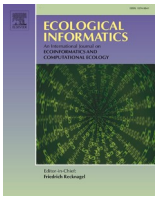
Ecological Informatics 82 (2024) 102699

Contents lists available at ScienceDirect

Ecological Informatics

journal homepage: www.elsevier.com/locate/ecolinf



Improving learning-based birdsong classification by utilizing combined audio augmentation strategies

Arunodhayan Sampath Kumar [a], Tobias Schlosser [a,b], Stefan Kahl [b,c], Danny Kowerko [a,*]

[a] Media Computing, Chemnitz University of Technology, Straße der Nationen 62, Chemnitz 09107, Saxony, Germany
[b] Media Informatics, Chemnitz University of Technology, Straße der Nationen 62, Chemnitz 09107, Saxony, Germany
[c] K. Lisa Yang Center for Conservation Bioacoustics, Cornell Lab of Ornithology, Cornell University, 159 Sapsucker Woods Rd., Ithaca 14850, NY, USA

ARTICLE INFO

Keywords:
Audio classification
Augmentation strategies
Birdsong soundscapes
Computer vision and pattern recognition
Convolutional neural networks
Vision transformers

ABSTRACT

In ecology, changes in environmental conditions are often closely linked to shifts in species diversity. This relationship can be investigated by analyzing avian vocalizations, which are robust indicators of trends in biodiversity. Within this contribution, we explored various data augmentation techniques and deep learning strategies for the classification of birdsong within natural soundscapes. For this purpose, we employed three fundamental deep neural network architectures, such as vision transformers, to classify 397 different bird species. To improve both the accuracy and generalizability of our models, we incorporated up to 19 well-established data augmentation techniques commonly used in audio classification. This included an iterative selection process where only augmentations that enhanced classification performance were selected. The primary augmentation technique involved the integration of various noise samples and non-bird audio elements, which significantly improved model performance as assessed on the BirdCLEF 2021 data set. Individual augmentations achieved F1-scores from 48.0 % (vertical flip) to 72.6 % (primary background noise soundscapes). Through the strategic combination of key techniques – namely simulated pink noise, interspecies sound mixing, and loudness normalization – we achieved a top F1-score of 73.7%. Depending on the selected classification model, this corresponds to an improvement by 4.81 % to 10.5 %. Improvements and deteriorations of all applied augmentation techniques appeared to be robust across our three evaluated models. Therefore, our approach highlights the potential of sophisticated audio augmentations in refining the accuracy and robustness of birdsong classification models.

1. Introduction and motivation

Recent research increasingly acknowledges the potential of (semi-) automated birdsong recognition systems that integrate computer vision (CV) and machine learning (ML) principles. These systems are pivotal for monitoring avian biodiversity, a task traditionally demanding considerable labor and specialized expertise when performed manually (Font et al., 2021; Kahl et al., 2021a). In recent evaluation campaigns, including previous BirdCLEF challenges (Kahl et al., 2018, 2019, 2020, 2021a), deep learning (DL) models have demonstrated their effectiveness. These models proficiently identify and classify birdsong by leveraging sophisticated, task-specific methods tailored for varied environmental contexts.

In 2021, BirdCLEF organized a challenge to classify 397 bird species in 5–300 s snippets of continuous audio recordings from different locations around the globe, with the training data set consisting of 62,874 recordings (Kahl et al., 2021a). The test data contained 80 soundscape recordings of 10 min each in length that were recorded at four locations, namely in COL (Jardín, Departamento de Antioquia, Colombia), COR (Alajuela, San Ramón, Costa Rica), SNE (Sierra Nevada, California, USA), and SSW (Ithaca, New York, USA) (Kahl et al., 2021b; Lasseck, 2019). The test data for classification encompassed different recordings, containing overlapping sounds of different bird events and background noise. Its most challenging parts stem from the presence of weakly-labeled training data with multiple distribution domain shifts, including shifts within the input space, the probability of present labels, and the functionality that links training and test samples to each other. Domain shifts are large differences in data characteristics between clean training recordings and their often noisy test recordings, resulting in an aggravated generalization difficulty for models given previously unseen

* Corresponding author.
E-mail address: danny.kowerko@cs.tu-chemnitz.de (D. Kowerko).

https://doi.org/10.1016/j.ecoinf.2024.102699
Received 18 October 2023; Received in revised form 18 June 2024; Accepted 19 June 2024
Available online 26 June 2024

data.

A total of 1001 participants grouped into 816 teams entered the competition. Their performance was evaluated using the F1-score (Schlosser et al., 2024; Usman and Versfeld, 2024) as a measure of accuracy. On the public test set, which accounts for 35 % of the total test data, the F1-scores for the top 100 teams ranged between 67 % and 80 %. Similarly, in the private test set, making up 65 % of the total test data, the F1-scores for the top 100 teams varied from 60 % to 69 %. Most participants employed ensemble models, incorporating between 10 and 60 different models in their analyses.

In this contribution, we explore various audio augmentation strategies to enhance birdsong classification using deep neural networks (DNN). To study the influence of single as well as combined augmentation strategies, DNNs are employed as single classifiers in order to study the influence of common augmentation methods. The ensuing sections will therefore review previous methodologies for augmentation and evaluation, and detail the contributions of our proposed approaches within this framework.

1.1. Related work

DL methodologies have been effectively applied to large-scale avian species recognition tasks, including the identification of bird species from soundscapes within complex acoustic settings (Kahl et al., 2021b; Lasseck, 2019). Various data augmentation techniques have demonstrated their efficiency in enhancing the accuracy of birdsong identification (see also Fig. 1). Lasseck (2019) presented a time stretching and pitch shifting augmentation by randomly choosing a duration from the audio signal while applying Gaussian noise with a fixed mean and standard deviation. Subsequently, randomly chosen audio signals of the same or even different bird species and non-bird audio events were combined. Kahl et al. (2021b) performed a signal-to-noise ratio based initial preprocessing step in order to reject samples that do not contain bird events. The deployed augmentation strategies encompassed random shifts in time and frequency (vertical and horizontal roll), randomized partial time and frequency stretching, which are frequently used in speech recognition, and randomly added weighted noise samples extracted from audio chunks that did not contain bird events.

Previous state-of-art augmentation strategies encompassed shifting the pitch of the audio signal by a predefined factor, followed by masking chosen segments from the signal while adding simulated white noise to the resulting audio chunks (Mühling et al., 2020). In comparison, the augmentation strategies proposed by Bai et al. (2020) focused on mixing up bird sounds of the same bird species, followed by the addition of noise to the resulting audio chunks. In contrast, Koh et al. (2019) implemented various augmentation techniques on spectrograms. These included randomized vertical and horizontal shifts, the application of a 3×3 blurring kernel, and randomized changes in brightness by scaling the spectrogram. Additionally, additive white noise, randomized cropping, and the randomized blackout of selected consecutive columns were introduced.

The strategies of Wu and Li (2018) focused on noise augmentations such as Gaussian noise, followed by adding noise samples that were recorded by similar equipment in a comparable environment. Schlüter (2021) deployed strategies including the addition of non-bird events for audio chunks, followed by randomized combinations with pitch shifting, the selection of a random pitch factor, and magnitude warping. Finally, Conde et al. (2021) performed augmentations by combining different bird species with simulated white and pink noise, whereas the approaches of Liaqat et al. (2018), Himawan et al. (2018), and Berger et al. (2018) focused on pitch shifting and time stretch augmentations.

In this context, the *Detection and Classification of Acoustic Scenes and Events* workshop (DCASE) (Font et al., 2021) organizes yearly bird classification challenges that focus on the recognition of bird events in given audio data sets. Within these bird recognition challenges, DNNs such as convolutional neural networks (CNN) played a key role in the recognition of bird vocalizations. For this purpose, an increasing number of learning-based approaches are being developed, which are leveraged by the improvements in computational power as well as the availability of large labeled data sets for research and application.

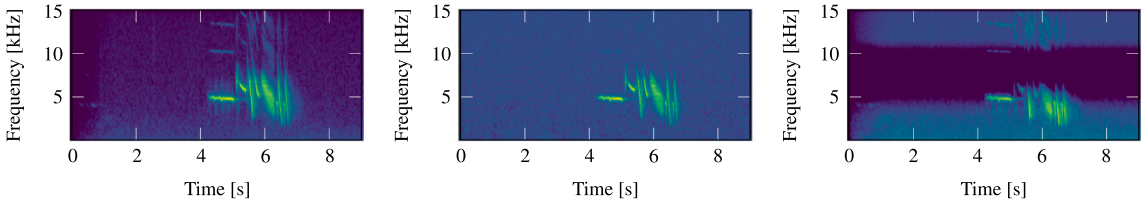
1.2. Contribution of this work

Within birdsong classification, currently more commonly utilized DNN architectures are often based on models such as residual neural networks (ResNet) (He et al., 2016a, 2016b), inception networks (Inception) (Szegedy et al., 2015, 2016, 2017), or DenseNets (Huang et al., 2017). Motivated by the CLEF competition and pretrained audio neural networks (PANN) (Kong et al., 2019), a sound event detection (SED) based approach is proposed that is able to efficiently classify bird vocalizations. Moreover, vision transformers have become a common approach for natural language processing tasks. Yet, their application in tasks of image understanding is still limited (Fu, 2022; Raghu et al., 2021). In this contribution, conventional SED models as well as vision transformers are adapted to the domain of audio classification by splitting the obtained spectrograms into 16×16 or 24×24 linear projections of flattened patches given their positional encodings. These inputs are fed into our models' encoding process to perform a corresponding image classification (Dosovitskiy et al., 2021; Vaswani et al., 2017).

For this purpose, different combinations of audio augmentation strategies for birdsong classification are utilized. Given our combined augmentation strategies, improved classification scores are obtained, for which the contributions of single augmentation approaches are identified and subsequently discussed. For this purpose, we present an iterative approach of combining only augmentation methods that elevate the resulting classification capabilities. Finally, a framework for combined augmentation strategies is proposed that highlights the benefits of different augmentation methods and their combinations.

1.3. Section overview

The remainder of this work is structured as follows. Firstly, we present an overview of the fundamental concepts as well as the implementation of our birdsong augmentation and classification framework (section 2). This includes a detailed description of the employed augmentation methods, our data preprocessing steps, the architectural design of our classification models, and our training configuration.



(a) Original audio (b) Gaussian noise (c) Frequency mask

Fig. 1. Exemplary overview of commonly utilized augmentation methods for data augmentation within learning-based birdsong classification.

Table 1
Overview of different audio augmentation techniques. Listed are audio augmentation approaches using, among others, pitch shifting, time stretching, mix up, and Gaussian noise.

Literature	Pitch shifting	Time stretching	Mix up	Gaussian noise	Vertical roll	Horizontal roll	Blur	Brightness	Masking	Warping
BirdCLEF 2021 (Kahl et al., 2021a)										
Lasseck (2019)	✓	✓	✓	✓						
Kahl et al. (2021b)	✓	✓	✓		✓	✓				✓
Mühling et al. (2020)	✓		✓						✓	
Bai et al. (2020)			✓		✓	✓				
Koh et al. (2019)			✓		✓	✓	✓	✓		
Wu and Li (2018)	✓		✓	✓						
Schlüter (2021)	✓		✓	✓						✓
Conde et al. (2021)			✓							
DCASE 2021 (Font et al., 2021)										
Liaqat et al. (2018)	✓	✓								
Himawan et al. (2018)	✓	✓								
Berger et al. (2018)	✓	✓								

Subsequently, our test results, including different augmentation methods, are discussed (section 3). These are provided as a comparison of baseline approaches, whereas their possible combinations are further investigated. These discussions aim to delve more deeply into the distinctions of the selected augmentation methods. Finally, we offer perspectives on potential future enhancements and possible further optimizations of our framework (section 4).

2. Fundamentals and implementation

Fundamental to the performance of audio classification in the context of birdsong identification, the following sections provide a formal definition of our leveraged approaches by utilizing the audio samples' spectrograms as input imagery. This includes all subsequent processing steps of data preprocessing, classification using different DL-based models, as well as training and testing given our proposed training setup.

2.1. Software requirements and reproducibility

Our implementations are powered by Python version 3.8 with PyTorch version 1.13.1. For reproducibility purposes, a Docker container has been created with the CUDA Toolkit version 11.8, containing all used dependencies. It can be found via Docker Hub.[2]

2.2. Augmentation methods

Table 1 presents commonly used augmentation strategies that are often deployed for different combinations of bird species and background noise, followed by a pitch shift and time stretch. Given their visualizations and general explanations, further detailed descriptions can be found in Iqbal et al. (2021). In the following, the audio data augmentation library Audiomentations[3] and the *2018 BirdCLEF Baseline System*[4] have been utilized with standard parameters. In addition to pink noise and noise soundscapes for audio background augmentation, background noise is differentiated into primary and secondary noise. Primary noise is characterizable as a global sample-wise augmentation, whereas secondary noise is applied locally by mixing various noise with or without overlapping samples with randomized pauses between them. Taking this differentiation into account, we obtain 19 different augmentation methods:

1) Gaussian noise. Gaussian noise is added to the audio signal with randomly chosen weights, which is then renormalized (Fig. 2b) (Audiomentations: function `AddGaussianNoise()`). It is calculated via

$$S' = S + A \cdot N, \tag{1}$$

where S' represents the audio samples after the addition of Gaussian noise, S the original audio samples, A the amplitude of the noise, and N the Gaussian noise generated with the same shape as S, with $n_i \sim \mathcal{N}(0,1)$ for each element n_i in N.

2) Pink noise (background). Background noise in the form of pink noise is added. The pink noise is generated using colorednoise.[5] Adding background noise is a so-called mix up augmentation method in which the labels of the background noise are neglected (Fig. 2c) (Audiomentations: function `AddBackgroundNoise()`).

3–7) **Noise soundscapes (background).** The background noise is noise data extracted from soundscapes (Fig. 2d) (Audiomentations: function `AddBackgroundNoise()`).

8.) **General mix up.** For augmentation methods 2–8, the training samples are constructed via

$$x = x_i + (1-\tau) \cdot x_j, \tag{2}$$

where x_i and x_j are two randomly selected samples from the training data. τ is the mix up's ratio, which ranges from 0 to 1 (Fig. 2e). It is randomly selected from this range. The *Pytorch Image Models (timm)'s* `mixup()` function is utilized.[6]

9.) **Horizontal roll.** This method is applied to the spectrograms. It rolls the spectrogram with respect to $width(W) \times \alpha$, where α is the roll factor (e.g., 0.5) (Fig. 2f) (2018 BirdCLEF Baseline System: function `roll()` with augmentation `'roll_h'`). Given a spectrogram S with dimensions $F \times T$, where F is the number of frequency bins and T is the number of time frames, we define a rolling operation that shifts the spectrogram horizontally. This operation results in a new spectrogram S'

[2] Docker Hub Docker container of this work, https://hub.docker.com/r/arunodhayan/journal_2024_ecologicalinformatics_birdsongaugmentation

[3] Audiomentations project page, https://github.com/iver56/audiomentations

[4] *Recognizing Birds from Sound - The 2018 BirdCLEF Baseline System*, https://github.com/kahst/BirdCLEF-Baseline

[5] colorednoise.py project page, https://github.com/felixpatzelt/colorednoise

[6] Pytorch Image Models (timm): Mixup & CutMix Augmentations, https://timm.fast.ai/mixup_cutmix

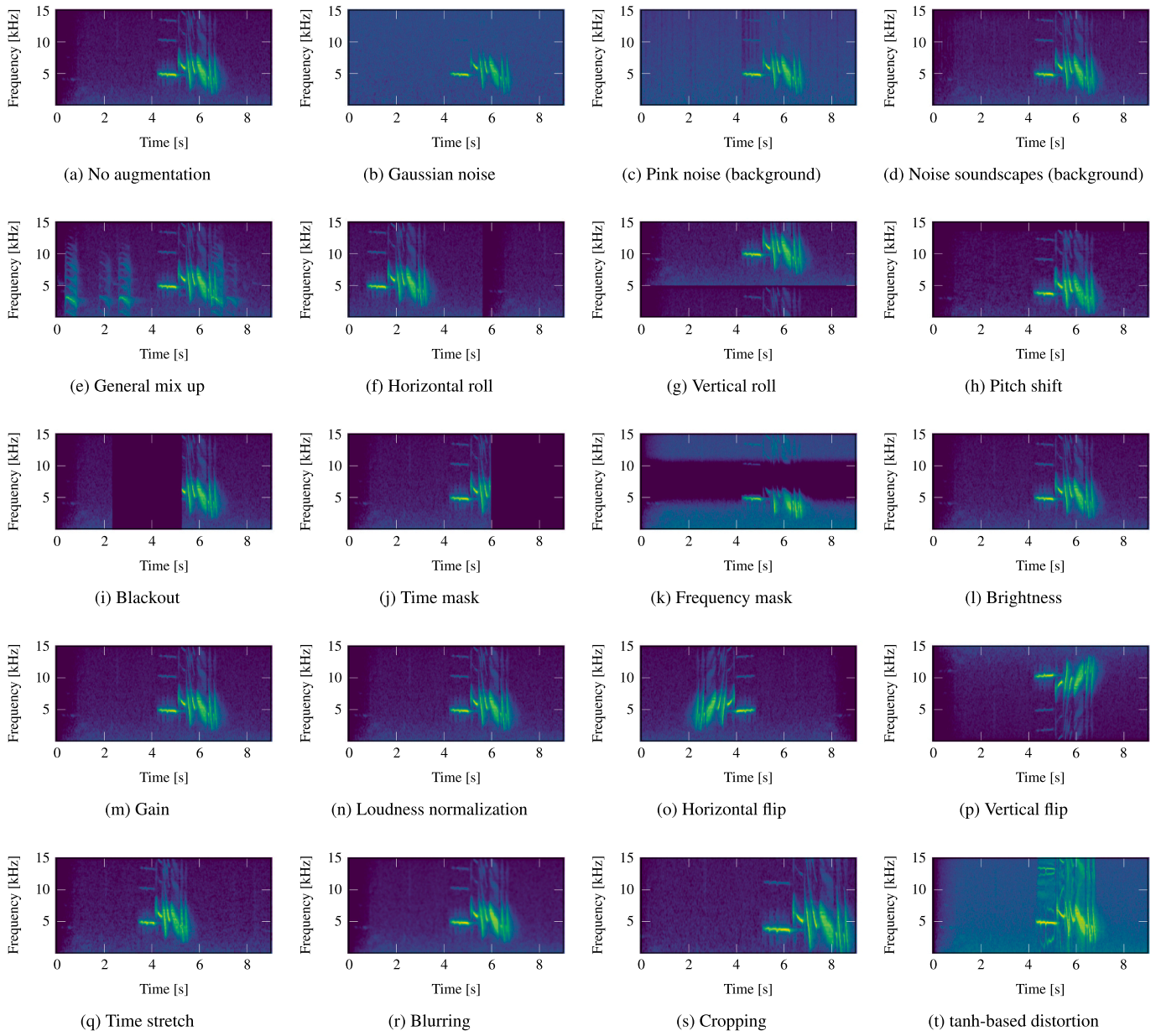


Fig. 2. Illustration of different augmentation techniques for birdsong augmentation given the audio's spectrogram. Table 2 shows a more detailed description of the depicted augmentations and their respective domains.

of the same dimensions. The new value of each element in S' is determined via

$$S'[i,j] = S[i, (j + W{\cdot}\alpha) \bmod T], \tag{3}$$

where i indexes the frequency bins, j indexes the time frames, W is the total width of the spectrogram in terms of the number of time frames (i.e., $W = T$), and *mod* is the modulo operation. It ensures that the indexing wraps around the matrix correctly.

10.) **Vertical roll.** This method is applied to the spectrograms. It rolls the spectrogram with respect to *height*$(H) \times \alpha$, where α is the roll factor (e.g., 0.05) (Fig. 2g) (2018 BirdCLEF Baseline System: function `roll()` with augmentation `'roll_v'`). Given a spectrogram S with dimensions $F \times T$, where F is the number of frequency bins and T is the number of time frames, we define a rolling operation that shifts the spectrogram vertically. This operation results in a new spectrogram S' of the same dimensions. The new value of each element in S' is determined via

$$S'[i,j] = S[(i + H{\cdot}\alpha) \bmod F, j], \tag{4}$$

where i indexes the frequency bins, j indexes the time frames, H is the total height of the spectrogram in terms of the number of frequency bins (i.e., $H = F$), and *mod* is the modulo operation.

11.) **Pitch shift.** Shifts the pitch of the sound up or down without changing the audio's pace (Fig. 2h) (Audiomentations: function `PitchShift()`). It is calculated as follows.

$$\begin{aligned} \alpha &= 2^{\frac{-pitch\ factor}{12}} \\ X(f) &= STFT(x(t)) \\ Y(f) &= X(f{\cdot}\alpha) \\ y(t) &= STFT^{-1}(Y(f)) \end{aligned} \tag{5}$$

Definitions:

- α is the rate by which the frequency is changed.
- $x(t)$ is the original audio signal as a function of the time t in the time domain.

- $X(f)$ is the Fourier transform of $x(t)$.
- $Y(f)$ is the modified audio signal in the frequency domain.
- $y(t)$ is the inverse Fourier transform of $Y(t)$.
- *STFT* is the short-time Fourier transform (STFT), whereas $STFT^{-1}$ is the inverse STFT.

The *pitch factor* can range from −12 to 12. It represents the number of semitones by which the pitch of the audio signal is to be shifted. Positive values cause the pitch to increase (upward shift), while negative values cause the pitch to decrease (downward shift).

12.) **Time mask.** t consecutive time steps are masked, where t is chosen from a uniform normal distribution (Fig. 2j) (Audiomentations: function `TimeMask()`).

Definitions:

- $S = [s_1, s_2, \ldots, s_\tau]$ represents the original sequence of length τ.
- T is the maximum number of consecutive time steps that can be masked.
- t is the number of consecutive time steps to mask. It is selected uniformly at random from the set $\{1, 2, \ldots, T\}$.
- t_0 is the starting index for the mask. It is selected uniformly at random from the set $\{0, 1, \ldots, \tau - t\}$.
- The masked period is randomly replaced with either the mean of the original values or zero.

Random selection of mask parameters:

$$t \sim Uniform(1, T)$$
$$t_0 \sim Uniform(0, \tau - t)$$

Application of the mask to the sequence: The masked sequence S' is defined via

$$s_i' = \begin{pmatrix} mask & if\ t_0 \le i < t_0 + t \\ s_i & otherwise. \end{pmatrix} \tag{6}$$

This step is repeated for each $i \in \{1, 2, \ldots, \tau\}$.

13.) **Frequency mask.** f consecutive mel frequency channels are masked, where f is chosen from a uniform normal distribution (Fig. 2k) (Audiomentations: function `SpecFrequencyMask()`).

Definitions:

- $M = [m_{ij}]$ represent the log-mel spectrogram. i indexes the frequency bins and j indexes the time frames.
- F is the maximum number of frequency channels that can be masked.
- v is total number of mel frequency bands.
- f is the number of consecutive frequency channels to mask. It is selected uniformly at random from $\{0, 1, \ldots, F\}$.
- f_0 is the starting index for the mask. It is selected uniformly at random from $\{0, 1, \ldots, v - f\}$.
- The masked frequencies are randomly replaced with either the mean of the original values or zero.

Random selection of mask parameters:

$$f \sim Uniform(0, F)$$
$$f_0 \sim Uniform(0, v - f)$$

Application of the mask to the log-mel spectrogram: The masked log-mel spectrogram M' is defined via

$$m_{ij}' = \begin{pmatrix} mask & if\ f_0 \le j < f_0 + f \\ m_{ij} & otherwise. \end{pmatrix} \tag{7}$$

This step is repeated for each i and j.

14.) **Gain.** The audio signal is multiplied by a random amplitude factor to reduce or increase the present volume. This technique can help a model approach invariance to the overall gain of the input audio (Fig. 2m) (Audiomentations: function `Gain()`). It is calculated via

$$x' = \alpha \cdot x, \tag{8}$$

where x is the original audio signal, x' is the modified audio signal after applying the gain, and $\alpha \sim Uniform(\alpha_{min}, \alpha_{max})$ is a random amplitude factor selected uniformly between α_{min} and α_{max}.

15.) **Loudness normalization.** A constant amount of gain is applied to match a specific loudness (Fig. 2n) (Audiomentations: function `LoudnessNormalization()`). It is calculated via

$$x' = \beta \cdot x, \tag{9}$$

where x is the original audio signal, x' is the audio signal after the loudness normalization, and $\beta = 10^{L_{target} - L_{current}/20}$ is the gain factor needed to adjust the signal from its current loudness $L_{current}$ to the target loudness L_{target}.

16.) **Horizontal flip.** The spectrogram is horizontally flipped along the y-axis (Fig. 2o) (2018 BirdCLEF Baseline System: function `flip()`). It is calculated via

$$S'(t, f) = S(T - t, f), \tag{10}$$

where $S(t, f)$ is the value of the original spectrogram at time t and frequency f, $S'(t, f)$ is the flipped spectrogram, and T is the total number of time frames in the spectrogram. t runs from 0 to $T - 1$.

17.) **Vertical flip.** The spectrogram is vertically flipped along the x-axis (Fig. 2p) (2018 BirdCLEF Baseline System: function `flip()`). It is calculated via

$$S'(t, f) = S(t, F - f), \tag{11}$$

where $S(t, f)$ is the value of the original spectrogram at time t and frequency f, $S'(t, f)$ is the flipped spectrogram, and F is the total number of frequency bins in the spectrogram. f runs from 0 to $F - 1$.

18.) **Time stretch.** The signal is stretched in time without changing the signal's pitch. For the application of time stretch, its rate factor is randomly sampled from $[0.9, 1.5]$ (Fig. 2q) (Audiomentations: function `TimeStretch()`). It is calculated as follows.

$$\begin{aligned} X(f) &= STFT(x(t)) \\ \phi'(X(f)) &= \phi(X(f)) \cdot \lambda \\ Y(f) &= X(\phi'(X(f))) \\ y(t) &= STFT^{-1}(Y(f)) \end{aligned} \tag{12}$$

Definitions:

- λ is the time-stretch factor.
- $x(t)$ is the original audio signal as a function of the time t in the time domain.
- $X(f)$ is the Fourier transform of $x(t)$.
- $\phi(X(f))$ is the phase information of $X(f)$, $\phi'(X(f))$ is the time-stretched phase information, and $X(\phi'(X(f)))$ is the Fourier transform of $x(t)$ with time-stretched phase information.
- $Y(f)$ is the modified audio signal in the frequency domain.
- $y(t)$ is the inverse Fourier transform of $Y(t)$.
- *STFT* is the short-time Fourier transform (STFT), whereas $STFT^{-1}$ is the inverse STFT.

Depending on the time-stretch factor λ, the signal is either sped up or slowed down. If $\lambda > 1$, the signal is sped up (the duration of the signal decreases). If $\lambda < 1$, the signal is slowed down (the duration of the signal increases).

19.) tanh**-based distortion.** This technique adds a tanh-based distortion to the audio recording. The hyperbolic tangent functionality can provide a soft clipping. The magnitude of the distortion is proportional to the loudness of the input and the pre-gain. As the hyperbolic tangent is symmetric, the positive and negative parts of the signal are equally squashed (Fig. 2t) (Audiomentations: function `TanhDistortion()`). It is calculated via

Table 2
Overview of our augmentation strategies, their IDs, and related information. For DenseNet-161 and ViT-B/16, their training times are provided in minutes per epoch for the different augmentation methods. For IDs 5 and 6, we added noise of the bird audio detection data set from the DCASE challenge (Berger et al., 2018; Himawan et al., 2018; Liaqat et al., 2018), which is abbreviated as *BAD*.

					DenseNet-161	ViT-B/16
ID	**Data augmentation**	**Time domain**	**Frequency domain**	**Spectrogram**	**Training time [min. / epoch]**	
1	Gaussian noise	✓			15	24
2	Pink noise (background)	✓			15	24
3	Primary background noise soundscapes	✓			13	30
4	Secondary background noise soundscapes	✓			13	30
5	Primary background noise (*BAD*)	✓			13	30
6	Secondary background noise (*BAD*)	✓			13	30
7	Background noise (e.g., wind, thunder, or aircrafts)	✓			13	30
8	Mixed up random bird species			✓	26	45
9	Horizontal roll			✓	15	26
10	Vertical roll			✓	15	26
11	Pitch shift		✓		40	60
12	Time mask	✓			29	45
13	Frequency mask			✓	16	40
14	Gain	✓			13	30
15	Loudness normalization	✓			13	30
16	Horizontal flip			✓	15	30
17	Vertical flip			✓	15	30
18	Time stretch		✓		16	35
19	tanh-based distortion	✓			10	20

Table 3
Summarized DNN layer configuration for the backbone models DenseNet-161 and ResNet-50 as well as ViT-S/16, ViT-B/16, and ViT-L/16. S, B, and L denote the ViT models' small, base, and large variants.

Model	Layers	Hidden size	MLP size	Params [m.]	Training time [min. / epoch]	Testing time [sec. / sample]
DenseNet-161	161	48	397	28.5	15	7
ResNet-50	50	64	397	25.6	15	7
ViT-S/16	8	786	2358	48.6	20	22
ViT-B/16	12	786	3072	86.8	35	27
ViT-L/16	24	1024	4096	304.6	45	36

$$x'(t) = tanh(\gamma \cdot x(t)), \tag{13}$$

where $x(t)$ is the original audio signal, $x'(t)$ is the distorted audio signal, and γ is the pre-gain applied to the signal before distortion, which controls its intensity.

At this point it is noted that the augmentation method blackout (Fig. 2i) is similar to a time mask augmentation (Fig. 2j) where a selected region of interest is set to zero. For this reason, the time mask augmentation has been excluded from further investigations. Likewise, brightness, blurring, and cropping were discarded, too. Table 2 shows a more detailed description of the depicted augmentations and their domains.

2.3. Data preprocessing

In this contribution, we utilize the 2021 BirdCLEF data set with its recordings. Since these recordings are weakly labeled, we used randomly selected chunks with a length of 5 s for the training process. Some of them also contain non-bird audio events such as wind, thunder, or aircrafts. To diversify our non-bird audio events, we additionally deployed related noise from AudioSet, "a large-scale dataset of manually annotated audio events"[7] (Fonseca et al., 2018). However, to also mitigate the possible side effects of these audio events, we decided to use the first as well as the last 5 s of each clip after examining them regarding their audio quality. For this purpose, we extracted clips without birdsong from the validation soundscape recordings based on the available metadata. These clips were added as background noise to the training samples.

Apart from clips without birdsong, we added noise from the bird audio detection data set from the DCASE challenge (Berger et al., 2018; Himawan et al., 2018; Liaqat et al., 2018) (abbreviated as *BAD*) as well as pink and other noise sources, including insect- and human-made sounds. To compute the audio's log-mel spectrograms, we used the audio analysis library librosa[8] (McFee et al., 2015). Subsequently, the short-time Fourier transform (STFT) was applied to the resulting waveform with a Hamming window size of 1 024 with 64 mel bins and a hop size of 384 (12 ms). The frequency range of bird vocalizations ranged from 250 Hz to 8.3 kHz (Hu and Cardoso, 2009), for which we restricted the general frequency range to values between 50 Hz and 15 kHz. Finally, our preprocessed data set encompassed 5 s long log-mel spectrograms extracted from 63 898 training samples.

2.4. Model architecture

The proposed model architectures are based on DenseNet-161 (Huang et al., 2017) and ResNet-50 (He et al., 2016a, 2016b) as well as different vision transformers (ViT) (Dosovitskiy et al., 2021; Steiner et al., 2022) for sound event detection.

The design of our SED models is inspired by the 2019 DCASE workshop and large-scale PANNs for audio pattern recognition (Kong et al., 2019). DenseNets are utilized to improve the information flow

[7] AudioSet project page, https://research.google.com/audioset/

[8] librosa project page, https://librosa.org/doc/latest/index.html

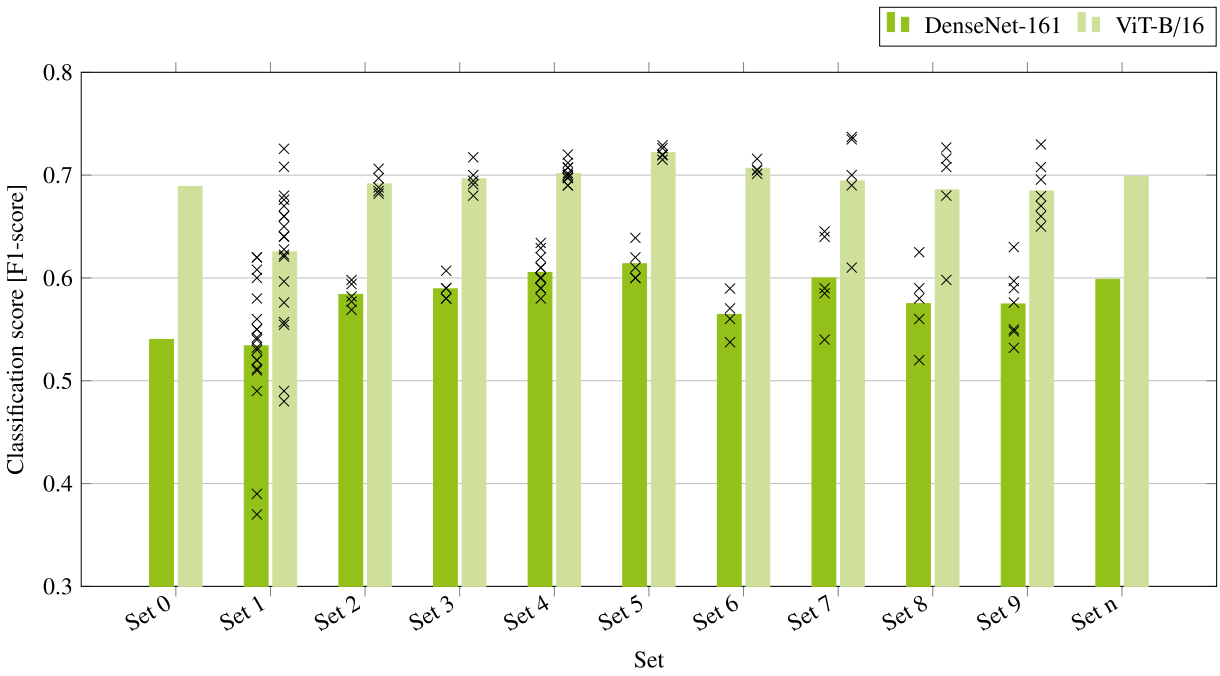


Fig. 3. Comparison of different augmentation methods and their combinations in Table 4 using DenseNet-161 and ViT-B/16. Additionally, plot marks for the conducted experiments and their results are provided.

between layers, whereby a different connectivity pattern is introduced with direct connections from all current layers to all subsequent layers as compared to conventional CNNs. The change within the resulting feature maps is facilitated by downsampling, whereas multiple densely connected layers result in an even deeper DNN. Subsequently, the log-mel spectrograms are fed as input imagery to our models. Only the features prior to the second-to-last fully connected layer are obtained. Finally, a modified one-dimensional attention-based fully connected layer is attached. The output of this network contains clip- and frame-wise outputs.

For the ViT models, we adapted the approach of pretraining deep bidirectional transformers for language understanding (BERT) by Devlin et al. (2018) for birdsong classification. Table 3 provides an overview of the investigated ViT models with their respective parameterizations. ViT-S/16, for example, denotes a ViT variant of reduced complexity with an input patch size of 16×16. The sequence length of the transformer models is inversely proportional to the square of the provided patch size. In the following, ViT-B/16 is further investigated as it strikes a balance between general model complexity and classification performance.

2.5. Training setup

ViT models are often computationally costly compared to conventional SED models. However, our 5 s long log-mel spectrograms are more feasible to process. The influences of class imbalances were alleviated by utilizing the binary cross entropy based focal loss as well as the so-called scaled focal loss (Arunodhayan Sampathkumar, 2021). Our training setup includes the Adam optimizer (Kingma and Ba, 2015) with a concatenating cosine-annealing linear scheduler with an initial learning rate of 0.0001, decaying by a factor of $0.001 \cdot$*learning rate*, and a batch size of 32. For validation, we used a 5-fold cross validation. Given the class imbalance, we additionally deployed a data set sampler by upsampling and duplicating randomly selected samples and augmenting them. Our models were trained for 50 epochs without mixing up augmentations as well as for 100 epochs with mixing up augmentations. Early stopping was introduced to prevent overfitting when no further training or validation progress could be observed. To fine-tune our ViT models, we deployed ImageNet-based weights for pretraining.

While the 2021 BirdCLEF challenge is hosted via Kaggle,[9] we also conducted local experiments. We evaluated our setup using general-purpose graphics processing units (GPGPU). Our test environment is composed solely of current consumer grade hardware. This test environment encompasses similar to Schlosser et al. (2022) (i) our CPU, "Intel(R) Core(TM) i9-9900K CPU @ 3.60GHz" with 7200 BogoMips and a maximum CPU load of 99%, (ii) our GPU, "TITAN RTX" with a maximum GPU load of 99%, (iii) our working memory with 128 GB of RAM, as well as (iv) our hard drive (SSD), "Samsung 970 EVO Plus SSD" with 500 GB.

3. Test results, evaluation, and discussion

The following sections provide an overview of our test results as well as our evaluation and discussion in the context of different augmentation methods. Therefore, a differentiation is made between test runs without any data augmentation, single augmentation runs, and runs with combined augmentations. Finally, the best results and their combinations are discussed.

3.1. Baseline augmentation methods

Table 2 shows the specific domains to which the respective augmentation methods were applied to. Fig. 3 visualizes the impact of these methods on model performance. A comprehensive overview of all results is compiled in Table 4, displaying F1-scores (Schlosser et al., 2024; Usman and Versfeld, 2024) for various single and combined augmentation strategies.

No augmentation methods (set 0). Scores within the range of 52.6 % (ResNet-50) to 68.9 % (ViT-B/16) are obtained. These results can be seen as the lower reference point for all experiments.

Single augmentation methods (set 1). Our single augmentation methods are applied with a probability of 50%. The following results are observed:

[9] BirdCLEF 2021 - Birdcall Identification, https://www.kaggle.com/c/birdclef-20211

Table 4
The impact of data augmentation strategies on model performance is quantified. The backbone architectures employed are DenseNet-161, ResNet-50, and ViT-B/16. The IDs of the data augmentations assessed are detailed in Table 2. Red checks (+) signify the introduction of additional augmentations in comparison to their preceding sets. For all experiments, the probability of applying any given augmentation technique was set to 50%. The symbol "/" denotes instances where significantly diminished scores were omitted from further evaluation.

																				DenseNet-161	ResNet-50	ViT-B/16
ID	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**F1-score**		
Set 0																				0.5402	0.5260	0.6890
Set 1	✓																			0.5200	0.5100	0.6404
		✓																		0.5300	0.4900	0.6226
			✓																	0.6200	0.5920	0.7256
				✓																0.6080	0.6000	0.6800
					✓															0.6200	0.5920	0.7080
						✓														0.5800	0.6000	0.6700
							✓													0.5500	0.5600	0.6400
								✓												0.5600	0.5400	0.6600
									✓											0.5323	0.5400	0.6206
										✓										0.5500	0.4800	0.6277
											✓									0.4900	0.4700	0.5571
												✓								0.5116	0.5200	0.5966
													✓							0.5200	0.4600	0.5545
														✓						0.5400	0.5300	0.6500
															✓					0.5420	0.5360	0.6600
																✓				0.3700	0.3900	0.4900
																	✓			0.3900	0.4000	0.4800
																		✓		0.5100	0.5000	0.5760
																			✓	0.6000	0.5600	0.6760
Set 2			✓	✓																0.5690	/	0.6820
					✓	✓														0.5820	/	0.6880
			✓			✓														0.5940	/	0.7062
				✓	✓															0.5980	/	0.6970
			✓	✓	✓	✓														0.5770	/	0.6845
Set 3	+		✓	✓																0.5800	/	0.6910
	+				✓	✓														0.5800	/	0.6800
	+		✓			✓														0.6070	/	0.7173
	+			✓	✓															0.5900	/	0.7000
	+		✓	✓	✓	✓														0.5900	/	0.6944
Set 4	✓		✓			✓	+													0.6100	/	0.7072
	✓		✓			✓		+												0.6340	/	0.7200
	✓		✓			✓				+										0.5900	/	0.6900

(*continued on next page*)

Table 4 (*continued*)

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	DenseNet-161 F1-score	ResNet-50	ViT-B/16
	✓		✓			✓								+						0.6290	/	0.7119
	✓		✓			✓									+					0.6000	/	0.7074
	✓		✓			✓													+	0.6000	/	0.7000
	✓			✓	✓		+													0.5900	/	0.6972
	✓			✓	✓			+												0.6000	/	0.6973
	✓			✓	✓					+										0.6100	/	0.7000
	✓			✓	✓									+						0.6200	/	0.7019
	✓			✓	✓										+					0.6000	/	0.6974
	✓			✓	✓														+	0.5800	/	0.6900
Set 5	✓		✓			✓	+	✓												0.6390	/	0.7289
	✓		✓			✓		✓		+										0.6000	/	0.7192
	✓		✓			✓		✓						+						0.6000	/	0.7200
	✓		✓			✓		✓							+					0.6100	/	0.7149
	✓		✓			✓		✓											+	0.6200	/	0.7262
Set 6	✓		✓			✓	✓	✓		+										0.5375	/	0.7046
	✓		✓			✓	✓	✓						+						0.5897	/	0.7012
	✓		✓			✓	✓	✓							+					0.5602	/	0.7158
	✓		✓			✓	✓	✓											+	0.5704	/	0.7053
Set 7	✓		✓			✓	✓	✓		✓				✓	✓	+	+			0.5400	0.5300	0.6100
	✓		✓			✓	✓	✓		✓	+			✓	✓					0.5850	0.5487	0.6900
	✓		✓			✓	✓	✓		✓	+			✓	✓			+		0.5900	0.5785	0.7000
	✓		✓			✓	✓	✓		✓				✓	✓				+	0.6452	0.6300	0.7347
	✓		✓			✓	✓	✓		✓				✓	✓					0.6400	0.6300	0.7371
Set 8	✓			✓	✓		✓	✓		✓				✓	✓	+	+			0.5200	0.5190	0.5980
	✓			✓	✓		✓	✓		✓	+			✓	✓					0.5600	0.5335	0.6800
	✓			✓	✓		✓	✓		✓	+			✓	✓			+		0.5800	0.5780	0.7078
	✓			✓	✓		✓	✓		✓				✓	✓				+	0.5900	0.5900	0.7160
	✓			✓	✓		✓	✓		✓				✓	✓					0.6250	0.6150	0.7268
Set 9	✓	✓			✓			✓												0.5320	0.5124	0.6500
		✓			✓	✓														0.5970	0.5785	0.6795
	✓	✓			✓															0.5900	0.5723	0.7077
	✓	✓			✓		✓	✓		✓				✓	✓					0.6300	0.6100	0.7298
	✓	✓			✓			✓		✓	✓			✓	✓			✓		0.5500	0.5083	0.6600
	✓	✓						✓		✓	✓			✓	✓			✓		0.5480	0.5132	0.6700
	✓	✓																		0.5760	0.5672	0.6956
Set n	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	0.5987	0.5512	0.6988

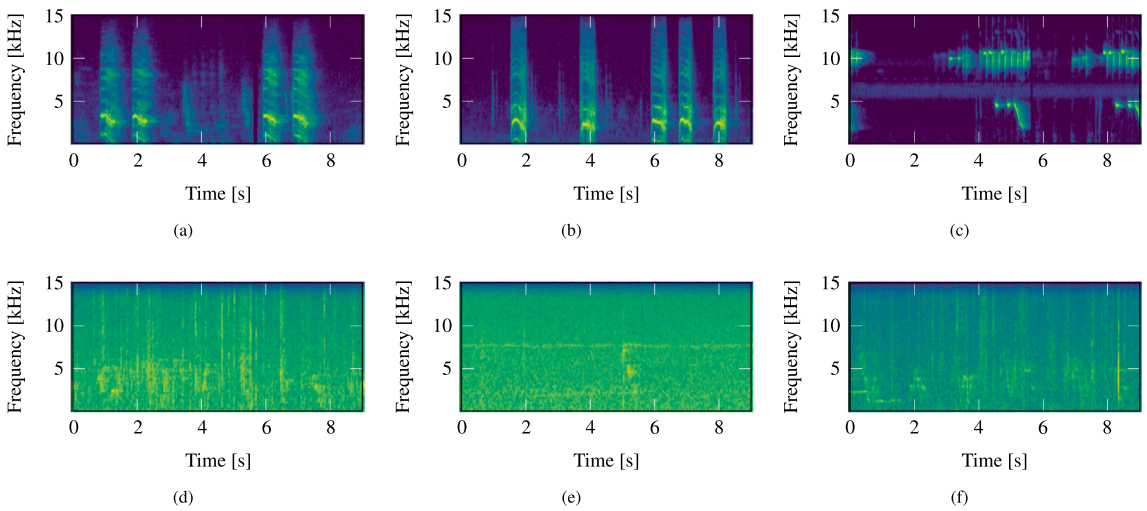


Fig. 4. Exemplary samples within our training (a–c) and test sets (d–f). (a–c) depict optimal training samples, whereas (d–f) show exemplary soundscapes that are to be evaluated. (a–c) show improved quality due to a higher microphone quality. (d–f) are often recorded via mobile devices, which in turn highlights the difficulty of model training and evaluation given the illustrated deviations. The training samples are part of the xeno-canto data set as they are provided with the BirdCLEF 2021 challenge.[11]

1,2) **Gaussian and pink noise.** Applying only Gaussian and pink noise has no noticeable effect on the model performance when compared with the baseline scores.

3,4) **Primary and secondary background noise soundscapes.** Applying only primary and secondary background noise using noise data extracted from the soundscape validation recordings has an increased impact on the model performance compared with the baseline scores.

5,6) **Primary and secondary background noise (*BAD*).** Applying only primary and secondary background noise from non-bird events (*BAD*) has an increased impact on model performance (LeBien et al., 2020). In comparison with soundscape recordings as background noise, the model with *BAD* noise is not able to outperform the other models. Since the test set is related to soundscape recordings, the impact of soundscape noise is greater than that of other non-bird event sounds.

7) **General background noise.** Applying soundscape-like background noise to the training samples does not influence the performance of the model.

8) **Mixing up random bird species.** Mixing up random bird species within the training samples has no impact on model performance compared with other noise augmentation methods. Unlike label-preserving methods, mix up constructs the soft labels of new samples by combining two labels. As labels are generally encoded as one-hot vectors, the newly generated soft labels can be considered to belong to multiple categories of raw samples. Therefore, mix up implicitly increases the training samples and reduces the generalization gap. However, when applied separately, it has no major impact on the model performance compared to noise-only methods. Yet, other augmentation techniques are still outperformed.

9,10) **Horizontal and vertical roll.** Horizontal roll and vertical roll are not able to outperform the other noise augmentation methods. However, the other spectrogram augmentation methods are outperformed.

11) **Pitch shift.** Pitch shift influences the time and frequency domains. Time stretch influences the time domain, hence changing the vocality of the training samples. When these augmentation methods are applied to the training samples, no noticeable effect on the model performance is observable.

12,13) **Time and frequency mask.** The time and frequency masks affect the time and frequency domains, respectively. Both augmentation methods are applied at random time intervals to the training samples. In some cases, when bird sounds are masked within the training samples, an overall decrease in performance is conceivable.

14,15) **Gain and loudness normalization.** The operations of gain and loudness normalization are similar to applying noise to the training samples. The augmentation methods show increased scores.

16,17) **Horizontal and vertical flip.** Horizontal and vertical flipping are similar to their respective image augmentation methods. Overall, a decline in performance, even in comparison to the baseline, is observable.

18,19) **Time stretch and** tanh**-based distortion.** Applying a time stretch or a tanh-based distortion to the samples resembles the pattern of adding soundscape noise to the samples. An improvement in model performance is obtained for the tanh-based distortion.

All augmentation methods (set n). Scores within the range of 59.9 % (ResNet-50) to 69.9 % (ViT-B/16) are obtained. These results can be seen as the upper reference point for all experiments.

3.2. Combined augmentation methods

Based on the outcomes presented in Table 4, various augmentation methods were combined to enhance our models' classification performance. To highlight the augmentation techniques used, red checks (✓) indicate added augmentations in comparison to their preceding sets. For example, the first subset of set 2 is composed of the augmentation methods with the IDs 3 and 4, namely, primary and secondary background noise from soundscapes (see Table 2). The following observations provide an overview of our sets of combined augmentation methods:

- **Set 2.** The top 4 results from set 1 are combined. The augmentation methods with the IDs 3 (primary soundscapes), 4 (secondary soundscapes), 5 (primary *BAD*), and 6 (secondary *BAD*) are combined.
- **Set 3.** Set 2 combinations with Gaussian noise.
- **Set 4.** The top 2 results from set 3 and top 5–10 ranked augmentations from set 1 are combined.
- **Set 5.** The top result from set 4 as well as the 5 remaining augmentation methods from set 4 are paired individually.
- **Set 6.** The top result from set 5 as well as the 4 remaining augmentation methods from set 4 are paired individually.
- **Set 7.** Set 5 (top 1–5 of set 4) combinations and other, weakly-performing augmentations from set 1 are combined.
- **Set 8.** Set 4 (top 7–12 of set 4) combinations and other, weakly-performing augmentations from set 1 are combined.

Table 5
Summary of the best augmentations from Table 4 in F1-score. The symbol "/" highlights results with strongly reduced scores that have been omitted from further evaluation. The best result is highlighted in bold.

ID	DenseNet-161	ResNet-50	ViT-B/16
Set 0	0.5402	0.5260	0.6890
Set 1	0.6200	0.6000	0.7256
Set 2	0.5980	/	0.7062
Set 3	0.6070	/	0.7173
Set 4	0.6340	/	0.7200
Set 5	0.6390	/	0.7289
Set 6	0.5897	/	0.7158
Set 7	**0.6452**	**0.6300**	**0.7371**
Set 8	0.6250	0.6150	0.7268
Set 9	0.6300	0.6100	0.7298
Set n	0.5987	0.5512	0.6988

- **Set 9.** Adds pink noise and augmentations from set 1 to obtain results without soundscape noise.

The following results are being observed:

- **Set 2.** The top 4 augmentation approaches from set 1 are combined to investigate the best augmented noise combinations. The combination of soundscape noise as primary as well as *BAD* noise as secondary noise shows increased scores. The second-best noise combination is *BAD* noise as primary noise with soundscape noise as secondary noise.
- **Set 3.** Gaussian noise is applied to the combinations of set 1 to analyze the best-performing model. The best result is obtained from the combination of Gaussian noise as primary noise with *BAD* noise as secondary noise. A minor improvement in classification score is observed.
- **Set 4.** The top 2 best noise combinations are combined with other best-performing augmentations. The top 6 best augmentation combinations with the top 2 best noise combinations from set 3 are combined. Random bird species are mixed up with vertical rolls, gain augmentation, tanh-based distortion, and other background noise as well as a loudness normalization. A minor improvement in classification score is obtained.
- **Set 5.** The top 1 combination along other augmentation combinations from set 5 are used. The best augmentation combinations from these sets are background noise, Gaussian noise with mix up, and vertical roll. Different augmentation combinations result in the data set becoming increasingly complex. However, an overall improvement in model performance is obtained.
- **Set 6.** The strategy from set 4 to set 5 has been continued. With the addition of further augmentation methods, a decline in F1-score is observable. As set n marks the upper reference point for an approach were all augmentation methods are combined (69.9 % for ViT-B/16), it is concluded that no additional major performance improvements can be made. For this purpose, a selective strategy for the combination of different augmentation methods is advisable starting with set 7.
- **Set 7.** Primary noise of soundscapes and *BAD* secondary noise shows an increased impact on model performance. Together with noise augmentations, spectrogram augmentations such as mix up, background noise, and vertical roll are combined that further improve model performance. The best overall model performance is achieved. Soundscape noise as primary noise improved the overall performance.
- **Set 8.***BAD* primary noise and secondary noise of soundscapes are combined with augmentations from set 5. The obtained results are similar to the results of set 7.
- **Set 9.** Soundscape noise augmentations are replaced with pink noise. From the obtained results with the best-performing model combination of set 7 it is evident that the replacement of soundscape noise with pink noise shows a decrease in classification score by about 1.52 %, 2.00 %, and 0.73 % for DenseNet-161, ResNet-50, and ViT-B/16.

After our initial strategy of adding only the best previous augmentation methods up to set 6, a change in strategy has been realized with the beginning of set 7. This change in the choice of augmentation methods was influenced by the provided test samples. Our rationale is summarized as follows. As depicted in Fig. 4, a significant domain shift exists between the training and test data sets. To enhance the model accuracy, we adopted noise-based augmentations (see Table 2, IDs 1–7). Additionally, the mix up bird species augmentation, marked as ID 8, was utilized due to the occurrence of multiple bird events in the test set. Furthermore, we integrated augmentations such as gain, loudness normalization, and a tanh-based distortion, denoted as IDs 12, 13, and 19, respectively. These were chosen to amplify the sound, guarantee a consistent volume perception, and introduce a smooth saturation effect, mimicking the warmth and distinct character of analog audio samples. Additionally, other augmentations marked as IDs 9–13, 16, 17, and 18 were employed given their common usage in general audio classification tasks.

3.3. Overall results

We evaluated our system during testing using the models' F1-scores. The trained system was evaluated with our soundscape data set. Table 5 illustrates the best results from our different augmentation sets. Set 0 shows the weakest performance since no augmentation methods were applied. Set n results in an improvement by 5.85 % (DenseNet-161), 2.52 % (ResNet-50), and 0.98 % (ViT-B/16) with all augmentation methods. Over all results, it is observable that all changes over all sets show the same magnitude for our three evaluated models for sets 0 to n. While the weaker-performing models, DenseNet-161 and ResNet-50, result in stronger relative improvements, ViT-B/16 shows the best overall scores.

The top-performing models from set 1 are noise-only augmentations. From the combined augmentation strategies, set 5 shows an increased model performance when soundscapes as primary noise and *BAD* noise as secondary noise are applied. Within sets 7 to 9, the other top-performing model is situated in set 7. Here, soundscape noise was replaced by pink noise, resulting in the overall best performance. Depending on the selected classification model, an improvement by 10.5 % (DenseNet-161), 10.4 % (ResNet-50), and 4.81 % (ViT-B/16) is obtained. Therefore, it is concluded that noise augmentations play an important role in improving model performance. Furthermore, they also reduce the possible risks of model overfitting while improving the models' generalization capabilities. Additionally, it is evident that by adding soundscape noise events to the training samples, an overall improved model performance could be achieved.

3.4. Discussion

Augmentations such as Gaussian noise, horizontal flip, and vertical flip can deteriorate the overall results compared to no augmentations (set 0). Significant improvements of the single set augmentation strategy (set 1) could be achieved by 8%, 6–7% and 2–4% increased F1-scores for DenseNet-161, ResNet-50, and ViT-B/16, respectively. Here, primary

[1] xeno-canto project page, https://xeno-canto.org/

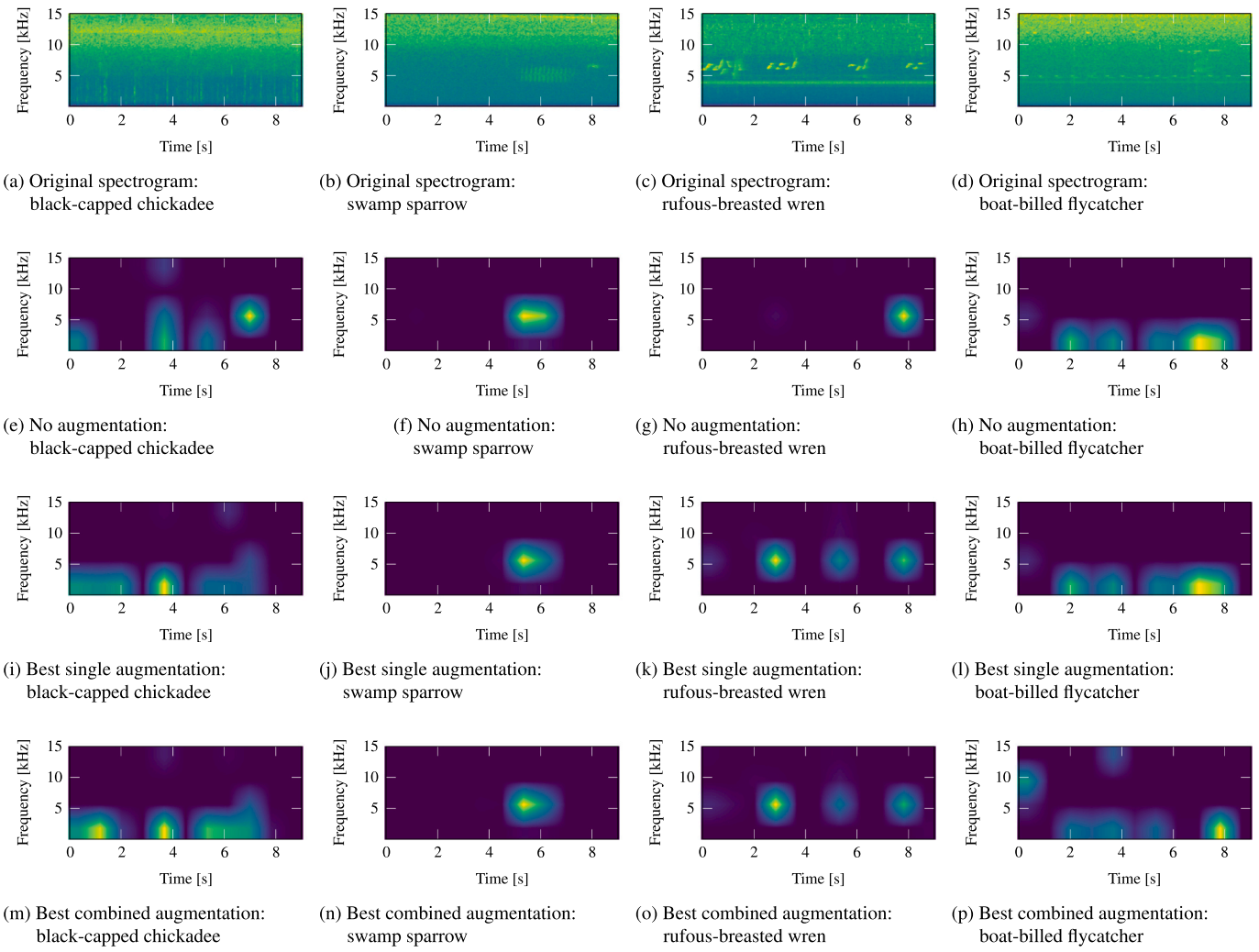


Fig. 5. Grad-CAM-based class activation maps for randomly selected samples of the classes black-capped chickadee (*bkcchi*), swamp sparrow (*swaspa*), rufous-breasted wren (*rubwre1*), and boat-billed flycatcher (*bobfly1*) (from left to right). From top to bottom, the original spectrograms (a–d) as well as their activation maps are depicted. These include: results without any augmentation methods (e–h), with the best single augmentation method (i–l: set 1, augmentation ID 3, *primary background noise soundscapes*), and with the best combined augmentation methods (m–p: set 7, line 5 in Table 4) are depicted.

background noise soundscapes and primary background noise (*BAD*) yielded the best results. Overall, a significant improvement in model performance could be achieved by adding combined augmentation techniques such as Gaussian noise, noise in general, mixing up random bird species, vertical roll, gain, loudness normalization, and a tanh-based distortion. These helped to further improve the model performance in set 7 to 64.52 % (DenseNet-161), 63.00 % (ResNet-50), and 73.71 % (ViT-B/16), with ViT-B/16 showing the best overall performance in F1-score.

In order to alleviate the effects of present class imbalances, the focal loss was introduced to the training process. To reduce the impact of computational costs within the context of the BirdCLEF challenge in terms of model runtime, our augmentation techniques were applied to the training samples with a probability factor 50%. Therefore, future investigations should also assess the differences in classification capabilities when different probabilities are compared with each other. When further assessing the resulting training and testing times in Table 3, it is evident that the improved performance of ViT-B/16 comes at a cost of increased training times by a factor of about 2.3 as well as increased testing times by a factor of about 3.9. The observed data augmentation trends – for increasing and decreasing classification capabilities – are merely robust against the investigated models. Therefore, future investigations focusing on augmentation strategies should also study and optimize the model performance in terms of classification capabilities and inference times.

Fig. 5 facilitates a further, qualitative evaluation, presenting gradient-weighted class activation mappings (Grad-CAM). Shown are the resulting class activation maps (Selvaraju et al., 2016, 2017) of four randomly selected spectrograms from the classes black-capped chickadee (*bkcchi*), swamp sparrow (*swaspa*), rufous-breasted wren (*rubwre1*), and boat-billed flycatcher (*bobfly1*) (from left to right). Displayed from top to bottom are the original spectrograms (a–d), followed by their corresponding class activation maps without any augmentation methods (e–h), with the best-performing single augmentation method (i–l: set 1, augmentation ID 3, *primary background noise soundscapes*), and with the most effective combination of augmentation methods (m–p: set 7, line 5 in Table 4).

When comparing the different results for all four spectrograms, it is observed that different benefits in class activations are obtained. With no augmentation methods, less focused class activations, especially for the class rufous-breasted wren, are obtained (see c, g, k, and o). When comparing the best single augmentation method and the best combination of augmentation methods, fewer differences are observable. However, when comparing the results for the first spectrogram (i.e., i and m) and the second spectrogram (i.e., l and p) with each other, slightly more focused class activations are obtained. These are

influenced by the application of local pink noise.

4. Conclusion and outlook

Birdsong classification challenges such as the BirdCLEF challenge series have a clear focus on identifying birds from soundscape recordings. The goal is to bridge the domain gap between training and test data as both data are often recorded by using different microphones with, in turn, different characteristics in terms of recording quality as well as species diversity. In previous BirdCLEF competitions, augmentation strategies were the key factor for improving the overall performance of various models. Many different augmentation techniques have been explored to improve model performance.

Within our contribution, we showed that augmentation methods such as added soundscapes and non-bird events helped to improve the previously obtained baseline scores with otherwise single augmentation methods. However, other augmentations that do not involve the addition of noise to the training samples were not able to improve model performance compared with our baseline approaches. We conclude that most single augmentations, apart from adding noise samples, do not notably improve model performance due to domain shifts within the training and test sets, for which we only obtained results with decreased scores. However, combinations of noise augmentations with other, single augmentations were able to achieve increased results. Using the vision transformer ViT-B/16, we obtain a final improvement from 68.90 %, without any augmentations, to 73.71 %, our best combined augmentation, by 4.81 %.

The evaluated augmentation techniques consistently exhibited the same qualitative effects across all three of our deep learning models. Specifically, if the performance of an augmentation set improved for one model, it also improved for the other two, and vice versa. However, further research is needed to determine if this observation can be generalized. For optimizing augmentation strategies, a more effective approach would be to use only the least computationally intensive models, e.g., DenseNet-161 or ResNet-50. The best combined augmentations would then be applied to the best-performing model (ViT-B/16).

In our future work, we plan on focusing on the creation of so-called no-call classifiers with training samples that do not contain bird events in general. For this purpose, as well as to explore different transformer-based models with improved prediction scores, encompassing, i.a., data-efficient image transformers (DeiT) as well as hybrid transformers (Han et al., 2021; Touvron et al., 2021a, 2021b), further investigations with application-specific fine-tuning will have to be performed. This also encompasses different probability factors for the application of the augmentation strategies, for which the differences in varying probability factors will have to be investigated.

Author contributions

Arunodhayan Sampath Kumar and Tobias Schlosser conducted this contribution's writing process and evaluation with the help of Stefan Kahl and Danny Kowerko in realizing this manuscript. The experiments performed and the associated implementations were carried out by Arunodhayan Sampath Kumar under the supervision of Tobias Schlosser, Stefan Kahl, and Danny Kowerko.

CRediT authorship contribution statement

Arunodhayan Sampath Kumar: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Conceptualization. **Tobias Schlosser:** Writing – review & editing, Writing – original draft, Visualization, Validation. **Stefan Kahl:** Supervision, Project administration, Funding acquisition. **Danny Kowerko:** Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no conflict of interest.

Data availability

Data is publicly available on the kaggle birdclef challenge 2021

Acknowledgment

The European Union and the European Social Fund for Germany partially funded this research. Our work in the K. Lisa Yang Center for Conservation Bioacoustics is made possible by the generosity of K. Lisa Yang to advance innovative conservation technologies to inspire and inform the conservation of wildlife and habitats.

References

Arunodhayan Sampathkumar, D.K., 2021. TUC media computing at BirdCLEF 2021: Noise augmentation strategies in bird sound classification in combination with DenseNets and ResNets. In: Faggioli, G., Ferro, N., Joly, A., Maistro, M., Piroi, F. (Eds.), Proceedings of the Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, Bucharest, Romania. Pp. 1–10. URL: https://ceur-ws.org/Vol-2936/paper-138.pdf.

Bai, J., Chen, C., Chen, J., 2020. Xception based method for bird sound recognition of BirdCLEF 2020. In: Cappellato, L., Eickhoff, C., Ferro, N., Névéol, A. (Eds.), Proceedings of the Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, CEUR-WS.org, Thessaloniki, Greece. Pp. 1–9. URL: https://ceur-ws.org/Vol-2696/paper_127.pdf.

Berger, F., Freillinger, W., Primus, P., Reisinger, W., 2018. Bird Audio Detection - DCASE 2018. Technical Report. DCASE2018 Challenge. Surrey, UK. URL: https://dcase.community/documents/challenge2018/technical_reports/DCASE2018_Berger_66.pdf.

Conde, M.V., Shubham, K., Agnihotri, P., Movva, N.D., Bessenyei, S., 2021. Weakly-supervised classification and detection of bird sounds in the wild. A BirdCLEF 2021 solution. In: Proceedings Working Notes CEURWS at CLEF 2021 (BirdCLEF 2021), CEUR-WS.org, Bucharest, Romania, pp. 1–12. URL: https://arxiv.org/abs/2107.04878.

Devlin, J., Chang, M.W., Lee, K., Toutanova, K., 2018. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. URL: http://arxiv.org/abs/1810.04805.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al., 2021. An image isWorth 16x16Words: Transformers for image recognition at scale. In: In: 2021 International Conference on Learning Representations (ICLR 2021), Vienna, Austria, pp. 1–22. URL: https://arxiv.org/abs/2010.11929.

Fonseca, E., Plakal, M., Font, F., Ellis, D.P.W., Favory, X., Pons, J., Serra, X., 2018. General-purpose tagging of freesound audio with audioset labels: task description, dataset, and baseline. In: Plumbley, M.D., Kroos, C., Bello, J.P., Richard, G., Ellis, D., Mesaros, A. (Eds.), Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE 2018)., Tampere University of Technology, Surrey, UK, pp. 1–5. URL: https://dcase.community/challenge2018/taskgeneral-purpose-audio-tagging.

Font, F., Mesaros, A., Ellis, D.P.W., Fonseca, E., Fuentes, M., Elizalde, B., 2021. Proceedings of the 6th workshop on detection and classication of acoustic scenes and events (DCASE 2021). In: Font, F., Mesaros, A., Ellis, D.P.W., Fonseca, E., Fuentes, M., Elizalde, B. (Eds.), Detection and Classication of Acoustic Scenes and Events. https://doi.org/10.5281/zenodo.5770113. URL: https://dcase.community/workshop2021/proceedings.

Fu, Z., 2022. Vision Transformer: Vit and its Derivatives. URL: https://arxiv.org/abs/2205.11239. https://doi.org/10.48550/ARXIV.2205.11239.

Han, K., Xiao, A., Wu, E., Guo, J., Xu, C., Wang, Y., 2021. Transformer in transformer. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (Eds.), Advances in Neural Information Processing Systems 34 (NeurIPS 2021), Curran Associates, Inc.. Pp. 15908–15919. URL: https://proceedings.neurips.cc/paper/2021/hash/854d9fca60b4bd07f9bb215d59ef5561-Abstract.html.

He, K., Zhang, X., Ren, S., Sun, J., 2016a. Deep residual learning for image recognition. In: In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, pp. 770–778. URL: https://ieeexplore.ieee.org/document/7780459.

He, K., Zhang, X., Ren, S., Sun, J., 2016b. Identity mappings in deep residual networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (Eds.), Computer Vision–ECCV 2016: 14th European Conference. Springer, Amsterdam, Netherlands, pp. 630–645. URL: https://link.springer.com/chapter/10.1007/978-3-319-46493-0_38, doi:10.1007/978-3-319-46493-0{_}38.

Himawan, I., Towsey, M., Roe, P., 2018. 3D convolution recurrent neural networks for bird sound detection. In: Plumbley, M.D., Kroos, C., Bello, J.P., Richard, G., Ellis, D., Mesaros, A. (Eds.), Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE 2018)., DCASE2018 Challenge. Tampere

University of Technology, Surrey, UK, pp. 1–4. URL: https://eprints.qut.edu.au/122760/.

Hu, Y., Cardoso, G.C., 2009. Are bird species that vocalize at higher frequencies preadapted to inhabit noisy urban areas? Behav. Ecol. 20, 1268–1273. URL: https://academic.oup.com/beheco/article/20/6/1268/200758.

Huang, G., Liu, Z., Maaten, L.V.D., Weinberger, K.Q., 2017. Densely connected convolutional networks. In: In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Honolulu, HI, USA, pp. 4700–4708. URL: https://ieeexplore.ieee.org/document/8099726.

Iqbal, T., Helwani, K., Krishnaswamy, A., Wang, W., 2021. Enhancing audio augmentation methods with consistency learning. In: ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, Toronto, ON, Canada, pp. 646–650. URL: https://ieeexplore.ieee.org/document/9414316.

Kahl, S., Wilhelm-Stein, T., Klinck, H., Kowerko, D., Eibl, M., 2018. Recognizing Birds from Sound - the 2018 BirdCLEF Baseline System. URL: https://arxiv.org/abs/1804.07177.

Kahl, S., Stöter, F.R., Goëau, H., Glotin, H., Planque, R., Vellinga, W.P., Joly, A., 2019. Overview of BirdCLEF 2019: Large-scale bird recognition in soundscapes. In: Cappellato, L., Ferro, N., Losada, D.E., Müller, H. (Eds.), Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum, CEUR-WS.org, Lugano, Switzerland, pp. 1–10. URL: https://ceurws.org/Vol-2380/paper_256.pdf.

Kahl, S., Clapp, M., Hopping, W.A., Goëau, H., Glotin, H., Planqué, R., Vellinga, W.P., Joly, A., 2020. Overview of BirdCLEF 2020: Bird sound recognition in complex acoustic environments. In: Cappellato, L., Eickhoff, C., Ferro, N., Névéol, A. (Eds.), Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, CEUR-WS.org, Thessaloniki, Greece, pp. 1–15. URL: https://ceur-ws.org/Vol-2696/paper_262.pdf.

Kahl, S., Denton, T., Klinck, H., Glotin, H., Goëau, H., Vellinga, W.P., Planqué, R., Joly, A., 2021a. Overview of BirdCLEF 2021: Bird call identification in soundscape recordings. In: Faggioli, G., Ferro, N., Joly, A., Maistro, M., Piroi, F. (Eds.), Proceedings of the Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, Bucharest, Romania. Pp. 1–14. URL: https://ceur-ws.org/Vol-2936/paper-123.pdf.

Kahl, S., Wood, C.M., Eibl, M., Klinck, H., 2021b. BirdNET: a deep learning solution for avian diversity monitoring. Eco. Inform. 61, 101236. URL: https://www.sciencedirect.com/science/article/pii/S1574954121000273.

Kingma, D.P., Ba, J., 2015. Adam: A method for stochastic optimization. In: International Conference on Learning Representations (ICLR 2015), Ithaca, NY, pp. 1–13. URL: https://arxiv.org/abs/1412.6980.

Koh, C.Y., Chang, J.Y., Tai, C.L., Huang, D.Y., Hsieh, H.H., Liu, Y.W., 2019. Bird sound classification using convolutional neural networks. In: Cappellato, L., Ferro, N., Losada, D.E., Müller, H. (Eds.), Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum, CEUR-WS.org, Lugano, Switzerland, pp. 1–10. URL: https://ceurws.org/Vol-2380/paper_68.pdf.

Kong, Q., Cao, Y., Iqbal, T., Wang, Y., Wang, W., Plumbley, M.D., 2019. PANNs: large-scale pretrained audio neural networks for audio pattern recognition. IEEE/ACM Trans. Audio Speech Language Proc. 28, 2880–2894. URL: https://ieeexplore.ieee.org/document/9229505.

Lasseck, M., 2019. Bird species identification in soundscapes. In: Cappellato, L., Ferro, N., Losada, D.E., Müller, H. (Eds.), Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum, CEUR-WS.org, Lugano, Switzerland, pp. 1–10. URL: http://ceur-ws.org/Vol-2380/paper_86.pdf.

LeBien, J., Zhong, M., Campos-Cerqueira, M., Velev, J.P., Dodhia, R., Ferres, J.L., Aide, T. M., 2020. A pipeline for identification of bird and frog species in tropical soundscape recordings using a convolutional neural network. Eco. Inform. 59, 101113 https://doi.org/10.1016/j.ecoinf.2020.101113.

Liaqat, S., Bozorg, N., Jose, N., Conrey, P., Tamasi, A., Johnson, M.T., 2018. Domain tuning methods for bird audio detection. In: Plumbley, M.D., Kroos, C., Bello, J.P., Richard, G., Ellis, D., Mesaros, A. (Eds.), Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE 2018)., Tampere University of Technology, Surrey, UK, pp. 1–5. URL: https://dcase.community/documents/challenge2018/technical_reports/DCASE2018_Liaqat_96.pdf.

McFee, B., Raffel, C., Liang, D., Ellis, D.P.W., McVicar, M., Battenberg, E., Nieto, O., 2015. librosa: Audio and music signal analysis in Python. In: Huff, K., Bergstra, J. (Eds.), Proceedings of the 14th Python in Science Conference, Austin, Texas, pp. 18–24. URL: http://conference.scipy.org.s3-website-us-east-1.amazonaws.com/proceedings/scipy2015/brian_mcfee.html. 10.25080/Majora-7b98e3ed-003.

Mühling, M., Franz, J., Korfhage, N., Freisleben, B., 2020. Bird species recognition via neural architecture search. In: Cappellato, L., Eickhoff, C., Ferro, N., Névéol, A. (Eds.), Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, CEUR-WS.org, Thessaloniki, Greece, pp. 1–13. URL: https://ceur-ws.org/Vol-2696/paper_188.pdf.

Raghu, M., Unterthiner, T., Kornblith, S., Zhang, C., Dosovitskiy, A., 2021. Do vision transformers see like convolutional neural networks? In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (Eds.), Advances in Neural Information Processing Systems 34 (NeurIPS 2021), Curran Associates, Inc.. Pp. 12116–12128. URL: https://proceedings.neurips.cc/paper_files/paper/2021/hash/652cf38361a209088302ba2b8b7f51e0-Abstract.html.

Schlosser, T., Friedrich, M., Beuth, F., Kowerko, D., 2022. Improving automated visual fault inspection for semiconductor manufacturing using a hybrid multistage system of deep neural networks. J. Intell. Manuf. 33, 1099–1123. URL: https://link.springer.com/article/10.1007/s10845-021-01906-9.

Schlosser, T., Friedrich, M., Meyer, T., Kowerko, D., 2024. A Consolidated Overview of Evaluation and Performance Metrics for Machine Learning and Computer Vision. https://doi.org/10.13140/RG.2.2.14331.69928. https://www.researchgate.net/publication/374558675_A_Consolidated_Overview_of_Evaluation_and_Performance_Metrics_for_Machine_Learning_and_Computer_Vision. URL:

Schlüter, J., 2021. Learning to monitor birdcalls FromWeakly-labeled focused recordings. In: Faggioli, G., Ferro, N., Joly, A., Maistro, M., Piroi, F. (Eds.), Proceedings of the Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, CEUR-WS.org, Bucharest, Romania. Pp. 1–12. URL: https://www.ceur-ws.org/Vol-2936/paper-139.pdf.

Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D., 2016. Grad-CAM: Why Did you Say that? URL: https://arxiv.org/abs/1611.07450.

Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D., 2017. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV 2017), IEEE, Venice, Italy, pp. 2380–7504. URL: https://ieeexplore.ieee.org/document/8237336. https://doi.org/10.1109/ICCV.2017.74.

Steiner, A., Kolesnikov, A., Zhai, X., Wightman, R., Uszkoreit, J., Beyer, L., 2022. How to train your ViT? Data, augmentation, and regularization in vision transformers. Trans. Machine Learn. Res. 1–16. URL: https://arxiv.org/abs/2106.10270.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Boston, MA, USA, pp. 1–9. URL: https://ieeexplore.ieee.org/document/7298594.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., 2016. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition, IEEE, Las Vegas, NV, USA, pp. 2818–2826. URL: https://ieeexplore.ieee.org/document/7780677. https://doi.org/10.1109/CVPR.2016.308.

Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A., 2017. Inception-v4, inception-ResNet and the impact of residual connections on learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, 31(1). AAAI Press, San Francisco, California USA, pp. 1–7. URL: https://ojs.aaai.org/index.php/aaai/article/view/11231.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H., 2021a. Training data-efficient image transformers & distillation through attention. In: Meila, M., Zhang, T. (Eds.), Proceedings of the 38th International Conference on Machine Learning, PMLR, PMLR. pp. 10347–10357. URL: https://proceedings.mlr.press/v139/touvron21a.

Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G., Jégou, H., 2021b. Going deeper with image transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, Montreal, QC, Canada, pp. 32–42. URL: https://ieeexplore.ieee.org/document/9710634. https://doi.org/10.1109/ICCV48922.2021.00010.

Usman, A., Versfeld, D., 2024. Principal components-based hidden Markov model for automatic detection of whale vocalisations. J. Mar. Syst. 242, 103941. URL: https://www.sciencedirect.com/science/article/pii/S0924796323000854.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 30 (NIPS 2017). Curran Associates, Inc, Long Beach, California, USA, pp. 1–11. URL: https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Wu, H., Li, M., 2018. Construction and improvements of bird Songs' classification system. In: Cappellato, L., Ferro, N., Nie, J.Y., Soulier, L. (Eds.), Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum, CEUR-WS.org, Avignon, France. pp. 1–8. URL: https://ceurws.org/Vol-2125/paper_77.pdf.