Acoustic Bird Species Recognition at BirdCLEF 2023: Training Strategies for Convolutional Neural Network and Inference Acceleration using OpenVINO

Lihang Hong [1]

[1] *Accenture Japan Ltd, Akasaka Intercity 1-11-44 Akasaka, Minato-ku, Tokyo, 107-8672, Japan*

Abstract

Monitoring of bird species plays a vital role in understanding biodiversity trends, as birds serve as reliable indicators of ecological change. Traditional observer-based bird surveys are often resource-intensive and logistically challenging, prompting the need for advanced technological solutions. In this study, we explored the use of Convolutional Neural Networks (CNNs) for feature extraction and classification, along with training strategies that maximize the performance of these models given limited training data. Furthermore, we evaluate the implementation of the OpenVINO toolkit to accelerate the inference speed. Our goal is to establish a reliable classification model that can, with limited training data, recognize bird species by their calls in real time. The solution based on the study achieved 2nd rank among 1189 teams at BirdCLEF 2023 challenge hosted in Kaggle.

Keywords

BirdCLEF2023, audio, bird species recognition, Convolutional Neural Network, Sound Event Detection, OpenVINO

1. Introduction

The rapid decline in global biodiversity has become a significant concern in recent years, putting numerous species at risk of extinction and threatening the stability of ecosystems. As birds serve as important indicators of biodiversity change, monitoring their populations is essential. Traditional bird surveys, which primarily rely on direct observation and human expertise, can be resource-intensive and face logistical challenges when applied at large scales and high temporal resolutions. This highlights the need for more efficient, scalable, and cost-effective methods to monitor bird populations. Advancements in passive acoustic monitoring (PAM) technology, combined with innovative machine learning algorithms, present a promising solution to these challenges.

The aim of BirdCLEF 2023 [1, 2] is to identify Eastern African bird species by sound, which is a pilot work of testing the effect of various management regimes and states of degradation on bird biodiversity in rangeland systems around Northern Mount Kenya. This is done with the aim of demonstrating the efficacy and cost-effectiveness of using machine learning algorithms in measuring the success of restoration projects. Ultimately, the goal is to achieve large-scale restoration and protection of the planet in a cost-effective manner.

2. Related Work

Challenges in training machine learning models with audios to identify bird species are [3]:

[1]CLEF 2023: Conference and Labs of the Evaluation Forum, September 18–21, 2023, Thessaloniki, Greece
EMAIL: rihanneko@gmail.com (A. 1)
ORCID: 0009-0006-7840-7857 (A. 1)

CEUR Workshop Proceedings (CEUR-WS.org)

1. Weak labels. The main challenge is that given an audio, we have no information about where the bird call appears. When we clip the audio, there is a chance that the audio clip does not contain bird call, which introduce noise to the training process.
2. The gap between the bird call in short audios(training data) and that in long soundscapes(test data). Short audios usually focus on one certain species and the bird call appears in the foreground. However, in soundscape, usually there are several species speaking over each other in the background. Making the classification model trained on short audios applicable to soundscape is very important because scientists need to identify birds recorded in a relatively noisy environment, while short audios are cost-effective as training data.
3. Long-tailed and imbalanced distribution. Rare species have less training audios available, while major species have many available audios. Classification model trained with imbalanced dataset may give a poor performance when classifying rare species.

In previous BirdCLEF challenges [4, 5, 6], state-of-the-art solutions transform the raw audio to Mel-spectrograms and train with Deep Convolutional Neural Networks (CNNs), treating the task as an image classification problem. In addition, ensembles and post-processing techniques are usually implemented.

To deal with weak labels, researchers proposed model which can be trained with longer audio clips by combining Convolutional Neural Networks (CNNs) with simple pooling layer on time and frequency dimension [7, 8]. Other approaches like Sound Event Detection (SED) employs two-dimensional CNNs to extract time and frequency information from audio samples, then the information is processed by an attention head to calculate the probability of the appearance of birds over the time dimension [9].

To deal with the gap between short audios and soundscape, data Augmentation and pretraining were implemented. Adding environmental sound without bird calls as background noise [10, 11] and Mixup are considered most effective. Researchers implemented Mixup [12] on both audio and Mel-spectrograms to mix different bird calls into one training sample in order to simulate the soundscape.

To deal with imbalanced distribution, oversampling the rare species by splitting one short audio to several audio clips are implemented [8].

In the previous BirdCLEF challenges, soundscapes are allowed to compute with GPU within 9 hours of inference time. In BirdCLEF 2023, soundscapes are only allowed to compute with CPU within 2 hours of inference time. This change encouraged a focus on efficient models with a good balance between accuracy and speed, which can be used in the real field work.

3. Methods

In this section, we explains the main components of our solution to the BirdCLEF 2023 Challenge.

3.1. Dataset

As in previous BirdCLEF challenges, training data is provided by the Xeno-canto [13] community. More than 16000 audios covering 264 species are provided by the competition host.

To further expand the dataset size, we collected additional 21000 audios which from Xeno-canto community. Besides the audios in which the target species appear in foreground, which we call them foreground audios, audios with duration less than 60 seconds in which the target species only appear in background, which we call them background audios, were also included.

For pretraining, audios from previous BirdCLEF challenges were included [4, 5, 6]. The total dataset size was about 119000 covering 834 species.

3.2. Evaluation

The evaluation metric for this challenge is padded cmAP, a derivative of the macro-averaged average precision score as implemented by scikit-learn. The prediction data and target data of each species are padded with five rows of true positives, which makes it possible for the metrics to accept zero positive labels and less influenced by the species with very few positive labels.

3.3. Preprocessing

10-20 second audio clip randomly selected from raw audio is converted to Mel-spectrograms using librosa library [14]. For background audios, to ensure that the target species appear in the audio clip, we first clipped 60 seconds from the audio, cut it into for example, six 10 second audio clips and summed up to one 10 second mixed audio clip. After converting the audio clip to Mel-spectrograms, Deltas and Delta-deltas are calculated as additional input feature using torchaudio library [15].

3.4. Augmentations

7 types of audio augmentations implemented using audiomentations [16] are as follows:

1. GaussianNoise: This technique involves adding random Gaussian noise to the audio signal.
2. PinkNoise: Noise with a power spectral density inversely proportional to frequency.
3. Gain: Gain is used to adjust the overall volume of the audio signal.
4. Background Noise: Adding background noise simulates the presence of other sounds in the environment, such as wind, rain, or human activity [10, 11].
5. PitchShift: Pitch shifting changes the pitch of the audio signal without altering its duration.
6. TimeShift: Moving the audio signal in time, without changing its pitch or duration.
7. OR Mixup: Compared to classic Mixup [12], OR Mixup uses the formula as follows:

$$x = x_i + (1 - \tau)x_j \quad (1)$$

$$y = y_i + (1 - \tau)y_j \quad (2)$$

where (x_i, y_i) and (x_j, y_j) are the two randomly selected samples and τ is Mixup ratio.

For Mel-spectrograms, Frequency Masking and Time Masking [17] were implemented using torchaudio. Classic Mixup was also implemented to the Mel-spectrograms.

3.5. Model Architecture

We used Custom CNN and Sound Event Detection Model, proposed in [7] and [9].

3.5.1. Sound Event Detection Model (SED)

As we can see in Figure 1, this model employs two-dimensional CNNs to extract and process time and frequency information from audio samples, then the information is processed by an attention head to calculate the probability of the appearance of birds over the time dimension.

We trained this model using 10 seconds audio clips randomly selected from the audio. For inference, we used 10 seconds audio clip, in which the 5 seconds to predict were in the center of the audio clip. With this implementation, we can make prediction with attention layer on the central 5 seconds of the deep features encoded by CNN, which contains extra global information useful for prediction.

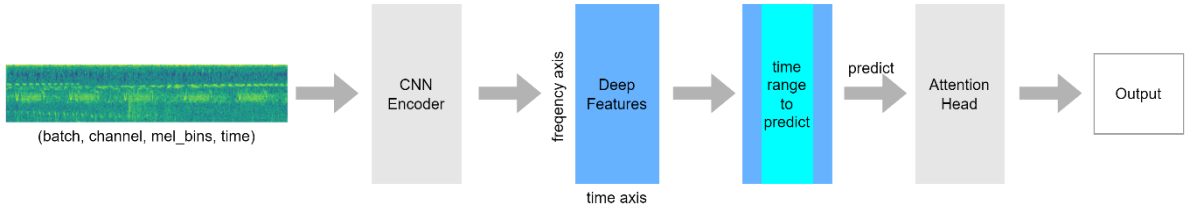

Figure 1: Model Architecture of SED

3.5.2. CNNs with simple pooling layer (Custom CNN)

As we can see in Figure 2, this model splits the Mel-spectrogram along the time axis and extract deep features on each splitted Mel-spectrogram. After that, a GeM layer implemented to apply pooling

on time and frequency to gather the overall information of each deep feature and create an embedding. Then the embedding is computed by a linear head to generate probabilities for each species. With this architecture, the model can be trained with long audio clip to deal with absence of bird call in short audio clip, while being able to make prediction on short audio clip.

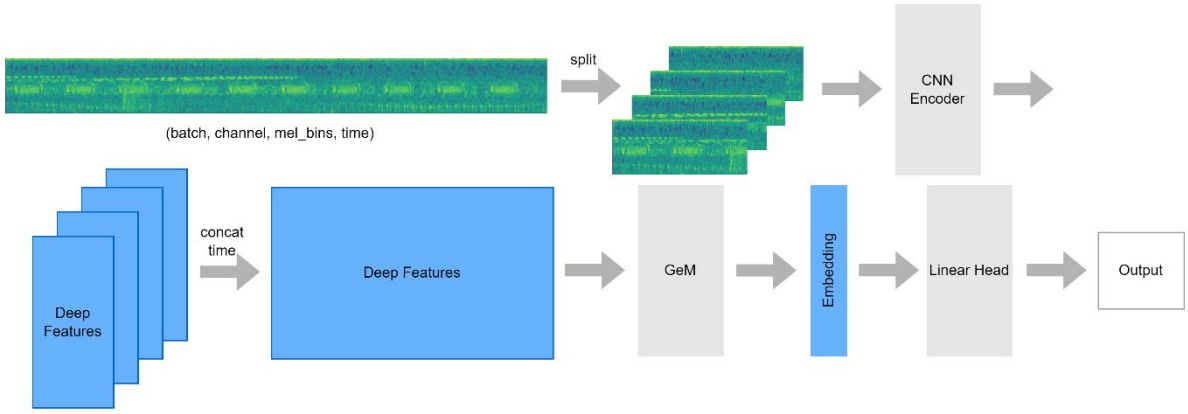

Figure 2: Model Architecture of Custom CNN

We trained this model using 15 to 20 seconds audio clips randomly selected from the audio. For inference, we used 5 seconds audio clips.

3.6. Training Details

Virtual environment on Google Colaboratory with an A100 GPU has been used for training. We used Pytorch to train our models and the CNN encoders were provided by timm library [18]. For training, a two-stage approach was utilized. Initially, pretraining was conducted on the entire dataset comprising 834 species. Subsequently, fine-tuning was performed on the subset of 264 species. Throughout both of these stages, the model was trained using CrossEntropyLoss (CE Loss) prior to employing BCEWithLogitsLoss (BCE Loss). The model exhibited a more rapid convergence with CE Loss compared to BCE Loss. However, the latter proved to yield higher performance.

For validation, we randomly selected about 4000 audios covering 264 species as validation subset. The number of each species in the validation subset was in proportion to that in the whole dataset. We computed the Padded cmAP as Cross Validation score (CV) on the first 60 seconds of the audio.

To deal with the class imbalance of the dataset, we used WeightedRandomSampler [19] to sample the audios of each species according to a uniform distribution.

To enhance the diversity of the models in ensemble process, models were developed based on Mel-spectrogram generated by varied parameters, with some trained utilizing CE Loss only, and with some trained without pretraining process. Moreover, three of the models' head layers were additionally fine-tuned on 30-second audio clips. Details of Mel-spectrogram parameters for each model are shown in Table 1 and details of other training conditions mentioned above are shown in Table 2.

Table 1
Varied Mel-spectrogram parameters were set for each model to make difference in input train data.

Model type	CNN encoder	Mel bins	Frequency	Window	Hop length
SED	EfficientNetV2-s	128	(0 Hz, 16000 Hz)	2048	417
SED	EfficientNet-b3-ns	128	(50 Hz, 14000 Hz)	1024	535
SED	SeResnext26t-32x4d	128	(0 Hz, 16000 Hz)	2048	627
Custom CNN	EfficientNetV2-s	64	(50 Hz, 14000 Hz)	1024	320
Custom CNN	EfficientNet-b3-ns	128	(50 Hz, 14000 Hz)	1024	535
Custom CNN	EfficientNet-b0-ns	128	(0 Hz, 16000 Hz)	2048	627
Custom CNN	ResNet34d	128	(0 Hz, 16000 Hz)	2048	627

Table 2
To further enhance the diversity of models, different training conditions were set for each model.

Model type	CNN encoder	Loss funtion	Pretrained	30s fine-tined
SED	EfficientNetV2-s	CE Loss and BCE Loss	Yes	Yes
SED	EfficientNet-b3-ns	CE Loss and BCE Loss	Yes	Yes
SED	SeResnext26t-32x4d	CE Loss	Yes	No
Custom CNN	EfficientNetV2-s	CE Loss and BCE Loss	Yes	Yes
Custom CNN	EfficientNet-b3-ns	CE Loss and BCE Loss	Yes	No
Custom CNN	EfficientNet-b0-ns	CE Loss	No	No
Custom CNN	ResNet34d	CE Loss and BCE Loss	Yes	No

3.7. Inference Acceleration using OpenVINO

In inference process, we generated predictions on each 5 seconds of the soundscape with 7 models and ensembled the predictions using weighted average method.

To accelerate the inference speed, we used OpenVINO toolkit [20]. OpenVINO is a comprehensive toolkit developed by Intel to facilitate the development and deployment of deep learning models for various applications. The toolkit supports several deep learning frameworks, such as TensorFlow, Caffe, and ONNX. For Pytorch framework, we first converted the models to ONNX format and then converted the ONNX model to OpenVINO format.

4. Results

4.1. Experimental Results of Training Strategies

Padded cmAP was calculated as the metrics in BirdCLEF 2023 challenge's Leaderboard, denoted as LB which consists of two variants of public and private. Table 3 presents the experimental results of training strategies. In our experiment, increasing dataset size and applying OR Mixup on audios significantly improved the LB of single model. In addition, although class balanced sampling did not increase the LB of single model, it increased the LB of ensemble prediction.

Table 3
Experimental results of training strategies. Effective training strategies include increasing dataset size(No.2), adding background noise(No.3), OR Mixup(No.4), class balanced sampling(No.5) and ensemble(No.7 and No.8).

No	Models	CNN encoder	CV	Public LB	Private LB
1	SED (baseline)	EfficientNetV2-s	0.86547	0.81257	0.71667
2	SED (1 + additional audios)	EfficientNetV2-s	0.86741	0.81725	0.72584
3	SED (2 + more background noise)	EfficientNetV2-s	0.86484	0.81935	0.73047
4	SED (3 + OR Mixup)	EfficientNetV2-s	0.86053	0.82349	0.73141
5	SED (4 + class balanced sampling)	EfficientNetV2-s	0.85610	0.82209	0.73008
6	Custom CNN (additional audios included)	ResNet34d	0.86511	0.81127	0.71187
7	Ensemble (4 + 6)	-	-	0.82922	0.73869
8	Ensemble (5 + 6)	-	-	0.83110	0.74318

From Table3, we can see that adding more background noise to SED model improves the performance, implying that heavy background noise augmentation can improve the robustness of the model. To further investigate the effect of background noise, additional experiment was conducted on Custom CNN with EfficientNetV2-s encoder. The results of the experiment are presented in table 4. Comparing the results in Table3 and Table 4, we can see that the impact of adding background noise actually varies. Adding background noise improves the performance of SED model while worsening

the performance of Custom CNN. In addition, with the same training dataset, SED model performs better than Custom CNN in the test set. The result indicates that model architecture with attention mechanism may be more robust to the complex acoustic environments in the real world.

Table 4
Experiment on background noise augmentation. The value in () shows the LB of SED model trained on whole dataset. The effect of aggressive background noise varies between SED and Custom CNN.

No	Models	CNN encoder	Public LB	Private LB
1	SED (trained on training subset)	EfficientNetV2-s	0.81725	0.72584
2	SED (1 + more background noise)	EfficientNetV2-s	0.81935 (0.82643)	0.73047 (0.73754)
3	Custom CNN (trained on whole dataset)	EfficientNetV2-s	0.82162	0.73684
4	Custom CNN (3 + more background noise)	EfficientNetV2-s	0.81823	0.73426

4.2. Inference time with OpenVINO toolkit

Inference was conducted on Kaggle CPU environment. The inference time estimation of 10 minutes soundscape with EfficientNetV2-s based SED model in Pytorch format and that with OpenVINO format is presented in Figure 3. We can reduce inference time by about 45% with OpenVINO toolkit.

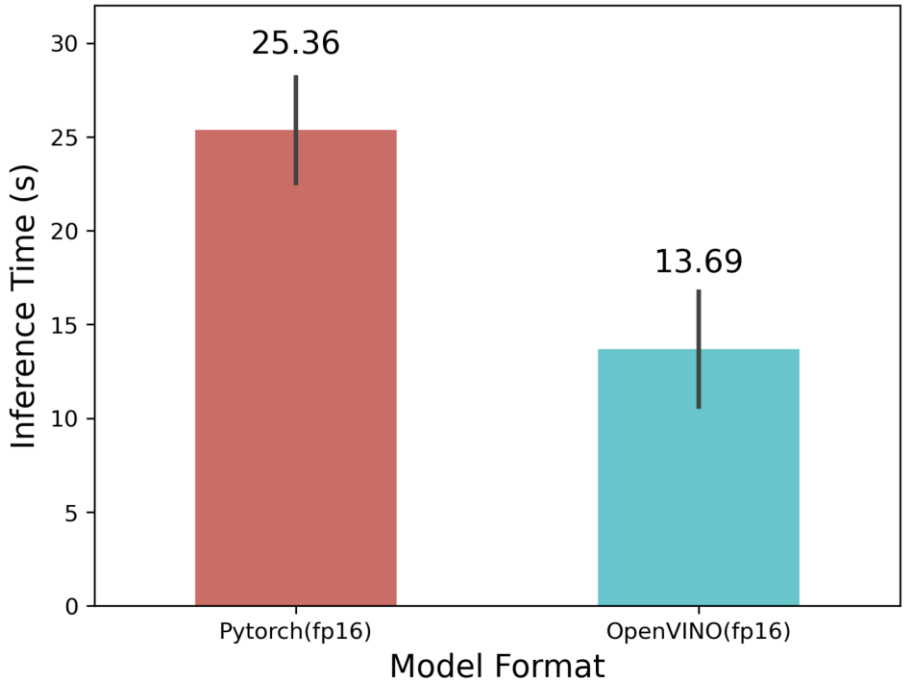

Figure 3: Inference time estimation of Pytorch model and OpenVINO model on 10 minutes soundscape, with 50 loops for each model.

4.3. Performance of Final Submission Models

The performance of single model for final submission and ensemble result are listed in Table 5. The final submission achieved 2nd rank among 1189 teams at BirdCLEF 2023. The top 2 best-performing models are SED with EfficienNetV2-s encoder and Custom CNN with ResNet34d encoder.

Table 5
Single model performance of final submission models and the ensemble result.

Model type	CNN encoder	Public LB	Private LB
SED	**EfficientNetV2-s**	**0.82643**	**0.73754**
SED	EfficientNet-b3-ns	0.82534	0.73602
SED	SeResnext26t-32x4d	0.81947	0.72688
Custom CNN	EfficientNetV2-s	0.82070	0.73681
Custom CNN	EfficientNet-b3-ns	0.81652	0.71963
Custom CNN	EfficientNet-b0-ns	0.81564	0.71734
Custom CNN	**ResNet34d**	**0.82593**	**0.73985**
Ensemble	-	**0.84123**	**0.76369**

5. Conclusion and future work

This study demonstrates the effectiveness of employing Convolutional Neural Networks and effective training strategies for recognizing bird species in complex acoustic environments. By expanding the dataset, applying various augmentation techniques, and utilizing different model architectures, we were able to enhance the model's performance and mitigate challenges presented by weak labels, gap between train and test audios, and imbalanced data distribution. Our experiments suggest that model architecture containing attention layer is more robust to the environmental noise and is better suited for recognizing bird species in complex acoustic environments of the real world.

Furthermore, the implementation of OpenVINO toolkit substantially accelerated the inference speed, highlighting the potential for real-time bird species recognition in biodiversity monitoring applications.

Future work may include experiments on more diverse datasets and encoders. In order to better gauge the generalizability of our models, performance should be evaluated on more diverse and challenging datasets, including recordings from different geographical regions, seasons, and habitats, as well as those containing rare or endangered species. The effect of the training strategies should be evaluated on other CNN encoders, as well as Vision Transformers. Integration with IoT devices and real-time monitoring systems is another challenging future work for the ultimate goal to achieve the vision of restoring and protecting the planet at scale.

6. References

[1] A. Joly, C. Botella, L. Picek, S. Kahl, H. Goëau, B. Deneu, D. Marcos, J. Estopinan, C. Leblanc, T. Larcher, R. Chamidullin, M. Šulc, M. Hrúz, M. Servajean, H. Glotin, R. Planqué, W.-P. Vellinga, H. Klinck, T. Denton, I. Eggel, P. Bonnet, H. Müller, Overview of LifeCLEF 2023: evaluation of ai models for the identification and prediction of birds, plants, snakes and fungi, in: International Conference of the Cross-Language Evaluation Forum for European Languages, Springer, 2023.

[2] Kahl, S., Denton, T., Klinck, H., Reers, H., Cherutich, F., Glotin, H., Goëau, H., Vellinga, W.P., Planqué, R., Joly, A.: Overview of BirdCLEF 2023: Automated bird species identification in Eastern Africa. Working Notes of CLEF 2023 – Conference and Labs of the Evaluation Forum (2023).

[3] M. V. Conde, U.Choi, Few-shot Long-Tailed Bird Audio Recognition. Working Notes of CLEF 2022 – Conference and Labs of the Evaluation Forum (2022).

[4] S. Kahl, M. Clapp, W. Hopping, H. Goëau, H. Glotin, R. Planqué, W.-P. Vellinga, A. Joly, Overview of birdclef 2020: Bird sound recognition in complex acoustic environments (2020).

[5] S. Kahl, T. Denton, H. Klinck, H. Glotin, H. Goëau, W.-P. Vellinga, R. Planqué, A. Joly, Overview of birdclef 2021: Bird call identification in soundscape recordings, in: Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum (2021).

[6] S. Kahl, A. Navine, T. Denton, H. Klinck, P. Hart, H. Glotin, H. Goëau, W.-P. Vellinga, R. Planqué, A. Joly, Overview of birdclef 2022: Endangered bird species recognition in soundscape recordings, Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum (2022).
[7] C. Henkel, P. Pfeiffer, P. Singer, Recognizing bird species in diverse soundscapes under weak supervision, 2021. URL: https://arxiv.org/abs/2107.07728. doi:10.48550/ARXIV. 2107.07728.
[8] E. Martynov, Y. Uematsu, Dealing with Class Imbalance in Bird Sound Classification. Working Notes of CLEF 2022 – Conference and Labs of the Evaluation Forum (2022).
[9] S. Adavanne, A. Politis, J. Nikunen, T. Virtanen, Sound event localization and detection of overlapping sources using convolutional recurrent neural networks, IEEE Journal of Selected Topics in Signal Processing 13 (2018) 34–48. URL: https://ieeexplore.ieee.org/ abstract/document/8567942. doi:10.1109/JSTSP.2018.2885636.
[10] Stowell, M. D. Plumbley, freefield1010 - an open dataset for research on audio field recording archives, in: Proceedings of the Audio Engineering Society 53rd Conference on Semantic Audio (AES53), Audio Engineering Society, 2014.
[11] V. Lostanlen, J. Salamon, A. Farnsworth, S. Kelling, J. Bello, Birdvox-full-night: A dataset and benchmark for avian flight call detection, ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, Institute of Electrical and Electronics Engineers Inc., 2018, pp. 266–270. doi:10.1109/ICASSP.2018.8461410.
[12] H. Zhang, M. Cissé, Y. N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, CoRR abs/1710.09412 (2017). URL: http://arxiv.org/abs/1710.09412. arXiv:1710.09412.
[13] Xeno-canto, Xeno-canto: Sharing bird sounds from around the world, 2022. URL: https://xeno-canto.org.
[14] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, O. Nieto, librosa: Audio and music signal analysis in python, in: Proceedings of the 14th python in science conference, volume 8, 2015.
[15] Y.-Y. Yang, M. Hira, Z. Ni, A. Chourdia, A. Astafurov, C. Chen, C.-F. Yeh, C. Puhrsch, D. Pollack, D. Genzel, D. Greenberg, E. Z. Yang, J. Lian, J. Mahadeokar, J. Hwang, J. Chen, P. Goldsborough, P. Roy, S. Narenthiran, S. Watanabe, S. Chintala, V. Quenneville-Bélair, Y. Shi, Torchaudio: Building blocks for audio and speech processing, arXiv preprint arXiv:2110.15018 (2021).
[16] I. Jordal, A. Tamazian, E. T. Chourdakis, C. Angonin, T. Dhyani, askskro, N. Karpov, O. Sarioglu, BakerBunker, kvilouras, E. B. Çoban, F. Mirus, J.-Y. Lee, K. Choi, MarvinLvn, SolomidHero, T. Alumäe, iver56/audiomentations: v0.30.0, 2023. URL: https://zenodo.org/record/7885479. doi: 10.5281/zenodo.7885479.
[17] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, Q. V. Le, SpecAugment: A simple data augmentation method for automatic speech recognition, in: Interspeech 2019, ISCA, 2019. URL: https://doi.org/10.21437%2Finterspeech.2019-2680. doi:10.21437/ interspeech.2019-2680.
[18] R. Wightman, N. Raw, A. Soare, A. Arora, C. Ha, C. Reich, F. Guan, J. Kaczmarzyk, mrT23, Mike, SeeFun, contrastive, M. Rizin, H. Kim, C. Kertész, D. Mehta, G. Cucurull, K. Singh, hankyul, Y. Tatsunami, A. Lavin, J. Zhuang, M. Hollemans, M. Rashad, S. Sameni, V. Shults, Lucain, X. Wang, Y. Kwon, Y. Uchida, rwightman/pytorch-image-models: v0.8.10dev0 Release. URL: https://zenodo.org/record/7618837. doi:10.5281/zenodo.7618837.
[19] D. S. pytorch, Imbalanced dataset sampler, 2022. URL: https://github.com/ufoym/imbalanced-dataset-sampler.
[20] Intel Corporation, Release Notes for Intel® Distribution of OpenVINO™ Toolkit 2023. URL: https://www.intel.com/content/www/us/en/developer/articles/release-notes/openvino-relnotes.html.