Birdsong Classification Using Convolutional Neural Networks

We are aiming to get a **Proposed Grade**.

1st Bódi Vencel
Faculty of Electronics Engineering and Informatics
Budapest University of Technology and Economics
team Train to Failure
VBW5N9 – bodi.vencel04@gmail.com

2nd Mitrenga Márk
Faculty of Transportation Engineering and Vehicle Engineering
Budapest University of Technology and Economics
team Train to Failure
OLLNTB – mitrengamark@sztaki.hu

***Abstract*—Avian biodiversity is a critical bioindicator for understanding ecological changes, particularly in biodiversity hotspots like the Western Ghats in south-west India. Traditional bird biodiversity surveys are expensive and logistically challenging, but Passive Acoustic Monitoring (PAM) combined with machine learning offers a scalable alternative. This work presents our solution to the BirdCLEF 2024 challenge.**

In this project, we propose a machine learning-based approach to address the challenges of limited training data and class imbalance in birdsong classification. Our preliminary results demonstrate a highly competitive ROC-AUC curve on benchmark datasets compared to existing solutions shared on the Kaggle platform.

I. Introduction

BirdCLEF 2024 is part of the LifeCLEF 2024 [1] competition family. It aims to leverage machine learning techniques to advance the identification and monitoring of Indian bird species through their sounds [2]. Participants are tasked with developing computational solutions capable of processing continuous audio data and accurately recognizing bird species based on their calls. This competition emphasizes creating reliable classifiers that perform well even with limited training data.

A. Background and Context

Birds serve as vital indicators of biodiversity, reflecting ecosystem changes due to their mobility and habitat diversity. However, traditional observer-based bird biodiversity surveys are costly and logistically challenging over large areas. In contrast, Passive Acoustic Monitoring (PAM), combined with advanced machine learning methods, provides a cost-effective way to monitor biodiversity.

The Western Ghats, a Global Biodiversity Hotspot in India, home to unique ecosystems and extraordinary bird diversity, including many endemic and endangered species. However, rapid habitat modification and climate change threaten this fragile region, emphasizing the urgent need for conservation technologies. BirdCLEF 2024 (which's logo is on Fig. 1) supports this goal by enabling automated detection and classification of bird species from soundscapes, particularly focusing on understudied and nocturnal species.

B. Current Solutions

The BirdCLEF 2024 challenge [3] ended on June 10, 2024, and all submitted solutions are now publicly available on the Kaggle platform. Participants developed a wide range of models and approaches for birdsong classification, which provides a valuable benchmark for evaluating new methods. These solutions offer insights into diverse methodologies, including neural network architectures and strategies for addressing challenges such as class imbalance and limited training data.

For this project, we compared our model's performance against the publicly available Kaggle solutions, focusing on ROC-AUC [4] as the primary final evaluation metric. During the training we monitored the validation accuracy. While we followed a similar preprocessing pipeline to those used in the Kaggle submissions, we deviated in the testing methodology. Since the competition is over, we could not test our solution with the private test dataset on the Kaggle website yet.

By comparing our results to those of the Kaggle submissions, we aimed to validate the robustness of our approach and identify potential areas for improvement. This comparative analysis underscores the competitiveness of our model and highlights its potential for real-world applications beyond the scope of the BirdCLEF competition.

Fig. 1. Logo of the BirdCLEF competition

II. BirdCLEF 2024 Dataset

The BirdCLEF 2024 dataset consists of 24,459 labelled and 8,444 unlabelled audio samples, complemented by a wide range of metadata. The labelled dataset includes annotations for bird species, geographic locations, and recording quality, making it suitable for training and validating machine learning models. The unlabelled samples present opportunities for semi-supervised or unsupervised learning approaches, enabling the exploration of methods that utilize unannotated data effectively.

In addition to the raw audio data, the dataset provides metadata fields such as recording quality and geographic coordinates which can serve as auxiliary features for advanced models. The wide distribution of species and the inherent class imbalance pose unique challenges that must be addressed during preprocessing and model training.

A. Exploratory Data Analysis

To better understand the dataset, we conducted an Exploratory Data Analysis (EDA) on the labelled portion of the dataset, focusing on key aspects such as class distributions and recording durations.

1) Class Imbalance:
The dataset exhibits significant disparities in the number of recordings per class, with certain bird species being underrepresented. This imbalance directly impacts model training and necessitates careful handling during preprocessing. Fig. 2 illustrates the distribution of the number of recordings per class, highlighting the skewed nature of the dataset.

2) Recording Length Distribution:
The recording lengths vary drastically, ranging from a few seconds to several minutes, maybe hours. This variability introduces challenges in data processing, particularly for models expecting fixed-size inputs.

Table I summarizes the percentile distribution of audio lengths, while Fig. 3 demonstrates the distribution of recording lengths. The presence of extreme outliers highlights the need for effective handling during preprocessing.

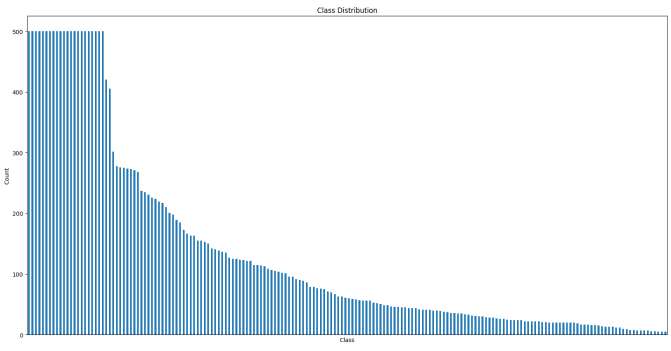

Fig. 2. Distribution of number of recordings per class.

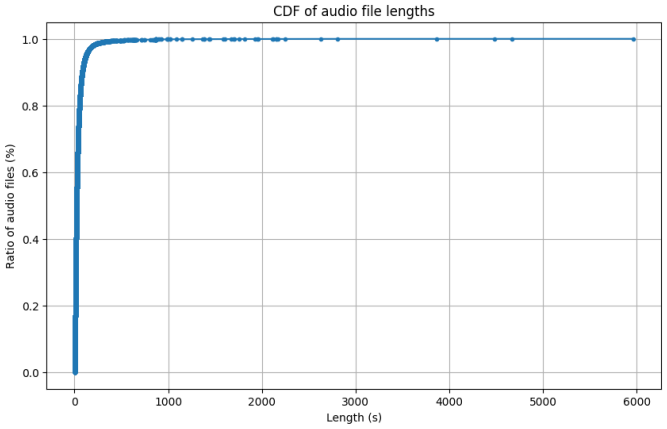

Fig. 3. Cumulative distribution of audio lengths

TABLE I
PERCENTILE DISTRIBUTION OF AUDIO LENGTHS.

Percentile (%)	Audio Length (s)
10	6.72
25	11.21
50	22.39
75	44.67
90	82.25
95	122.47
99	300.13
100	5964.23

III. Development Environment

Our project relies on several key libraries to efficiently process audio data and train deep learning models. Below is a summary of the primary libraries we used and the reasons for their selection.

- **PyTorch** is a versatile and widely-used deep learning framework. It provides dynamic computation graphs, making it easier to debug and implement complex neural network architectures. Its flexibility and efficient GPU acceleration made it an ideal choice for training our CNN-based models.
- **TorchAudio** extends PyTorch's capabilities to handle audio data, offering built-in tools to load, transform and enhance audio. It seamlessly integrates with PyTorch and allows us to implement preprocessing pipelines such as waveform resampling with minimal effort.
- **Librosa** is a Python library specialized in audio and music analysis. It provides advanced tools for feature extraction and audio transformation, including high-level mel-spectrogram generation. We used it mainly for the generation of mel-spectograms, conducting EDA and visualizations.
- **Ray Tune** is a scalable and flexible hyperparameter optimization library. It supports various search algorithms and schedulers, enabling efficient exploration of hyperparameter spaces. This ensured the best possible performance on the bird song classification task while minimizing manual effort in experimentation.

IV. From Spectrograms to Classification

The main goal of the project was to train a deep neural network that is capable of classifying birds by song. There are multiple conventional approaches regarding audio processing [5]. One of the widely spread approaches is based on the spectral decomposition of audio waves using the short time Fourier transform (STFT).

A. Short Time Fourier Transform

The STFT provides a time-frequency representation of an audio signal by dividing it into overlapping segments and applying the Fourier Transform to each segment [6]. Mathematically, the STFT of a signal $x(t)$ is defined as in eq. 1:

$$\text{STFT}(x(t))(t,\omega) = \int_{-\infty}^{\infty} x(\tau)w(\tau - t)e^{-j\omega\tau}d\tau \quad (1)$$

where $w(\tau - t)$ is a sliding window function that localizes the signal in time. This decomposition allows us to convert raw audio into a spectrogram, which is a 2D representation of frequency over time. [7]

B. Mel-Spectrogram Transformation

While spectrograms provide a detailed frequency representation, they are not directly aligned with human auditory perception. To address this, we transformed the spectrograms into mel-spectrograms using the mel scale, which is a perceptually motivated scale of pitches [8]. This highly contributes to achieving better model performance [9].

The mel frequency f_{mel} corresponding to a linear frequency f (in Hz) is given by eq. 2:

$$f_{\text{mel}} = 2595 \cdot \log_{10}(1 + \frac{f}{700}) \quad (2)$$

Using this scale, the frequency bins of the spectrogram are mapped to the mel scale. The resulting mel-spectrogram $M(t,m)$ is computed as seen in eq. 3:

$$M(t,m) = \sum_k |S(t,f_k)|^2 \cdot H_m(f_k) \quad (3)$$

where $S(t,f_k)$ is the STFT of the signal at frequency f_k, and $H_m(f_k)$ is the triangular filter corresponding to the m-th mel band. The power $|S(t,f_k)|^2$ represents the energy at frequency f_k and time t.

C. Amplitude Scaling

The raw mel-spectrogram values are further converted to a logarithmic amplitude scale, commonly referred to as the decibel (dB) scale. This step enhances the representation of lower energy signals, making it more suitable for neural network processing. The conversion to the dB scale is given by the equation numbered as 4:

$$M_{\text{dB}}(t,m) = 10 \cdot \log_{10}(\max(M(t,m),\epsilon)) \quad (4)$$

where ϵ is a small constant to avoid taking the logarithm of zero.

D. Data Segmentation and Augmentation

A key aspect of our approach is data segmentation. Since many recordings in the dataset are significantly longer than the desired input size for our neural network, we split each recording into fixed-length segments. Unlike conventional methods that discard excess audio, our segmentation approach retains all segments, ensuring maximum utilization of the dataset [10].

To further enhance model performance, we applied data augmentation techniques such as spectrum random masking, and adding Gaussian noise [11]. These augmentations increase the diversity of the training data and help the model generalize better to unseen samples.

E. Spectrogram Classification Using CNNs

Once the spectrograms were generated, we treated them as image-like inputs for classification. A Convolutional Neural Network (CNN) was employed to extract spatial and temporal features from the spectrograms. CNNs are particularly well-suited for this task as they can identify intricate patterns in the time-frequency domain, such as unique frequency modulations or harmonic structures characteristic of bird species.

Our model architecture was based on pre-trained EfficientNet [12] variants, which were fine-tuned for the task of birdsong classification. Using weights of pre-trained models can be highly beneficial in audio recognitions and can highly improve the convergence of the models [13]. These models provide a balance between computational efficiency and high accuracy, making them ideal for handling the challenges posed by the dataset, such as class imbalance and noisy samples.

V. Our Solution

In this section, we outline the pipeline used to develop our method, a machine learning model designed for birdsong classification. The pipeline consists of three major stages: raw data processing, dataset preparation, and the training and validation process. Each stage plays a critical role in transforming raw audio recordings into a robust model capable of identifying bird species.

Please note that our primary objective was not to create the simplest or quickest pipeline for solving the competition task. Instead, we focused on developing a modular and adaptable design that can be extended to more complex and diverse challenges. While the dataset provided for the competition is uniform in certain aspects (e.g., consistent sampling rates across all samples), we intentionally included checks for such properties in our implementation. This approach ensures that our pipeline remains robust and reusable for datasets that may not share the same uniformity.

A. Raw Data Processing

The raw data processing stage focuses on transforming audio recordings and associated metadata into a standardized format suitable for machine learning. This whole process is parallelized. This involves the following steps:

1) **Audio Loading**: The raw audio files are read and loaded one-by-one into memory using *TorchAudio*. This library allows for seamless integration with PyTorch and provides tools for resampling and waveform processing.
2) **Mono Conversion**: Recordings might come in stereo format. To simplify the processing and reduce computational requirements, we convert all stereo audio files into mono by averaging the channels.
3) **Segmenting and Padding**: Audio files are segmented into fixed-length chunks to ensure compatibility with the input requirements of the neural network. For recordings longer than the target duration, we use a sliding window approach (*slice mode*) to create multiple chunks. For shorter recordings, padding is applied using silence or repeating the edges.
4) **Mel-Spectrogram Generation**: The processed audio waveforms are converted into mel-spectrograms using *TorchAudio* or *Librosa*. While both libraries offer similar functionality, we found that using *TorchAudio* provides worse results compared to *Librosa*. Key parameters, such as the number of mel bins, window size, and hop length, are tuned to best capture the frequency patterns of bird calls. After generating the mel-spectrograms we normalize them.
5) **Standardization**: We added an optional standardization step.
6) **Saving and Metadata Construction**: The resulting mel-spectrograms are saved as `.npy` files to optimize data loading during training. A pandas DataFrame is constructed to store metadata such as the file path, label, and any additional information (e.g., recording location, duration), providing a structured format for the dataset.

B. Dataset Preparation

After raw data processing, the next step is to convert the organized metadata into a format that can be directly utilized by the model. This involves:

- **Custom Dataset Class**: A custom PyTorch `Dataset` class (`BirdClefDataset`) was implemented to read the mel-spectrogram files and their corresponding labels dynamically. This ensures that data is efficiently loaded during training without requiring all samples to be loaded into memory simultaneously. Additionally, the custom dataset allows us to implement augmentation methods that are applied upon indexing the dataset object.
- **DataLoader Creation**: A `DataLoader` is used to handle batching, shuffling, and parallel data loading. Multiple workers are assigned to load data, optimizing GPU utilization during training.

C. Training and Validation

The final stage involves training the neural network and evaluating its performance. Key steps include:

1) **Training Loop**: The training loop iterates over batches of data, optimizing the model weights using the chosen optimizer. Focal-loss [14] was employed as the primary loss function, though additional experiments with Cross-entropy loss are planned for future work.
2) **Validation Pipeline**: After each training epoch, the model is evaluated on the validation set. Key metrics, including accuracy and average loss, are computed to monitor performance and prevent overfitting.
3) **Model Checkpointing**: A model checkpointing mechanism saves the model with the best validation performance. This ensures that the final model represents the best state achieved during training.
4) **Test Evaluation**: Once training is complete, the model is tested on the held-out dataset to assess its generalization capabilities. The test results are compared[1] against benchmarks from the Kaggle BirdCLEF 2024 competition.

VI. Hyperparameter Optimization

We utilized the RayTune Python library to implement an efficient hyperparameter optimization framework. Our approach leverages the Asynchronous Hyperband Scheduler (ASHA), which ensures computational efficiency by terminating poorly performing configurations early and reallocating resources to promising ones. This strategy significantly reduces the time and computational cost of the optimization process.

The hyperparameter optimization explored 100 different configurations, each evaluated based on the validation accuracy of the model. Results were logged, and the best configuration was extracted and saved for reproducibility.

Although the optimization process provided valuable insights, it did not outperform the manually selected configuration derived from our domain knowledge and prior experimentation.

VII. Current Solutions

The BirdCLEF 2024 challenge has concluded, leaving behind a wealth of insights from participants worldwide. This section outlines the existing approaches to bird song classification and summarizes the public results achieved in the competition.

A. Existing Approaches to Birdsong Classification

The BirdCLEF challenge has traditionally inspired innovative solutions leveraging various machine learning techniques. Key trends observed in past competitions include ([15] [16] [17]).

Feature Engineering and Preprocessing: Many participants utilized advanced feature extraction methods, such as mel-spectrograms, along with domain-specific representations like

[1]We are planning to submit our solution to Kaggle as a late submission to get an accurate and fair comparison of our model's performance.

BirdNET embeddings [18]. Some teams applied pseudo-labeling to annotate unlabeled data, combining supervised and unsupervised learning techniques effectively

Model Architectures: The most successful submissions adopted deep learning frameworks, including convolutional neural networks and ensemble approaches. Test-time augmentation (TTA) was widely used to enhance model robustness by averaging predictions over multiple augmented versions of the test data.

Domain Adaptation Techniques: Considering the variations in the soundscape data (e.g., background noise, dialectal differences in bird calls), some teams implemented domain adaptation methods to align their models better to the test distribution. This included contrastive adversarial domain bottlenecks and using region-specific metadata to guide predictions.

B. Current Public Results

The competition evaluation was based on the macro-averaged ROC-AUC score, emphasizing balanced performance across species. The top submission achieved a public leaderboard score of 0.738 and a private leaderboard score of 0.690, showcasing significant improvements over baseline models [3].

VIII. Results and Discussion

In terms of results, we have found a good practice to solve the problem. The method we propose achieves better results in terms of ROC-AUC than the best solutions tested on the available database published on BirdCLEF.

The two methods we applied are shown in Fig. 4 and Fig. 5. The two investigated methods are ”single” and ”slice,” where the gray curve represents the validation accuracy and average validation loss during training for the ”single” approach, while the red curve corresponds to the ”slice” approach. It is clearly visible that the ”slice” approach achieves better results in terms of the evaluated accuracy and loss.

The results of our proposed methods and the best publicly available solution from the BirdCLEF 2024 competition are summarized in Table II. It is evident that even with the ”single” method, we achieved better performance compared to the winning solution. Furthermore, the ”slice” method enhanced

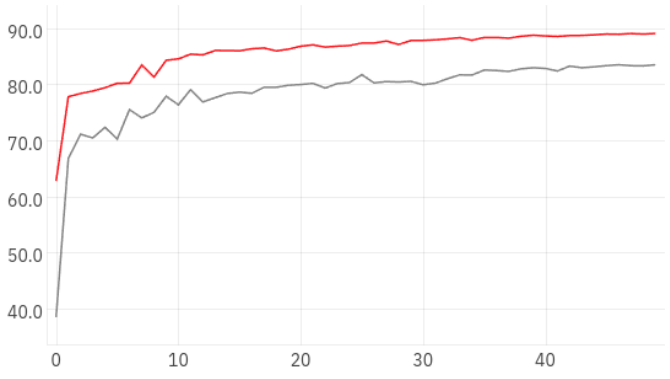

Fig. 4. Validation accuracy

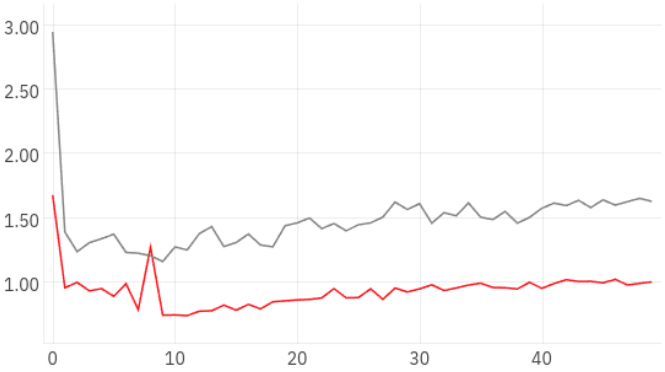

Fig. 5. Average validation loss

TABLE II
BEST RESULTS ACHIEVED WITH THE METHODS.

	Best of BirdCLEF	Single	Slice
Final metrics (AUC)	98.77	98.6303	99.4243

the robustness of the model, leading to even better results. Please note, that these are not the results from the official testing dataset (which is known only by the organizers), but from the test dataset established by the teams. We are currently working on a notebook to submit, to get the final results from the official evaluation of our model.

IX. Conclusion

In this study, we developed a method that improves the performance of classification for audio files through data augmentation.

By dividing the data into uniform segments and supplementing segments shorter than the desired length, it is possible to enhance classes with fewer samples. This type of preprocessing improves the model’s performance, as it is trained, validated, and tested with a larger number of samples than in the original dataset. Consequently, with a larger dataset, the model is more likely to learn the unique characteristics of the different classes, allowing it to solve the classification task with excellent results.

The methodology outlined in this study holds promise for several future applications. First, it can be adapted to other domains involving time-series data, such as speech recognition, music genre classification. The segmentation and augmentation techniques can also benefit tasks where data imbalance is a critical issue, such as rare event detection.

X. Acknowledgments

We would like to thank the organizers of the BirdCLEF 2024 competition for providing the dataset, which served as the foundation for this work.

We utilized generative artificial intelligence to assist with various aspects of this project, including code commenting and writing, drafting the `README.md` file, translating these

texts into English, and providing explanations for the usage of certain Python library functions. While the AI played a limited role in the first two tasks, it was moderately relied upon for the translation and offered valuable insights for understanding specific library functionalities.

REFERENCES

[1] Alexis Joly, Lukas Picek, Stefan Kahl, Hervé Goëau, Vincent Espitalier, Christophe Botella, Benjamin Deneu, Diego Marcos, Joaquim Estopinan, César Leblanc, Théo Larcher, Milan Sulc, Marek Hruz, Maximilien Servajean, Jiri Matas, Hervé Glotin, Robert Planqué, Willem-Pier Vellinga, Holger Klinck, and Henning Müller. *LifeCLEF 2024 Teaser: Challenges on Species Distribution Prediction and Identification*, pages 19–27. 03 2024.

[2] Holger Klinck, Maggie, Sohier Dane, Stefan Kahl, Tom Denton, and Vijay Ramesh. Birdclef 2024. https://kaggle.com/competitions/birdclef-2024, 2024. Kaggle.

[3] Stefan Kahl, Tom Denton, Holger Klinck, Vijay Ramesh, Viral Joshi, Meghana Srivathsa, Akshay Anand, Chiti Arvind, Harikrishnan Cp, Suyash Sawant, Hervé Glotin, Hervé Goëau, Willem-Pier Vellinga, Robert Planqué, and Alexis Joly. Overview of BirdCLEF 2024: Acoustic Identification of Under-studied Bird Species in the Western Ghats. In *CLEF 2024 Working Notes - 25th Conference and Labs of the Evaluation Forum*, volume 3740 of *CEUR workshop proceedings*, pages 1948–1957, Grenoble, France, September 2024.

[4] Jin Huang and C.X. Ling. Using auc and accuracy in evaluating learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 17(3):299–310, 2005.

[5] MA Raghuram, Nikhil R Chavan, Ravikiran Belur, and Shashidhar G Koolagudi. Bird classification based on their sound patterns. *International journal of speech technology*, 19:791–804, 2016.

[6] Mercity AI. Short time fourier transform and spectrograms. *Audio Analysis Techniques*, 2024. Accessed: 2024-12-07.

[7] An Zhao, Krishna Subramani, and Paris Smaragdis. Optimizing short-time fourier transform parameters via gradient descent. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 736–740. IEEE, 2021.

[8] UCSD Noise Lab. Audio recognition using mel spectrograms and convolution neural networks. *ECE228 2019 Reports*, 2019. Accessed: 2024-12-07.

[9] Yeongtae Hwang, Hyemin Cho, Hongsun Yang, Insoo Oh, and Seong-Whan Lee. Mel-spectrogram augmentation for sequence to sequence voice conversion. *CoRR*, abs/2001.01401, 2020.

[10] Muhammad Huzaifah. Comparison of time-frequency representations for environmental sound classification using convolutional neural networks. *arXiv preprint arXiv:1706.07156*, 2017.

[11] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. Specaugment: A simple data augmentation method for automatic speech recognition. In *Interspeech 2019*, interspeech$_2$019.$ISCA$, $September$2019.

[12] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks, 2020.

[13] Eleni Tsalera, Andreas Papadakis, and Maria Samarakou. Comparison of pre-trained cnns for audio classification using transfer learning. *Journal of Sensor and Actuator Networks*, 10(4), 2021.

[14] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection, 2018.

[15] Anthony Miyaguchi, Adrian Cheung, Murilo Gustineli, and Ashley Kim. Transfer learning with pseudo multi-label birdcall classification for ds@gt birdclef 2024, 2024.

[16] Mario Lasseck. Bird species recognition using convolutional neural networks with attention on frequency bands. In *CLEF (Working Notes)*, pages 2071–2079, 2023.

[17] Anthony Miyaguchi, Nathan Zhong, Murilo Gustineli, and Chris Hayduk. Transfer learning with semi-supervised dataset annotation for birdcall classification. *arXiv preprint arXiv:2306.16760*, 2023.

[18] Haohe Liu, Xubo Liu, Qiuqiang Kong, Wenwu Wang, and Mark D Plumbley. Learning temporal resolution in spectrogram for audio classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 13873–13881, 2024.