

# Learning to Monitor Birdcalls From Weakly-Labeled Focused Recordings

Jan Schlüter<sup>1</sup>

<sup>1</sup>*Institute of Computational Perception, Johannes Kepler University Linz, Austria*

## Abstract

Passive acoustic monitoring can support biodiversity assessments, but requires automated analysis to be affordable at scale, which in turn requires labeled training data. Obtaining labeled data for each deployed device or location is expensive. The BirdCLEF 2021 scientific challenge tasked participants to train models on freely available weakly-labeled recordings of individual birds from xeno-canto, and apply them to recordings of passive devices. The ensemble of Convolutional Neural Networks (CNNs) described in this work achieved an F-Score of 0.672 across six recording locations, the twelfth best entry among 816 teams.

## Keywords

birdcall identification, soundscapes, domain mismatch, deep learning

## 1. Introduction

For some species, such as birds, passive acoustic monitoring is an interesting option to assess the status and trends of populations in an area. Detecting animal vocalizations and identifying the species in audio recordings is labor-intensive, prompting research for automated solutions to analyze recordings of monitoring devices. Many such solutions are based on machine learning, which requires fitting a prediction model to labeled recordings. Current prediction models are sensitive to recording conditions [1] and benefit from being constrained to the set of species to be expected. For ideal performance, the labeled recordings should thus be prepared with the same recording device and at the same location that is to be monitored later using the model. Since labeling recordings is labor-intensive, this represents a big hurdle for deployments at new locations.

For some species, such as birds, there are freely available online databases of audio recordings, such as the Macaulay Library [2], Tierstimmenarchiv [3], or xeno-canto [4]. The BirdCLEF 2021 scientific challenge [5], a part of the LifeCLEF initiative [6], explored tapping into this resource for training prediction models for passive acoustic monitoring of birds. Specifically, participants were provided with 62 874 recordings of 397 bird species from xeno-canto as well as 20 labeled recordings of passive devices from two locations, and tasked to detect and identify birds in 5-second intervals on 80 passive recordings

---


*CLEF 2021 – Conference and Labs of the Evaluation Forum, September 21–24, 2021, Bucharest, Romania*

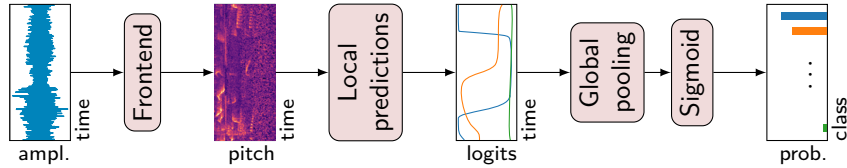
✉ jan.schluter@jku.at (J. Schlüter)

ORCID 0000-0003-3862-6888 (J. Schlüter)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)



**Figure 1:** Outline of the prediction model architecture.

from six locations. For all recordings, the geographic location and recording date and time are known.

This scenario is highly interesting in practice: if prediction models achieve satisfying accuracy in this setting, they can be deployed to new locations without requiring matching training data. Solving this task poses two major challenges:

- A. Most recordings from xeno-canto are focused recordings intended to capture the vocalizations of a particular bird, often done with directional microphones. In contrast, the recordings to predict on were done unattended and with omnidirectional microphones (sometimes referred to as “soundscapes” [5]). This creates a strong domain mismatch between training and test recordings.
- B. The xeno-canto recordings are weakly-labeled: There is no timing information regarding bird vocalizations, and only the species intended to be recorded is known for sure, other species occurring in the background may be labeled or omitted. In contrast, predictions on unattended recordings are to be done in 5-second intervals and include all audible species.

In this work, we attempt to tackle the first challenge with preprocessing and data augmentation, and the second one with multiple-instance learning and a two-level inference procedure.

The following section details our prediction models, followed by the training procedure in Section 3 and inference in Section 4. Section 5 describes our experimental setup and compares results both of single models and ensembles on a validation set and the official test set. Section 6 discusses ideas that did not work, and Section 7 concludes the paper.

## 2. Models

The general model architecture is depicted in Figure 1: From an (arbitrarily long) monophonic raw audio recording, a frontend computes a spectrogram-like representation. A Fully-Convolutional Network (FCN) processes this representation into a time series of logits for every class. When passed through a sigmoid, these would give us local predictions at every time step. Since we do not have temporally accurate labels to train these, only recording-wise labels, we apply a global pooling operation (over time) to obtain a single logit per class. Passed through a sigmoid, these serve as our recording-wise predictions.

There are several options for the frontend, the local predictor and the global pooling operation, which we will look at in the following sections.

## 2.1. Frontend

The frontend processes monophonic audio recordings of sample rate 32 kHz (this is the rate the unattended recordings are done at, and all training recordings were resampled to). It consists of the following operations applied in sequence:

**A Short-Time Fourier Transform (STFT)** computes a linear spectrogram. For two of the predictors, windows are 1486 samples long and start every 457 samples, resulting in 70 frames per second. For two pretrained predictors, the window and hop size is 1024 and 320, respectively, giving 100 frames per second. In both cases, Hann windowing is used and only magnitudes are kept.

**A mel filterbank** transforms spectrograms to mel spectrograms. Depending on the predictor, it has 80 bands from 27.5 Hz to 10 kHz, 80 bands from 27.5 Hz to 15 kHz, 64 bands from 50 Hz to 8 kHz, or 64 bands from 50 Hz to 14 kHz.

**A pointwise nonlinearity** compresses magnitude values by passing them through either  $y = \log(1 + 10^a x)$  or  $y = x^{\sigma(a)}$ , where  $\sigma(a) = 1/(1 + \exp(-a))$  denotes the logistic sigmoid,  $a$  is initialized to zero and learned by backpropagation.

**Denosing:** The recordings have very different background noise floors, both due to the environment and recording equipment. To help generalization, recordings are preprocessed by subtracting the median over time from each frequency band (separately for each recording or excerpt). For pretrained models, this step is either skipped, or a global offset is added after median subtraction to retain the maximum input value (otherwise the value range would not match what the models were trained on).

**Normalization:** To ensure inputs are in a suitable range (even when the magnitude compression changes during training), each frequency band is normalized over the batch and time dimension with Batch Normalization [7].

## 2.2. Local predictions

The purpose of the local predictor is to take the spectrogram computed by the frontend, and produce 397 time series of logits, one for each bird species. Regarding the input as an 80 pixel (or 64 pixel) high one-channel image, and the output as a 1 pixel high 397-channel image, the predictor should consist of a series of convolutions and pooling operations that reduces the image height to 1 pixel and produces 397 channels. We used four different such local predictors:

**Vanilla ConvNet:** An 8-layer network described in [8, p.3], trained on 70 frames per second, 80 mel bands spectrograms. It has a receptive field of  $79 \times 103$  (79 frequency bands, 103 spectrogram frames, about 1.5 seconds), a temporal stride of 9 spectrogram frames (7.778 predictions per second), and 3.74 million parameters. Compared to [8], it uses group normalization [9] with 16 groups instead of batch normalization.

**ResNet:** A 14-layer residual network with pre-activations described in [8, p.4], also trained on 70 frames per second, 80 mel bands spectrograms. Its receptive field is  $80 \times 119$ , temporal stride is 9 frames, and it has 3.87 million parameters. Compared to [8], it uses group normalization with 16 groups instead of batch normalization, and crops the shortcut connections instead of padding the convolutions.

**Cnn14:** A 16-layer network described in [10], pretrained on AudioSet [11], trained on 100 frames per second, 64 mel bands spectrograms (from 50 Hz to 8 kHz) with  $\log(1 + 10^a x)$  magnitudes initialized to  $a = 5$ . The pretrained network’s global max + average pooling and final two layers are replaced with two  $1 \times 1$  convolutions of 1024 and 397 channels, respectively (with batch normalization and leaky rectification in between, and both convolutions preceded by 50% dropout [12]). It has a receptive field of  $284 \times 284$  (2.84 seconds), stride of 32 frames (3.125 predictions per second), and 77.98 million parameters. The size of the receptive field exceeds the number of mel bands by far; this is possible because all convolutions are zero-padded.

**ResNet38 :** A 38-layer residual network described in [10], also pretrained on AudioSet, trained on 100 frames per second, 64 mel bands spectrograms (from 50 Hz to 14 kHz). The pretrained network’s final layers are replaced as described for Cnn14. Its receptive field is  $2997 \times 2997$  (30 seconds), stride is 256 frames (one prediction every 2.56 seconds), and it has 71.01 million trainable parameters.

We optionally use 8-fold multi-sample dropout [13], implemented by replicating the inputs and targets before the second to last dropout layer.

### 2.3. Global pooling

Up to here, the model produces a time series of logits for each class. The final step is to pool these logits into a single prediction per class for the full recording, such that we can compute (and minimize) the classification error wrt. the given global labels for the recording.

Global average pooling would distribute the gradient of the loss uniformly over all time steps, training the network to predict a labeled bird everywhere in the recording. Global max pooling would route the gradient only to the most confident detection of each species. As a compromise, log-mean-exp pooling with  $\frac{1}{a} \log \left( \frac{1}{T} \sum_{t=1}^T \exp(ax_t) \right)$  [14, Eq. 6] allows to interpolate between taking the maximum ( $a \rightarrow \infty$ ) and mean ( $a \rightarrow 0$ ). With  $a = 1$ , the output depends on the largest few values, which is also where the gradient is distributed to. This setting was used for most models. Some models updated  $a$  with backpropagation, possibly using a separate  $a$  per species (since some species might vocalize densely, warranting a small  $a$ , and others sparsely, requiring a large  $a$ ).

## 3. Training

The training procedure considers both challenges explained in Section 1: weak labels and domain mismatch.

### 3.1. Excerpts

Ideally, the model would be trained on complete xeno-canto recordings – this is the only way we can be sure all weak labels are correct. If we pick a random excerpt, it is not guaranteed that all birds annotated to be present in the recording are also audible in the chosen excerpt. However, the longest recordings are 3 hours, which is impractical. As in [8], we train on randomly selected 30-second excerpts instead, hoping that most annotated birds will be audible at least once. Too short files are looped to make up 30 seconds. Validation uses the central 30 seconds of a recording.

### 3.2. Augmentation

To help the model generalize, especially in the light of the domain mismatch from focused training recordings to unattended test recordings, we employ several data augmentation strategies:

**Adding noise:** To make the models work under low signal-to-noise ratios (i.e., the conditions found in the unattended recordings), we mix them with excerpts from the Chernobyl BiVA [15] and BirdVox-full-night [16] datasets. These datasets are precisely annotated with bird occurrences, so we can extract all parts void of birds. We started out carefully, but the best setting turned out to be mixing every training example with background noise, drawing a value  $p \in [0, 1)$  and scaling the noise with  $p$  and the bird recording excerpt with  $1 - p$ . For some models, we also set  $p = 1$  with 1% or 0.1% probability, setting the labels to all zero in this case. And for some models, the chosen noise excerpt is scaled to a maximum absolute value in  $[0, 1)$  before mixing it.

**Random downmixing:** The model is trained on monophonic recordings, but xeno-canto usually has stereo recordings.<sup>1</sup> We downmix them during training with a randomly uniform weight  $p$  for the left channel, and  $1 - p$  for the right channel.

**Filterbank pitch shifting:** Some models are trained with pitch shifting, cheaply implemented by modifying the mel filterbank: Instead of precomputing the filterbank, it is constructed on-the-fly, scaling the minimum and maximum frequency by the same random factor chosen uniformly between 0.95 and 1.05. A separate filterbank is constructed for every example in a minibatch (and applied to the minibatch with a batched matrix multiplication).

**Magnitude warping:** Drawing on an idea of Vladislav Kramarenko<sup>2</sup>, in the  $\log(1 + 10^a x)$  magnitude transformation, for some models, we scaled  $a$  by a random factor between 0.5 and 1.5 during training, shifting the result such that the maximum output value matches the unmodified  $a$  (to not drastically change the value range).

---

<sup>1</sup>For BirdCLEF 2021, only downmixed xeno-canto recordings were provided; we partly replaced them with the original stereo files.

<sup>2</sup><https://www.kaggle.com/c/birdsong-recognition/discussion/183269>

**High-frequency damping:** Also drawing on an idea of Vladislav Kramarenko, for some models, we lowered up to 50% of the high-frequency part of the spectrum by up to 50% magnitude, with a linear fade to the lowest frequency bin of the spectrum (the only bin not damped at all). His reasoning was that in the unattended recordings, birds are often farther away than in focused recordings, and high frequencies are damped more strongly with distance (which is indeed the case in forests [17]). We only apply this damping to models without median subtraction – it is applied before compressing magnitudes, so for models with approximately logarithmic magnitudes ( $\log(1 + 10^5x)$ ) and median subtraction its effect would be canceled.

### 3.3. Optimization

To be able to monitor training progress, we split off 10% of the xeno-canto recordings as a validation set, ensuring that no recordist is part of both the training and validation set (as they tend to visit the same locations using the same equipment).

Optimization uses ADAM with mini-batches of 16 examples, an initial learning rate of  $10^{-3}$ ,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ ,  $\epsilon = 10^{-8}$ , minimizing binary cross-entropy against all species (for a single model, we reduced targets for background species to 0.6). The validation loss is computed every 1000 update steps. If it does not improve over the current best value for 10 such evaluations in a row, the learning rate is reduced to a tenth, and training is continued. Training is stopped when the learning rate reaches  $10^{-6}$ . For the two pretrained models, we tried reducing the learning rate for the pretrained layers to 1% or 10% compared to the novel layers or to freeze the pretrained layers for some time, but it turned out that using the full learning rate for all the layers from the start works best.

## 4. Inference

While the model is trained on 30-second excerpts, at test time, it needs to predict the set of birds for a 10-minute recording in non-overlapping 5-second windows. We perform this in two steps: establishing a set of species present in the 10-minute recording, and detecting these species in 5-second windows. This allows to tune a separate threshold for the second step, to reduce false negatives without impacting false positives too much.

### 4.1. Species set

The way the model is constructed (Section 2), it can be applied to arbitrarily long recordings. We could thus apply it directly to a 10-minute recording to obtain a set of species. However, the employed global pooling method (log-mean-exp pooling) includes a division by the input length. If a bird appears only in the first two minutes of a recording, its pooled prediction is lower than if it appeared throughout the ten minutes. We could thus not distinguish low-confidence detections from high-confidence detections. Our solution is to apply the model to 30-second windows overlapping by 50% with a threshold of 0.5 and taking the union of all detections (or, equivalently, taking the maximum over the prediction windows and applying the threshold afterwards).

## 4.2. Windowed detections

Again, the way the model is constructed, we can apply it directly to 5-second windows even when it was trained on 30-second excerpts. Since log-mean-exp pooling is dependent on the input length, we need to adjust the threshold to make up for the mismatch. As we have established a set of bird species in the previous step, we can afford to set the threshold very low, removing detections that are outside the established set. Optimized for a single model on the 20 unattended recordings available for training, we found a threshold of 0.18 to be optimal; when optimizing an ensemble of 18 models, the optimal threshold was 0.08.

As an alternative, we can opt to use max pooling, not distinguishing single detections from repeated detections within a 5-second window and relying on the established set of species to filter out false positives. For the same 18-model ensemble, the optimal threshold on the 20 unattended recording using max pooling is 0.55.

## 4.3. Ensembling

To combine results from multiple models, we average their predicted logits directly after pooling, and apply thresholds afterwards. Applying thresholds first and combining models by vote counts per species performed worse.

## 4.4. Implementation

For improved performance, instead of splitting up the audio recording into overlapping 30-second and non-overlapping 5-second windows and applying the models to each excerpt, we apply the model to the full 10-minute recording up to the global pooling. Using information on the model’s receptive field size, padding and stride, we compute a timestamp for every prediction time step. This way we can extract windows from the series of logits and pool them as needed. For an 18-model ensemble, inference takes 6 seconds for a 10-minute file on an NVIDIA GTX 1080 Ti. In addition to improving computational efficiency, this method also limits potential artifacts from zero-padding the audio excerpts in the two pretrained models.

## 5. Experiments

As mentioned in Section 3.3, we formed a train/validation split, grouped by recordist, using about 90% of the 62 874 recordings for training and the remainder for validation. We use the 20 provided unattended recordings (which are labeled in 5-second windows) as additional validation files, separated into the two recording locations Costa Rica (COR) and Sapsucker Woods (SSW).

Evaluation is done in terms of F-score: The numbers of true positives, false positives and false negatives are determined and added up across all 397 species as well as a “nocall” class for 5-second windows without any audible bird vocalization. From the total numbers, precision, recall and F-score are computed.

We trained several combinations of frontend, predictor, global pooling, and data augmentation and evaluated them both on the xeno-canto recordings left for validation (using the central 30 seconds only), and the 20 unattended recordings (using the full inference procedure of Section 4 with a second-level threshold of 0.18). Table 1, columns “F-score” show the results. The variations were chosen in search of good models for the challenge, not for evaluating the effect of any particular measure. Thus, we can only draw limited conclusions from these results:

- Unsurprisingly, results vary when repeating an experiment (rows 5 and 6 differ by up to 0.01 F-score with the same hyperparameters). Comparing single experiments with close results is thus meaningless.
- Comparing the predictors, on the xeno-canto recordings, Cnn14 performs best, followed by ResNet38, the small ResNet, and Vanilla ConvNet.
- On the unattended recordings (from locations COR and SSW), the order is the same, except that ResNet38 performs much worse than all others. A possible explanation is its large receptive field and large stride, which may make the series of logits too inaccurate for 5-second window predictions.
- None of the augmentations (except for adding noise) has a clear positive or negative effect when tested in isolation. It is possible that the augmentations diversify the set of models in a way that is helpful for ensembling.

For the challenge, we formed ensembles of multiple models. An initial ensemble of 5 models was picked by hand (Table 1, column “5-model ensemble”). It performs better than all Cnn14 models combined, even when adding all small ResNets, and all Vanilla ConvNets, and achieved an F-score of 0.667 on the official challenge data. Finally, a slightly better ensemble of 18 models was found in three trials of starting from an all-model ensemble (except ResNet38) and greedily removing randomly chosen single models until no improvement on SSW and COR combined was observed. It obtained an F-score of 0.672 on the challenge dataset, the 12th best entry. After observing that the second inference stage performs better using max pooling (Section 4.2), another three trials found an 11-model ensemble slightly improving on this with an F-score of 0.676 submitted after the end of the challenge.

## 6. Failed ideas

For each xeno-canto recording and for each unattended recording (both in the training and test set), the geographic location is known. It would thus be possible to focus on predicting those species that are likely to be present at each recording site. Exploring the data, we saw that most species present in the 20 labeled recordings from COR and SSW have also been uploaded to xeno-canto within short distance of the recording site, and only few species have not been uploaded within 60 km of the site. We tried to make use of this in three ways:

1. Computing the set of species uploaded within 60 km of sites COR and SSW results in reduced lists of 133 and 89 species, respectively. We filtered the predictions of a



397-species model using these site-specific lists, but it did not improve results on the 20 recordings.

2. Using the reduced species lists, we trained site-specific prediction models. Both for single models and for a 5-model ensemble, this worked worse than the generic 397-species models.
3. We weighted xeno-canto recordings by their distance to a particular site during training, giving close recordings higher importance in the loss function. We computed these weights as  $1/\sqrt{1 + (d/500)^k}$ , where  $d$  is the distance in kilometers, setting  $k = 2$  for a softer and  $k = 3$  for a harder distance dependence (inspired by the shape of a Butterworth filter response). These site-specific models performed comparable to unspecific models, not warranting the extra effort.

## 7. Conclusions

We obtained competitive results in the BirdCLEF 2021 challenge following a common recipe for such competitions: Training a lot of prediction models with varying settings, then blending them into an ensemble using some means of automatic model selection.

For practical purposes, we deem the following to be important: (1) Training on long enough excerpts to increase the chance that the weak labels are correct, (2) including a form of global pooling in the prediction model that encourages proper credit assignment to local predictions, (3) augmenting data by adding noise, (4) performing inference on two timescales to filter short-term detections using long-term information, (5) using pretrained audio models as a basis.

While these aspects were followed in our work, each offers room for improvement. For example, the global log-mean-exp pooling is oblivious to the typical frequency of calls. Even with a trainable  $a$  hyperparameter per species, it is forced to use the same pooling for different call types. The noise used for augmentation in our work is limited in diversity, as it stems from only two locations. Extending it requires manual screening of data for the absence of birds, or a highly accurate bird detector. The pretrained Cnn14 performs comparably well, but has an overly large receptive field in frequency dimension, and high computational demands. Another interesting route for future work is to make more use of long-term temporal structure. The two-stage inference procedure only captures the assumption that a particular bird (or bird species) will be audible multiple times during a 10-minute recording, but some bird calls have more complex regularities that could help detecting or distinguishing them.

## Acknowledgments

First of all, thanks a lot to the organizers of BirdCLEF 2021: Stefan Kahl, Tom Denton, Holger Klinck, Hervé Glotin, Hervé Goëau, Willem-Pier Vellinga, Robert Planqué and Alexis Joly. Apart from that, thanks to Paul Primus for suggesting the datasets used for background noises, and to Khaled Koutini for the idea of using a batched matrix multiplication to allow different pitch shifts for each minibatch item.

## References

- [1] T. Heittola, A. Mesaros, T. Virtanen, Acoustic scene classification in DCASE 2020 challenge: generalization across devices and low complexity solutions, in: Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020), 2020. URL: <https://arxiv.org/abs/2005.14623>.
- [2] Macaulay library, 2003. URL: <https://www.macaulaylibrary.org>, accessed: 2021-07-02.
- [3] Tierstimmenarchiv, 1951. URL: <https://www.tierstimmenarchiv.de>, accessed: 2021-07-02.
- [4] xeno-canto, 2005. URL: <https://www.xeno-canto.org>, accessed: 2021-07-02.
- [5] S. Kahl, T. Denton, H. Klinck, H. Glotin, H. Goëau, W.-P. Vellinga, R. Planqué, A. Joly, Overview of BirdCLEF 2021: Bird call identification in soundscape recordings, in: Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, 2021.
- [6] A. Joly, H. Goëau, S. Kahl, L. Picek, T. Lorieul, E. Cole, B. Deneu, M. Servajean, R. Ruiz De Castañeda, I. Bolon, H. Glotin, R. Planqué, W.-P. Vellinga, A. Dorso, H. Klinck, T. Denton, I. Eggel, P. Bonnet, H. Müller, Overview of LifeCLEF 2021: a system-oriented evaluation of automated species identification and species distribution prediction, in: Proceedings of the Twelfth International Conference of the CLEF Association (CLEF 2021), 2021.
- [7] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: Proceedings of the 32nd International Conference on Machine Learning (ICML), volume 37 of *Proceedings of Machine Learning Research*, PMLR, Lille, France, 2015, pp. 448–456. URL: <http://proceedings.mlr.press/v37/ioffe15.html>.
- [8] J. Schlüter, Bird identification from timestamped, geotagged audio recordings, in: Working Notes of CLEF, Avignon, France, 2018.
- [9] Y. Wu, K. He, Group normalization, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018.
- [10] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, M. D. Plumbley, Panns: Large-scale pretrained audio neural networks for audio pattern recognition, *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 28 (2020) 2880–2894. doi:10.1109/TASLP.2020.3030497.
- [11] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, M. Ritter, Audio set: An ontology and human-labeled dataset for audio events, in: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017, pp. 776–780. doi:10.1109/ICASSP.2017.7952261.
- [12] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R. R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors, arXiv e-prints abs/1207.0580 (2012). URL: <http://arxiv.org/abs/1207.0580>.
- [13] H. Inoue, Multi-sample dropout for accelerated training and better generalization, *CoRR* abs/1905.09788 (2019). URL: <http://arxiv.org/abs/1905.09788>. arXiv:1905.09788.

- [14] P. O. Pinheiro, R. Collobert, From image-level to pixel-level labeling with convolutional networks, in: Proceedings of the 28th IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015, pp. 1713–1721.
- [15] P. Kendrick, L. Barçante, N. Beresford, S. Gashchak, M. Wood, Bird vocalisation activity (biva) database: annotated soundscapes from the chernobyl exclusion zone, 2018. URL: <https://doi.org/10.5285/be5639e9-75e9-4aa3-afdd-65ba80352591>. doi:10.5285/be5639e9-75e9-4aa3-afdd-65ba80352591.
- [16] V. Lostanlen, J. Salamon, A. Farnsworth, S. Kelling, J. P. Bello, Birdvox-full-night: A dataset and benchmark for avian flight call detection, in: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2018, pp. 266–270. URL: <https://doi.org/10.1109/ICASSP.2018.8461410>. doi:10.1109/ICASSP.2018.8461410.
- [17] M. Padgham, Reverberation and frequency attenuation in forests—implications for acoustic communication in animals, *The Journal of the Acoustical Society of America* 115 (2004) 402–410. URL: <https://doi.org/10.1121/1.1629304>. doi:10.1121/1.1629304.

