Bird Species Identification in Soundscapes*

Mario Lasseck

Museum für Naturkunde Berlin, Germany
Mario.Lasseck@mfn.berlin

Abstract. This paper presents deep learning techniques for audio-based bird identification in soundscapes. Deep Convolutional Neural Networks are trained to classify 659 species. Different data augmentation techniques are applied to prevent overfitting and improve model accuracy and generalization. The proposed approach is evaluated in the BirdCLEF 2019 campaign and provides the best system to identify bird species in wildlife monitoring recordings. With an ensemble of different single- and multi-label classification models it obtains a classification mean average precision (c-mAP) of 35.6 % and a retrieval mean average precision (r-mAP) of 74.6 % on the official BirdCLEF test set. In terms of classification precision, single model performance surpasses previous state-of-the-art by more than 20 %.

Keywords: Bird Species Identification, Biodiversity Assessment, Soundscapes, Convolutional Neural Networks, Deep Learning, Data Augmentation

1 Introduction

For the LifeCLEF bird identification task participating teams have to identify different bird species in a large collection of audio recordings. The 2019 edition mainly focuses on soundscapes. This is a more difficult task compared to previous editions where species had to be identified mostly in mono-directional recordings with usually only one prominent species present in the foreground. Soundscapes on the other hand are recorded in the field, e.g. for wildlife monitoring, not targeting any specific direction or individual animal. There can be a large number of simultaneously singing species overlapping in time and frequency, arbitrary background noise depending on weather conditions and sometimes very distant and faint calls. Identifying as many species as possible in such a scenario remains challenging but is an important step towards real-world wildlife monitoring and reliable biodiversity assessment. An overview and further details about the BirdCLEF task is given in [1]. It is among others part of the LifeCLEF 2019 evaluation campaign [2].

The approach described in this paper uses neural networks and deep learning. It builds on the work of previous solutions to the task and combines proven techniques with new methods for data augmentation and multi-label classification.

2 Implementation

Data Preparation

All audio recordings are first high pass filtered at a frequency of 2 kHz (Q = 0.707) and then resampled to 22050 Hz with the Sound eXchange (SoX) v14.4.1 audio processing tool [3]. Soundscapes from the validation set are prepared for training by cutting them into individual files according to their annotations. Starting from the beginning of a file, whenever the label or set of labels changes, a new audio file is generated with the corresponding labels. Additionally, a "noise only" file is created from each soundscape by merging all parts without bird activity via concatenation. Those files containing only background noise are later used together with other background recordings for noise augmentation.

In order to also use the validation set for training, it is split into 8 folds via iterative stratification for multi-label data [4]. As a result, a small part of the validation set can be used to evaluate model performance while the rest of the set can be added to the Xeno-Canto [5] training set.

To allow faster prototyping and to create a more diverse set of models for later ensembling, different data subsets are formed targeting different numbers of species or sound classes:

- Data set 1: 78 classes (with 7342 files from the training set)
- Data set 2: 254 classes (with 21542 files from the training set)
- Data set 3: 659 classes (with all 50145 files from the training set)[1]

The smallest data set covers all 78 species present in the annotated soundscapes of the validation set and only contains training files belonging to those classes (not considering background species). The second data set consists of all files belonging to species mainly present in the recording locations of the United States. To find out which species are likely to be recorded in the US, the additionally provided eBird [6] data is taken into account and all files belonging to a species with a frequency value above zero for any time of the year are added to the first data set. The third data set finally covers all 659 species and all available training files. The eBird data is also used to create a list of unlikely species for both the Colombia and the US recording locations. For some submissions this list is later used to set predictions of unlikely species to zero for soundscapes in the test set depending on their recording location.

[1] 8 files of the Xeno-Canto training set are excluded because they are corrupt or too small

Training Setup

For audio-based bird species identification Deep Convolutional Neural Networks pre-trained on ImageNet [7] are fine-tuned with mel scaled spectrogram images representing short audio chunks. Models are trained with PyTorch [8] utilizing PySoundFile and LibROSA [9] 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 a duration of ca. 5 seconds
- Apply short-time Fourier transform
- Normalize and convert power spectrogram to decibel units (dB) via logarithm
- Convert linear spectrogram to mel spectrogram
- Remove low and high frequencies
- Resize spectrogram to fit input dimension of the network
- Convert grayscale image to RGB image

In each training epoch all training files are processed in random order to extract audio chunks at random position. Training is done with a batch size of ca. 100 - 200 samples using up to 3 GPUs (Nvidia 1080, 1080 Ti, Titan RTX). Categorical cross entropy [10] is used as loss function for single-label classification considering only foreground species as ground truth targets. Stochastic gradient descent is used as optimizer with momentum 0.9, weight decay 1e-4 and an initial learning rate of 0.1. Learning rate is decreased at least once during training by ca. 10^{-1} whenever performance on the validation set stops improving. If more than one species is assigned to an audio chunk, in case of validation soundscapes, one species or label is chosen randomly as ground truth target during training. Background species annotated for Xeno-Canto files are not taken into account.

Besides the common single-label classification approach, multi-label classification models are trained as well to take advantage of the fact that there are multi-label annotations existing for validation soundscapes with two or more species present at the same time in many cases. For soundscapes, the multi-label approach also seems the more suited classification method since recordings are mostly not focused on a single target species. Two loss functions are tested for multi-label training. PyTorch's MultiLabelSoftMarginLoss creates a criterion that optimizes a multi-label one-versus-all loss based on max-entropy [11]. The loss function BCEWithLogitsLoss combines a sigmoid layer and a binary cross entropy layer [12].

For the validation and test set audio chunks are extracted successively from each file with an overlap of 10 % for validation files during training and 80 % for files in the test set. Predictions are summarized for each file and time interval by taking the maximum over all chunks. For most submissions, different models are ensembled by averaging their predictions for each species after normalizing the entire prediction matrix to have a minimum of 0.0 and a maximum value of 1.0. To increase ensemble performance a little further, in some cases it helped to clip very low and high prediction values to -7.0 and 10.0, respectively before normalization.

Data Augmentation

To increase model performance and improve generalization to different recording conditions and habitats, the most effective data augmentation techniques from the previous BirdCLEF edition [13] are applied in both time and frequency domain. New methods are highlighted and explained below. The following methods are applied in time domain regarding audio chunks:

- Chunk extraction at random position in file
- Duration jitter
- **Local time stretching and pitch shifting**
- **Filter with random transfer function**
- Random cyclic shift
- Adding audio chunks from files containing only background noise
- Adding audio chunks from files belonging to the same bird species (single-label)
- **Adding audio chunks from files belonging to random bird species (multi-label)**
- Random signal amplitude of chunks before summation
- Time interval dropout

A few new methods are added for this year's challenge to augment individual audio chunks in time domain before mixing them together:

Local time stretching and pitch shifting in time domain. The audio signal is divided into segments, each having a randomly chosen duration between 0.5 and 4 seconds. To each segment time stretching or pitch shifting or both is applied individually using the LibROSA library. The time stretching factor is randomly chosen from a gauss distribution with a mean value of 1 and a standard deviation of 0.05. The pitch is shifted by an offset randomly chosen from a gauss distribution with a mean value of 0 and a standard deviation of 25 cents (8th of a tone).

Filter with random transfer function. With a chance of ca. 20 %, audio chunks are filtered in time domain using a butterworth filter design with variable transfer function. The following filter parameters are chosen randomly:

- Type: lowpass, highpass, bandpass, bandstop
- Order: 1-5
- Cutoff frequency: 1-22049 Hz

For bandpass and bandstop filter types the second (high) cutoff frequency is chosen between the (low) cutoff frequency + 1 and 22049 Hz (nyquist frequency - 1). Depending on filter parameters and audio input, filter stability is not always guarantied. To prevent unbounded signals the original input is passed as output if the filter output contains anything that is not a number between -1.0 and 1.0. Examples of a randomly filtered audio recording are visualized in Figure 1.

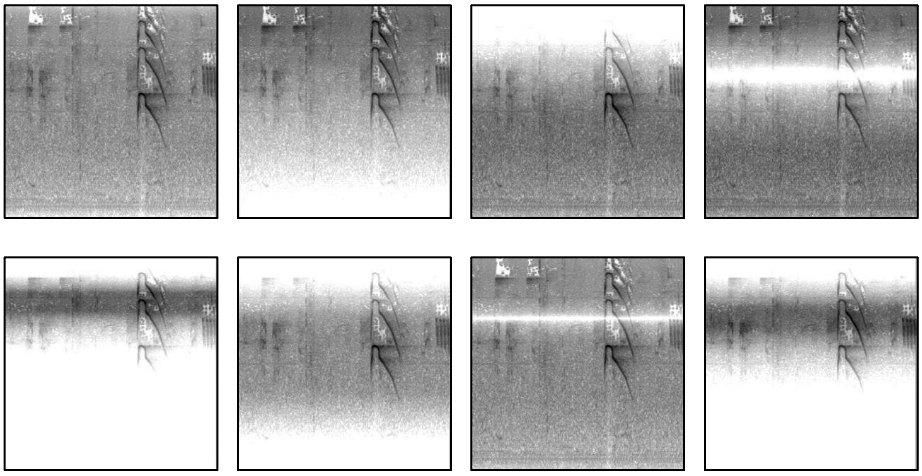

Fig. 1. Examples of a randomly filtered audio recording.

Mixing random audio chunks for multi-label classification. For multi-label classification, audio chunks from random files are mixed together and their corresponding labels added to the target label set during training. Up to four audio chunks are added with random signal amplitude to the original training sample with conditional probabilities of 50, 40, 30 and 20 %. A similar technique is originally used by [14] for image classification and has shown good results for multi-label audio classification as well [15]. Here, however, labels are not weighted by signal amplitudes (or influenced by weighting of the linear combination).

For background noise augmentation, besides using noise from validation files, recordings without bird activity of the Bird Audio Detection (BAD) task 2018 [16] are used. The BAD data set is part of the IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE) 2018 [17]. It consists of audio files from three separate bird sound monitoring projects each recorded under differing conditions regarding recording equipment and background sounds.

The audio chunk (or sum of chunks) is transformed to frequency domain via short-time Fourier transform with a window size of 1536 samples and a hop length of 360 samples. Frequencies are mel scaled with low and high frequencies removed resulting in a spectrogram with 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 [18] network or 224x224 pixel for ResNet [19] models. Resizing is performed with the Python Image Library fork Pillow using randomly chosen interpolation filters of different qualities. Because audio chunks are extracted with a random length (e.g. between 4.55 and 5.45 s by applying a duration jitter of ca. half a second) a global time stretching effect is obtained after resizing the variable length spectrogram images to a fixed width. Image resizing is also applied to individual vertical and horizontal spectrogram segments to accomplish piecewise or local time and frequency stretching (see below and [13] for more details). Since networks are pre-trained on RGB images, the grayscale image is copied to all three colour channels. Further augmentation is applied in frequency domain to the spectrogram image during training:

- Global frequency shifting/stretching
- Local time and frequency stretching
- Different interpolation filters for spectrogram resizing
- Colour jitter (brightness, contrast, saturation, hue)

Table 1 demonstrates the effect of augmentation and single- vs. multi-label training on model performance. All models were trained with the 78 classes data set using the same parameter settings with a learning rate of 0.1 and 0.01 until performance on the validation set stopped improving.

Table 1. Influence of data augmentation on model performance.

ID	Description	InceptionV3 c-mAP [%]	ResNet-152 c-mAP [%]
E1	Baseline	26.5	21.1
E2	E1 with BAD noise augmentation	40.1	38.3
E3	E2 with validation files for training	42.5	38.7
E4	E3 with validation noise augmentation	50.9	51.3
E5	E4 with 2019 augmentation methods	53.1	52.2
E6	E4 with multi-label training[2]	49.7	52.9
E7	E6 with 2019 augmentations (post challenge result)	50.1	54.7

More details on individual augmentation methods and their effect on identification and detection performance in previous challenges can be found in [13] and [20].

3 Results

For the first two submitted runs a single model was used to predict the species for each file and time interval in the test set. All other runs used an ensemble of different models. The main properties of individual models are listed in Table 2. Selected results on the official BirdCLEF test set are summarized in Table 3 and further described in the next section.

[2] Loss function: torch.nn.MultiLabelSoftMarginLoss

Table 2. Main properties of models used for submitted runs.

Model ID	M1	M2	M3	M4	M5	M6	M7	M8
Included in run	1	2-10	6-10	3,4	5-10	5-10	4-10	6-10
Number of classes	659	659	254	254	254	254	78	78
Network type	Incept.	Incept.	Incept.	ResNet	ResNet	ResNet	ResNet	ResNet
Input size [pixel]	299	299	299	224	224	224	224	224
Chunk duration [s]	5	5	5	5	5	2.6	5	5
Label type	single	single	single	single	single	single	multi	multi
Loss function	Cross entropy	Cross entropy	Cross entropy	Cross entropy	Cross entropy	Cross entropy	Multi Label	BCEWit hLogits
BAD data used	no	yes	yes	yes	yes	yes	yes	yes
Val. data (labelled)	no	yes	yes	yes	yes	yes	yes	yes
Val. data (noise)	no	yes	yes	yes	yes	yes	yes	yes
New augmentations	no	no	no	no	yes	no	no	no
Val. set c-mAP [%]	29.7	49.1	51.3	51.2	53.9	52.1	53.8	49.1
Test set c-mAP [%]	21.3	25.9	-	-	-	-	-	-

Run 1. In order to better compare results and to find out if and how much progress was made on identification performance since last year, the best performing model of the 2018 BirdCLEF edition [13] was retrained on this year's data set. All files from the Xeno-Canto training set but no validation soundscapes were used for training. The model obtained a classification mAP of 21.3 % and a retrieval mAP of 44.7 % on the test set.

Run 2. For the second run, a single model (M2) was trained with main properties listed in Table 2. Also validation soundscapes were used for training and noise augmentation. The third generation Inception model M2 used all above mentioned augmentations except the new time domain methods filtering and local time stretching and pitch shifting. For this and all following models, BAD 2018 files were used for noise augmentation. As a result, it is not necessary any more to segment training files into signal and noise parts to get background material from the training set for noise augmentation like done in previous challenge editions. This greatly simplifies the preprocessing step. A c-mAP of 25.9 % and a r-mAP of 69.1 % is obtained on the test set resulting in a performance increase of 21.6 % and 54.6 %, respectively compared to previous state-of-the-art (M1). Since M1 and M2 didn't use the exact same training set and for the training of M2 not all new time domain augmentations were applied the given progress is only a rough approximation.

Run 3. For the third run, two models were ensembled: the 2nd run model and a ResNet-152 model trained on the 254 classes training set. For this and the following runs, predictions of unlikely species regarding recording location were set to zero.

Run 4. The 4th run used the ensemble of run 3 plus an additional multi-label classification model (M7) trained on the 78 classes set. This 3 model ensemble obtained the highest retrieval mAP of 74.6 % on the test set.

Run 5. The ensemble of run 5 consists of all previous models (except M1) plus an additional 254 classes ResNet-152 model (M6) yielding a higher temporal resolution of spectrogram image inputs. It mainly differs in the following parameters:

- FFT size: 512 (instead of 1536) samples
- FFT hop length: 256 (instead of 360) samples
- Chunk duration: 2.6 (instead of 5.0) seconds
- Duration jitter: 0.2 (instead of 0.45) seconds
- Number of mel bands: 155 (instead of 310)
- Start frequency: 0 (instead of 160) Hz
- End frequency: 11025 (instead of 10300) Hz
- Local time stretch chance: 40 (instead of 50) %
- Local time stretch factor min.: 0.95 (instead of 0.9)
- Local time stretch factor max.: 1.05 instead of 1.1)

The run 5 ensemble obtained the highest classification mAP of 35.6 % on the test set.

Run 6 to 10. Different combinations of the previously mentioned models were used for run 6 to 10. Also different snapshots of the same model were included for ensembling and two models were trained on different folds of the validation set. Nevertheless, no further progress on identification performance on the test set was obtained. For run 9 the same ensemble as for run 8 was used except run 9 didn't use the eBird data to set predictions of unlikely species to zero in the post-processing step. This demonstrates once again, performance can be increased when unlikely birds are filtered out for a certain recording location where species composition is known in advance.

Table 3. Official scores on the BirdCLEF 2019 test set (for selected runs).

Run	#Models	#Snap- shots	c-mAP [%] Val. set	c-mAP [%] Test set	r-mAP [%] Test set
1	1	1	29.7	21.3	44.7
2	1	1	49.1	25.9	69.1
3	2	2	57.5	29.7	71.0
4	3	3	62.0	30.9	**74.6**
5	4	4	63.7	**35.6**	71.5
7	7	9	64.9	34.9	73.2
8	9	12	-	35.1	74.4
9	9	12	-	32.8	71.1
10	all	all	-	35.5	72.2

4 Discussion

The 2019 BirdCLEF edition had a clear focus on identifying birds in soundscapes originating from real-world wildlife monitoring recordings. Although this was a much more difficult task compared to previous editions, progress in model performance was obtained by exploring new augmentation techniques and by combining different single- and multi-label classification models.

A large performance increase was obtained by adding random background noise from other and/or similar habitats. A very good source for noise augmentation is the data set of the DCASE 2018 Bird Audio Detection challenge (E1 vs. E2 in Table 1). It is easily available and published under the Creative Commons Attribution licence CC-BY 4.0 [17]. The BAD recordings cover a wide range of background noise and atmosphere from a diverse set of different monitoring scenarios and are therefore well suited to improve model generalization. On the other hand, in cases where the target monitoring location is known in advance, a model can specifically be designed for a certain habitat and greatly benefit by using background sounds of this particular recording location for noise augmentation during training (E3 vs. E4 in Table 1).

With additional methods like filtering audio chunks with random transfer function or applying local time stretching and pitch shifting in time domain, identification performance can be further increased (E4 vs. E5 & E6 vs. E7 in Table 1). Unfortunately, training takes significantly longer especially if LibROSA's time stretching and pitch shifting algorithms are applied very frequently. Due to the longer training time it was not possible to investigate the individual influence of each method or different parameter settings on model performance. To save time, those techniques were only applied to the original training sample (first audio chunk in the mix) and not, or only with very low chance, to chunks added for augmentation. Both algorithms also seem to blur the resulting spectrogram even with very subtle use (time stretching factor close to 1, pitch shifting offset close to 0). Maybe a more efficient implementation regarding processing time and quality would be a better choice in the future.

Multi-label training was successfully applied for the 78 classes set (E4 vs. E6 in Table 1). In contrast to the single-label approach, for multi-label classification the residual network obtained better results compared to the Inception architecture (3rd vs. 4th column in Table 1). Unfortunately, training with a larger number of classes didn't work so well even when passing a weight vector as argument to the BCEWithLogitsLoss function to compensate for class imbalances. One explanation for this might be the exponential growth of possible label combinations depending on the number of individual labels (classes) and the number of labels considered to be possible for a single audio chunk. Maybe if species combination constrains are known a priori (e.g. by distinguishing between diurnal and nocturnal birds) and applied to reduce the number of possible label sets, models can also be trained successfully in a multi-label fashion for a much larger number of species.

To reproduce results and to provide a baseline for future BirdCLEF challenges and further research on bird species identification, source code will be made available at: www.animalsoundarchive.org/RefSys/BirdCLEF2019.

Acknowledgments. I would like to thank Stefan Kahl, Alexis Joly, Hervé Goëau, Willem-Pier Vellinga and Hervé Glotin for organising this task, Xeno-Canto for providing the training data and the Cornell Lab of Ornithology and Paula Caycedo for providing and annotating the soundscapes. I also want to thank the Museum für Naturkunde for supporting my research.

References

1. Kahl S, Stöter FR, Glotin H, Planqué R, Vellinga WP, Joly A (2019) Overview of BirdCLEF 2019: large-scale bird recognition in soundscapes. In: CLEF working notes 2019
2. Joly A, Goëau H, Botella C, Kahl S, Servajean M, Glotin H, Bonnet P, Vellinga WP, Planqué R, Stöter FR, Müller H (2019) Overview of LifeCLEF 2019: Identification of Amazonian Plants, South & North American Birds, and Niche Prediction. In: Proceedings of CLEF 2019
3. SoX Homepage, http://sox.sourceforge.net/, last accessed 2019/06/19
4. Sechidis K, Tsoumakas G, Vlahavas I (2011) On the Stratification of Multi-Label Data. In: Gunopulos D, Hofmann T, Malerba D, Vazirgiannis M (eds) Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2011. Lecture Notes in Computer Science, vol 6913. Springer, Berlin, Heidelberg.
5. Xeno-Canto Homepage, https://www.xeno-canto.org/, last accessed 2019/06/19
6. eBird Homepage, https://ebird.org/home, last accessed 2019/06/19
7. Deng J et al. (2009) Imagenet: A largescale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition, 2009. pp. 248–255
8. Paszke A et al. (2017) Automatic differentiation in PyTorch. In: NIPS-W
9. McFee B, McVicar M, Balke S et al. (2019) librosa/librosa: 0.6.3. Zenodo. https://doi.org/10.5281/zenodo.2564164
10. https://pytorch.org/docs/stable/nn.html#crossentropyloss, last accessed 2019/06/19
11. https://pytorch.org/docs/stable/nn.html#multilabelsoftmarginloss, last accessed 2019/06/19
12. https://pytorch.org/docs/stable/nn.html#bcewithlogitsloss, last accessed 2019/06/19
13. Lasseck M (2018) Audio-based Bird Species Identification with Deep Convolutional Neural Networks. In: Working notes of CLEF 2018
14. Zhang H, Cisse M, Dauphin YN, Lopez-Paz D (2018) mixup: Beyond empirical risk minimization. In: International Conference on Learning Representations, 2018
15. Xu K, Feng D, Mi H et. al (2018) Mixup-Based Acoustic Scene Classification Using Multi-Channel Convolutional Neural Network. In: arXiv:1805.07319
16. 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
17. http://dcase.community/challenge2018/task-bird-audio-detection, last accessed 2019/06/19
18. Szegedy C, Vanhoucke V, Ioffe S (2015) Rethinking the Inception Architecture for Computer Vision. arXiv preprint arXiv:1512.00567
19. He K, Zhang X, Ren S, Sun J (2016) Deep Residual Learning for Image Recognition. In: CVPR, 2016
20. Lasseck M (2018) Acoustic Bird Detection with Deep Convolutional Neural Networks. In: Plumbley MD et al. (eds) Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018), pp. 143-147, Tampere University of Technology.