different conditions, as can be seen in Table III and Table IV.
The idea of *puffin* is to use the predictive power of specialized models combined with a pre-classification stage which assigns new recordings to one of the training sets based on similarity. For this similarity pre-classification a CNN with the same architecture as *bulbul* is used. We use a 3-way softmax to indicate dataset simliarity for the final model. Predictions of specialized models are combined by weighting predictions with the corresponding softmax activation of the pre-classification stage, as can be seen in Figure 1.

[2]http://machine-listening.eecs.qmul.ac.uk/bird-audio-detection-challenge/

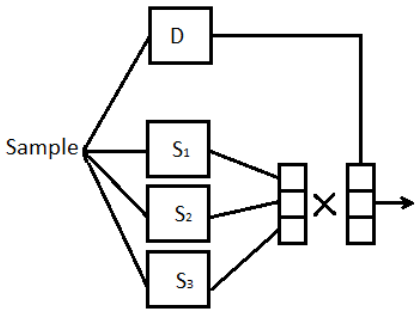

Fig. 1: Architecture of *puffin*. D outputs the sample similarity to each of the three training datasets (3-way softmax). S_1, S_2 and S_3 are specialized classifiers, each trained on one of the training sets. The final prediction is obtained by weighting the specialized prediction with the corresponding similarity value.

Discrimination between warblrb10k - ff1010bird reaches an area under the receiver operating characteristics curve (AUROCC) of 97%. For BirdVox-DCASE-20k - warblrb10k and freefield1010bird - BirdVox-DCASE-20k both discriminators reach AUROCC over 99%, suggesting highly different characteristics of datasets. Evaluation of discriminators is done on hold-out sets with 30% of the original data.

Validation Set	AUROCC	ACC	TPR	FPR
BirdVox-DCASE-20k	0.9275	0.8674	0.8733	0.1390
warblrb10k	0.9234	0.8656	0.8159	0.0872
freefield1010bird	0.8984	0.8338	0.7787	0.1089

TABLE III: Results of training and validating *bulbul* model on the same dataset (intra). Validation is done with randomly selected 10% of samples.

4) Training: Training is done by stochastic gradient decent with mini-batches of size 64. The Adam optimizer [6] with learning rate of 0.001 is used for updates. The learning rate is decreased by a factor of 2

Validation Set	AUROCC	ACC	TPR	FPR
freefield1010bird	0.8592	0.8000	0.7297	0.1297
warblrb10k	0.8411	0.7693	0.8005	0.2620
BirdVox-DCASE-20k	0.7234	0.6587	0.5276	0.2103

TABLE IV: Results of training and validating *bulbul* model on different datasets (inter) . Validation is done with randomly selected 10% of samples.

after three epochs of no improvement. The number of epochs is determined by early stopping, the patience parameter is set to twelve epochs. Improvement is measured in terms of AUROCC on the validation set. One epoch consist of 1500 batch updates.
Recordings are augmented by cyclic random shifting in time and random shifting of frequency bands $\pm$1band. Empty bands created by frequency shifting are linearly interpolated.
All models use sigmoid activation as output function for the bird classification task, therefore logistic loss is the natural choice. In order to be able to extend the pre-classification stage of *puffin* to multiple datasets we chose softmax as activation and categorical cross entropy as the loss function.

V. Results

A. Baseline

The baseline uses stratified 3-fold cross validation. Two training sets are combined and the third is used for validation

Model	AUROCC	ACC	TPR	FPR
random forest	0.4909	0.5006	0.1756	0.3806
logistic regression	0.4921	0.5268	0.2241	0.3316
framewise	0.4854	0.4909	0.4883	0.5303

TABLE V: Baseline Validation results. Results are the harmonic mean of individual results per fold

B. CNN

For *bulbul* and *puffin* we apply stratified 3-fold cross validation, where one fold corresponds to one dataset. Two folds are used for training and one for validation. This procedure allows to estimate how models behave under different recording conditions. For early stopping and learning rate decay we randomly select a 10% subsample of the data in the training folds as a validation set.
For model performance estimation, the AUROCC of the validation set is computed for each fold and averaged via harmonic mean. This method is equal to the evaluation procedure of submitted predictions on the evaluation set. Results for *bulbul* and *puffin* are summarized in Table VI. Best results where obtained by combining both methods by averaging over their predictions.

Model	AUROCC	ACC	TPR	FPR
bulbul+puffin	0.8171	0.7471	0.6303	0.1358
puffin	0.8170	0.7479	0.6353	0.1377
bulbul	0.8032	0.7374	0.6645	0.1843

TABLE VI: 3-fold cross validation results of models *puffin*, *bulbul* and a averaged combination.

VI. Conclusion

Out of the three approaches we conducted experiments on, only the CNN seems worthwhile to pursue. The complexity of the problem is too large for the simplest methods we used as our baseline. A handcrafted approach could be worthwhile, however a high amount of domain knowledge would be necessary to incorporate it in the classification process. What we tried to achieve with it is most likely learnable by the CNN without the manual adjustment of additional hyperparameters.

References

[1] D. Stowell, M. Wood, Y. Stylianou, and H. Glotin, "Bird detection in audio: a survey and a challenge," in *Machine Learning for Signal Processing (MLSP), 2016 IEEE 26th International Workshop on*. IEEE, 2016, pp. 1–6.

[2] T. Grill and J. Schlüter, "Two convolutional neural networks for bird detection in audio signals," in *25th European Signal Processing Conference, EUSIPCO 2017, Kos, Greece, August 28 - September 2, 2017*, 2017, pp. 1764–1768. [Online]. Available: https://doi.org/10.23919/EUSIPCO.2017.8081512

[3] M. Lasseck, "Bird song classification in field recordings: winning solution for nips4b 2013 competition," in *Proc. of int. symp. Neural Information Scaled for Bioacoustics, sabiod. org/nips4b, joint to NIPS, Nevada*, 2013, pp. 176–181.

[4] M. Gales, S. Young *et al.*, "The application of hidden markov models in speech recognition," *Foundations and Trends® in Signal Processing*, vol. 1, no. 3, pp. 195–304, 2008.

[5] J. Shen, Y. Qu, W. Zhang, and Y. Yu, "Wasserstein distance guided representation learning for domain adaptation," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018*, 2018. [Online]. Available: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17155

[6] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980