ACOUSTIC BIRD DETECTION WITH DEEP CONVOLUTIONAL NEURAL NETWORKS

Mario Lasseck

Museum fuer Naturkunde
Leibniz Institute for Evolution and Biodiversity Science
Invalidenstraße 43, 10115 Berlin, Germany
Mario.Lasseck@mfn.berlin

ABSTRACT

This paper presents deep learning techniques for acoustic bird detection. Deep Convolutional Neural Networks (DCNNs), originally designed for image classification, are adapted and fine-tuned to detect the presence of birds in audio recordings. Various data augmentation techniques are applied to increase model performance and improve generalization to unknown recording conditions and new habitats. The proposed approach is evaluated on the dataset of the Bird Audio Detection task which is part of the IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE) 2018. It surpasses previous state-of-the-art achieving an area under the curve (AUC) above 95 % on the public challenge leaderboard.

Index Terms— Bird Detection, Deep Learning, Deep Convolutional Neural Networks, Data Augmentation

1. INTRODUCTION

Automated bird detection is an important tool for acoustic wildlife monitoring. It can serve as a first step to filter large datasets and reduce human as well as computational effort by focusing on regions of interest with bird activity before conducting further analysis like e.g. species identification or population estimation.

In the DCASE 2018 Bird Audio Detection challenge participants are asked to decide whether or not there are any birds present in short excerpts of audio recordings. For training, three development datasets are provided recorded in different parts of the world. The test data is recorded in monitoring scenarios not matching the training data making it necessary to develop models inherently generalizing well to unknown recording conditions and new habitats. The task is an expanded version of the Bird Audio Detection challenge which ran in 2016/2017. An overview and further details about task and data provided for training and evaluation are given in [1] and [2].

2. IMPLEMENTATION DETAILS

To detect the presence/absence of bird sounds in audio recordings the best model of the LifeCLEF 2018 [3] Bird Identification Task [4] is adapted to binary classification. The original model designed to identify 1500 bird species is described in [5].

2.1. Data Preparation

All audio files of the development and evaluation sets are first pre-processed by applying a shallow high pass filter (Q = 0.707) with a cutoff frequency at 2 kHz and furthermore resampled to 22050 Hz. The filter reduces low frequency energy improving signal-to-noise ratio for frequency bands relevant to bird sounds. Downsampling reduces the amount of data to process without losing too much relevant acoustic information.

2.2. Training Setup

To develop an acoustic bird detection system, DCNNs pre-trained on ImageNet [6] are fine-tuned with mel spectrogram images representing short audio chunks. Training is done via PyTorch [7] utilizing PySoundFile and librosa [8] python packages for audio file reading and processing. The same basic pipeline as for the BirdCLEF 2018 task is used for data loading and can be summarized as follows:

- extract audio chunk from file with duration of ca. 4 seconds
- apply short-time Fourier transform
- convert to mel spectrogram
- remove low and high frequencies
- normalize and convert power spectrogram to decibel units
- resize spectrogram to fit input dimension of the network
- convert grayscale image to RGB image

As recommended by the challenge organizers, two development sets are used for training and the remaining one for performance validation. Since BirdVox-DCASE-20k is larger than the other two datasets combined, it is always part of the training set and therefore only two folds (see Table 1) out of three possible combinations are used for cross-validation:

Table 1: Training/validation splits used for training

Fold	Training sets	Validation set
1	ff1010bird, BirdVox-DCASE-20k	warblrb10k
2	warblrb10k, BirdVox-DCASE-20k	ff1010bird

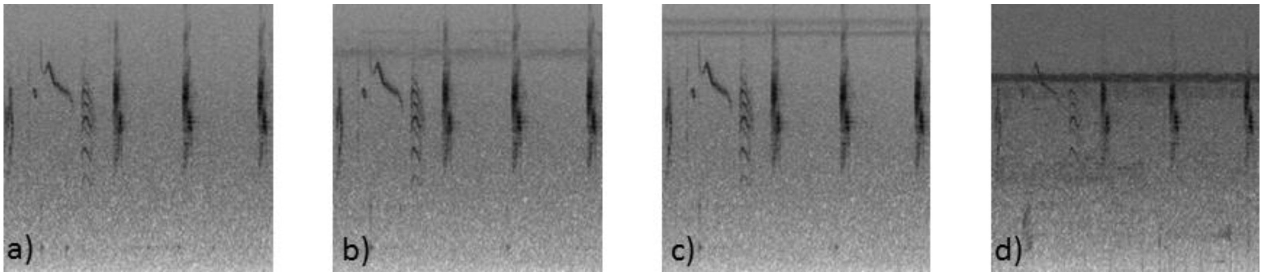

Figure 1: Examples of data augmentation via adding chunks from random files with same label or label 0 (no bird). a) mel spectrogram of original audio chunk without augmentation; b), c) & d) mel spectrogram with augmentation

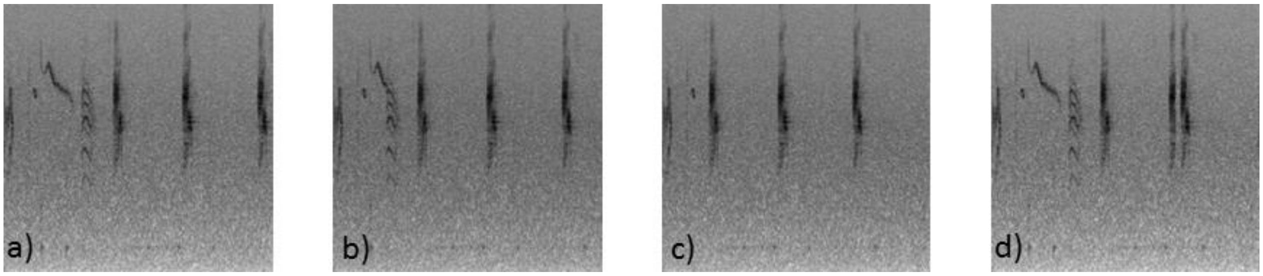

Figure 2: Examples of data augmentation via time interval dropout. a) mel spectrogram of original audio chunk without augmentation; b), c) & d) mel spectrogram with augmentation

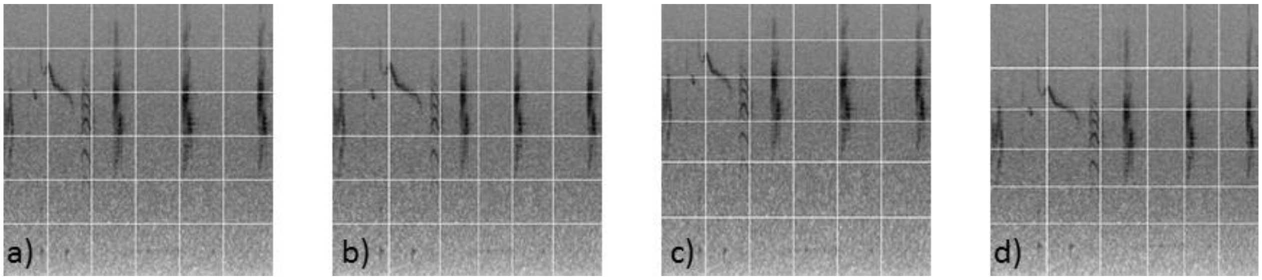

Figure 3: Examples of data augmentation via piecewise time and frequency stretching (with grid overlay for better visualization). a) mel spectrogram of original audio chunk without augmentation; b) local time stretching; c) local frequency stretching; d) combined local time and frequency stretching

In each training epoch all files of the training set are processed in random order to extract audio chunks at random position. Training is done with a batch size between 80 and 90 samples using up to three GPUs (Nvidia Geforce 1080 and 1080 Ti). Categorical cross entropy is utilized as loss function and stochastic gradient descent as optimizer with Nesterov momentum 0.9, weight decay 1e-4 and a constant learning rate of 0.01. For validation and test sets audio chunks are extracted successively from each file with an overlap of 10 % for validation files during training and 50 % for files in the evaluation test set. Predictions are summarized for each file by taking either the mean or maximum over all chunk predictions per file.

2.3. Data Augmentation

To increase model performance and improve generalization to new recording conditions and habitats, various data augmentation techniques are applied in both time and frequency domain. The following methods are applied in time domain regarding audio chunks:

- apply jitter to chunk duration (ca. ± 0.4 s)
- extract chunks from random position in file (wrap around if end of file is reached and continue from beginning)
- add 3 audio chunks from random files with label 0 (no bird)
- add 2 audio chunks from random files with the same label
- apply random factor to signal amplitudes of all chunks before summation (superposition)
- apply random cyclic shift
- apply time interval dropout by skipping random number of samples

The audio chunk (or sum of chunks) is then transformed to frequency domain via short-time Fourier transform with a window size of 1536 samples and a hop length of 360 samples.

Table 2: Model properties

Model ID	M1	M2	M3	M4	M5	M6
Included in submission	1	2,3,4	3,4	3,4	3,4	4
Network architecture	Inception	Inception	Inception	Inception	Inception	ResNet
Chunk duration [s]	4.0	4.0	4.0	3.5	4.0	4.0
Chunk duration jitter [s]	0.45	0.45	0.45	0.4	0.4	0.45
Chance to add 1st chunk w. label 0 [%]	75	75	75	80	70	75
Chance to add 2nd chunk w. label 0 [%]	75	75	50	80	70	75
Chance to add 3rd chunk w. label 0 [%]	50	50	0	60	40	50
Chance to add 1st chunk w. same label [%]	50	50	50	60	60	50
Chance to add 2nd chunk w. same label [%]	50	50	50	50	60	50
Batch size	90	90	90	84	84	78
Training epochs	76	52	177	44	68	217
Pooling method	mean	max	max	max	max	max
Fold	1	2	2	2	2	2
AUC val. set [%]	93.46	93.21	92.22	93.03	92.98	92.01
AUC test subset [%]	90.40	93.66	-	-	-	-

Frequencies are mel scaled with low and high frequencies removed resulting in a spectrogram of 310 mel bands representing a range of approximately 160 to 10300 Hz. Normalization and logarithm is applied to the power spectrogram yielding a dynamic range of approximately 100 dB. The final spectrogram image is resized to 299x299 pixel to fit the input dimension of the InceptionV3 network (or 224x224 pixel for ResNet). Since networks are pre-trained on RGB images, the grayscale image is duplicated to all three color channels. Further augmentation is applied in frequency domain to the spectrogram image during training:

- frequency shifting/stretching by cutting a random number of the first 10 and last 6 rows of the image
- piecewise time/frequency stretching by resizing a random number of columns/rows at random position
- use of different interpolation filters for resizing
- apply color jitter (brightness, contrast, saturation, hue)

Some of the above augmentation techniques were already applied successfully by other teams in previous bird identification or detection tasks like e.g. adding background noise or sounds from files belonging to the same class with random intensity [9,10,11] or applying cyclic shift to the sample array by a random amount [9,11,12]. Pitch or frequency shifting and stretching was previously used in similar ways by e.g. [13], [14] and applying color jitter is very common for training image classifiers. A few augmentation methods, further improving model accuracy and generalization, were newly introduced this year for the LifeCLEF 2018 Bird Identification Task. They also significantly help to increase performance regarding bird detection and are described briefly in the following sections.

With a chance of 30 % **time interval dropout** is realized by skipping a random number of samples (between zero and length of an entire chunk) at a randomly chosen position when reading from the audio file (see Figure 2).

Besides manipulating the duration or speed of an entire chunk, **piecewise time stretching** is applied with a 50 % chance to change speed at multiple times within a chunk. This is accomplished by dividing the spectrogram image in several vertical pieces, each having a width randomly chosen between 10 and 100 pixel. Afterwards, pieces are resized individually by a factor randomly chosen between 0.9 and 1.1 along the horizontal (time) axis. To realize local or **piecewise frequency stretching** the same procedure is applied in an analogous manner to the vertical frequency axis with a 40 % chance and a stretch factor between 0.95 and 1.15 (see Figure 3).

For resizing, the high-quality Lanczos filter of the Python Imaging Library is used by default. However, in 15 % of the cases a **random choice of interpolation filter** is realized using different resampling filters from the library (Nearest, Box, Bilinear, Hamming and Bicubic).

3. RESULTS

Detection performance is evaluated via the area under the curve metric. Scores on validation data and public leaderboard test data are listed in Table 3 along with models or ensemble of models belonging to each DCASE 2018 submission. Properties of individual models are summarized in Table 2. Models mainly differ in duration of chunks, duration jitter and conditional probabilities of chunks superimposed for augmentation. Most models use an InceptionV3 architecture [15] except one which uses a 152-layer residual network (ResNet-152) [16]. For the first two submissions a single model was trained with equal properties but on different training/validation splits (see Table 1): M1 on fold1 and M2 on fold2. M1 gives better performance on the warblrb10k validation set compared to M2 regarding the ff1010bird validation set but M2 performs much better on the public test subset. Also M2 was submitted twice to the public leaderboard using different pooling methods to summarize chunk predictions per audio file. Taking the mean over all chunks results in 93.32 % AUC compared to taking the maximum which obtains an AUC of 93.66 %. As a consequence all following models were trained on fold2 with max pooling of chunk predictions. For the third and fourth DCASE submission different models (see Table 2 and 3) were ensembled by averaging their file-based predictions. For submission five, model M2 was trained from scratch without using pre-trained weights (see discussion section).

Table 3: Submission models and evaluation scores

Submission index	1	2	3	4	5
Submission date	24/7	26/7	28/7	30/7	3/8
Models in ensemble	M1	M2	M2, M3, M4, M5	M2, M3, M4, M5, M6	M2* (not pre-trained)
AUC val. set [%]	93.5	93.2	93.8	93.8	91.0
AUC test set [%]	90.4	93.7	95.0	95.3	91.4

4. DISCUSSION

The BirdCLEF 2018 model designed to identify individual bird species was successfully adapted to the binary classification task detecting bird activity of any kind in audio recordings. Even with a single model, detection performance of more than 93 % AUC can be achieved on unseen data not matching the conditions of the training set. By ensembling models with different properties results can be further improved to above 95 % AUC.

In the following section a few differences between BirdCLEF and DCASE system are pointed out. For bird detection, smaller chunks tend to work better (4s vs. 5s). Interestingly, decreasing the learning rate after a few epochs or squaring chunk predictions before pooling didn't help to further improve detection performance. Also, some augmentation techniques applied for species identification were not used for the bird detection task for example reconstructing the audio signal by mixing individual sound elements or choosing files from different versions of the development set (with/without high pass filtering, artificially degrading audio quality by encoding files to mp3 with low bit rate, removing silent parts containing only background noise, etc.). Without these augmentation methods data preparation is greatly simplified, especially since no segmentation of audio files into signal and noise parts is required. It would be interesting to investigate whether or not these additional techniques are able to further increase detection results. An overview to what extent individual augmentation techniques are able to increase performance on species identification is given in Table 1 in [5]. The most effective augmentation methods are:

1. adding noise/content from random files
2. piecewise time and frequency stretching
3. time interval dropout

These techniques also proved to be successful for the bird detection task (see examples in Figure 1-3).

Other approaches weren't able to further improve detection performance. For example keeping the original sample rate of 44.1 kHz didn't achieve better results but made the training significantly slower. Also, fine-tuning a model pre-trained on the BirdCLEF dataset didn't lead to better results but convergence was much faster during training (92 % AUC in 5 epochs compared to ca. 10 epochs for networks pre-trained on ImageNet).

Although getting rather low scores on the validation set, adding a ResNet-152 based model to the ensemble of submission 3 helped to increase performance of submission 4. The different network architecture seems to complement the Inception predictions quite well.

As mentioned before, models were fine-tuned using neural networks pre-trained on the "trimmed" Large Scale Visual Recognition Challenge (ILSVRC) [17] version of ImageNet, a dataset with almost 1.5 million photographs of 1000 object categories scraped from the web. It is not related to spectrogram images of bird sounds or audio in general. Since it is an external dataset not on the list of pre-registered datasets allowed in the challenge, results of submission 1-4 cannot be directly compared with those of other teams and are not part of the official final challenge scores. However, in a post-challenge experiment model M2 was trained from scratch without transfer learning. It obtains 91.4 % AUC on the public leaderboard (see submission 5 in Table 3 and Figure 4) and provides the best system for the task ranking first place in the official challenge results [18]. This demonstrates, with the presented approach competitive results are possible also without using pre-trained weights. But when adapting off-the-shelf ConvNets for sound detection, starting with pre-trained weights can have some advantages. Training can be significantly faster and even lead to better detection performance, especially in cases where there is only a limited amount of audio data available. Nevertheless, if fine-tuning a network originally designed for image classification, re-training the entire network, not just the last layers is essential. When training the final classification layer of model M2 exclusively, using features from the penultimate layer, only 77 % AUC was obtained on the validation set.

Finally the results of this work show, sound detection benefits from fine-tuning DCNNs pre-trained on large amounts of image data, even if this data comes from a completely different domain.

5. ACKNOWLEDGMENT

I would like to thank Dan Stowell, Hervé Glotin, Yannis Stylianou and Mike Wood for organizing this challenge. I especially want to thank Elias Sprengel for the fruitful cooperation during the previous Bird Audio Detection challenge [19]. I also want to thank the Museum fuer Naturkunde Berlin and the Bundesministerium fuer Wirtschaft und Energie for supporting my research.

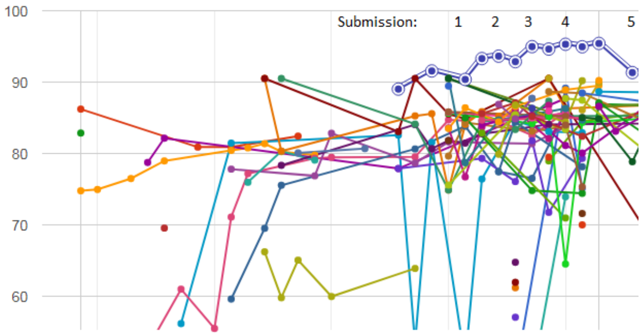

Figure 4: Public leaderboard of the DCASE 2018 Bird Audio Detection challenge. The above described methods and submissions belong to the highlighted and numbered blue dots.

6. REFERENCES

[1] http://dcase.community/challenge2018/

[2] Stowell D, Stylianou Y, Wood M, Pamuła H, Glotin H (2018) Automatic acoustic detection of birds through deep learning: the first Bird Audio Detection challenge. In: Methods in Ecology and Evolution

[3] Joly A, Goëau H, Botella C, Glotin H, Bonnet P, Planqué R, Vellinga WP, Müller H (2018) Overview of LifeCLEF 2018: a large-scale evaluation of species identification and recommendation algorithms in the era of AI, In: Proceedings of CLEF 2018

[4] Goëau H, Glotin H, Planqué R, Vellinga WP, Stefan K, Joly A (2018) Overview of BirdCLEF 2018: monophone vs. soundscape bird identification. In: CLEF working notes 2018

[5] Lasseck M (2018) Audio-based Bird Species Identification with Deep Convolutional Neural Networks. In: Working Notes of CLEF 2018 (Cross Language Evaluation Forum)

[6] Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) Imagenet: A largescale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition, 2009. pp. 248–255

[7] Paszke A, Gross S, Chintala S et al. (2017) Automatic differentiation in PyTorch. In: NIPS-W

[8] McFee B, McVicar M, Balke S et al. (2018) librosa/librosa: 0.6.0. Zenodo. http://doi.org/10.5281/zenodo.1174893

[9] Sprengel E, Jaggi M, Kilcher Y, Hofmann T (2016) Audio based bird species identification using deep learning techniques. In: Working notes of CLEF 2016

[10] Kahl S, Wilhelm-Stein T, Hussein H et al. (2017) Large-Scale Bird Sound Classification using Convolutional Neural Networks. In: Working Notes of CLEF 2017

[11] Fazekas B, Schindler A, Lidy T (2017) A Multi-modal Deep Neural Network approach to Bird-song Identification. In: Working Notes of CLEF 2017

[12] Grill T, Schlüter J (2017) Two Convolutional Neural Networks for Bird Detection in Audio Signals. In: 25th European Signal Processing Conference (EUSIPCO2017). Kos, Greece. https://doi.org/10.23919/EUSIPCO.2017.8081512

[13] Sevilla A, Glotin H (2017) Audio bird classification with inception-v4 extended with time and time-frequency attention mechanisms. In: Working Notes of CLEF 2017

[14] Fritzler A, Koitka S, Friedrich CM (2017) Recognizing Bird Species in Audio Files Using Transfer Learning. In: Working Notes of CLEF 2017

[15] Szegedy C, Vanhoucke V, Ioffe S (2015) Rethinking the Inception Architecture for Computer Vision. arXiv preprint arXiv:1512.00567

[16] He K, Zhang X, Ren S, Sun J (2016) Deep Residual Learning for Image Recognition. In: CVPR, 2016

[17] Russakovsky O, Deng J et al. (2015) ImageNet Large Scale Visual Recognition Challenge. IJCV, 2015

[18] http://dcase.community/challenge2018/task-bird-audio-detection-results

[19] Stowell D, Wood M, Stylianou Y, Glotin H (2016) Bird detection in audio: a survey and a challenge. arXiv preprint arXiv:1608.03417