BirdCLEF 2023

Project Report

Course:

Applied Machine Learning

Project Goal:

Identify bird calls of Eastern African bird species to monitor avian biodiversity and measure the impact of restoration interventions

Project category:

Sound detection and classification

Proposed by:

Aziza Ben Tanfous; n° 27156

Johanna Rauberger; n° 27492

Proposed to:

Professor Manuel Campagnolo

2022-2023

Index

Introduction..... 1
Data..... 2
Methods..... 2
Discussion & Conclusion.....5
References.....7
Contributions.....8

Introduction

Biodiversity loss is a major global issue that has significant consequences for ecosystems and human well-being *(Chapin III et al., 2000; Sura, 2020)*. To help with conservation efforts and keep track of biodiversity in important ecosystems, we participated in the BirdCLEF 2023 - Birdcall Identification competition. The objective of that competition and the goal of our project was the development of a computer program that can accurately identify bird species in Eastern Africa based on their calls. Using audio analysis and machine learning (ML) methods to build a bird call classifier, we aim to contribute to protecting biodiversity and evaluating the effectiveness of restoration projects.

Why birds? Birds are suitable indicators of changes in biodiversity: they are high up in the food chain, move across different areas and have diverse habitat needs. So, monitoring changes in bird populations and species assemblage allows to comprehensively track biodiversity and in particular evaluate the success or failure of restoration projects – when decision-makers know if and how well current restoration projects are working, they can take better targeted conservation measures in future.

However, traditional methods of surveying birds in extensive areas are challenging and costly. The approach presented in this report leverages modern technologies and machine learning to develop a classifier that detects bird calls in soundscapes and reliably identifies the bird species based on their unique sounds, as a means to make bird monitoring easier and more affordable. We hope it will be a useful contribution to the field of avian biodiversity monitoring and support ongoing efforts to protect and conserve biodiversity in Eastern Africa.

This report contains a detailed account of the project described, including information about the data used, the steps performed to prepare and preprocess them, the models selected, our analysis of the results, discussion and conclusion as well as the individual contributions of each team member.

Data

The data used were provided within the competition and can be found here. The competition includes 264 classes of birds. The training data are short recordings of individual bird calls with maximum frequency of 32kHz, converted to *ogg audio format*[1]. As the organizers stated that the provided training data is quite complete and that no benefit was expectable to looking for more, we refrained from doing so.

The metadata for the training data are contained in a csv file *(train_metadata.csv*), where the following fields/columns are considered the most directly relevant:

- primary_label – a code[2] for the bird species.
- latitude and longitude – the coordinates for where the recording was taken.
- author – the user who provided the recording.
- filename – the name of the associated audio file.

Further provided was a csv file containing data on the relationships between different species *(eBird_Taxonomy_v2021.csv).*

The competition uses a *hidden test*, i. e. the test data to be used for scoring are not provided. It consists of approximately 200 soundscape recordings of 10 min length in ogg audio format with randomized file names. Only when the notebook gets scored, the actual test data (including a sample submission) will be made available to the notebook (in the directory test_soundscapes).

Methods

After importing the data to our kaggle notebook, we began by thoroughly exploring and cleaning the dataset to ensure its quality and understand its structure. We carefully examined the metadata, checked dimensions and data types, and analyzed unique labels to get an overview over the bird species present in the dataset. To visualize the distribution of species across different latitudes and their relationship with geography, we used statistical summaries and visualizations such as histograms and scatter plots. We also made sure to not have any duplicates for maintaining the integrity of the data.

[1] details about the ogg audio format can be found here
[2] to learn more about the codes, see https://ebird.org/home

Once the dataset was ready, we prepared functions containing all necessary pre-processing and augmentation steps to be applied to the training data. We experimented with waveform-based and spectrogram-based approaches, but eventually chose to go with the spectrograms in order to save memory/computational resources. The audio processing steps applied to the data, mainly based on functionalities of the module *torchaudio*, are collected as static methods in the class *AudioUtil()*. These include, among others, the adjustment of the given audio signal to the required number of audio channels, the resampling of the audio to the required sample rate and the padding or truncating of the audio signal to a fixed length for consistency in the model input, the framing of the audio into smaller, overlapping frames and the generation of the spectrogram. It also includes augmentation steps, namely the application of a random time shift to the audio signal and the masking out of sections of the spectrogram to prevent overfitting. Then, a second class – *SoundDS()* – was utilized to handle data loading and the preprocessing and augmentation steps defined before.

To address the audio classification problem, we used a custom PyTorch model[3] using convolutional neural networks (CNN) and a linear classifier to classify the spectrograms of the sound recordings. The model architecture includes four convolutional blocks for extracting hierarchical features from the audio spectrograms with varying output channel sizes and kernel sizes. The weights of the convolutional layers are initialized using the Kaiming initialization method. ReLU activation functions and batch normalization are applied after each convolutional block to enhance the model's learning capabilities. An adaptive average pooling layer is then used to reduce the spatial dimensions to 1x1. The output of the pooling layer is flattened and passed through a linear layer with an input size of 64 and an output size of 264.

To efficiently handle the training data, we split the dataset into training and validation sets and created a data loader for the effective feeding of data to the model during the training process. In the model training loop, components and hyperparameters such as the loss function (*CrossEntropyLoss)*, optimizer (*Adaptive Moment Estimation (Adam)*[4] with a learning rate of 0.001), and a learning rate scheduler (*OneCycleLR*[5]) are defined and the

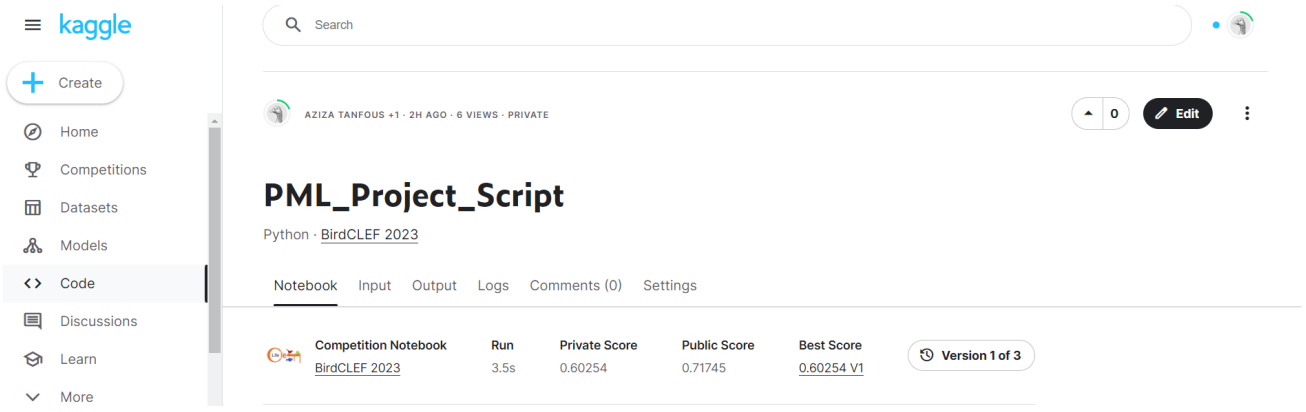
[3] built in the class named "*AudioClassifier*" which was shared within the competition
[4] based on gradient descent
[5] The *OneCycleLR scheduler* cyclically sadjust the learning rate during training to improve model convergence and generalization.

forward and backward passes are performed for each batch to optimize the model's parameters. The model's progress was monitored continuously by calculating the loss and accuracy, and we printed loss and accuracy at regular intervals (every 10 mini-batches) and computed and displayed the average loss and accuracy at the end of each epoch to track the training progress.

Finally, the audio samples were processed using the previously defined functions and then fed to the trained classifier to generate the desired predictions for the presence of bird species in each frame.

Results

The results produced (for each provided sample the predicted probability of belonging to each class bird species) were stored in tabular form inside a csv file "submission.csv". Birdclef2023 competition uses hidden test data to evaluate the submitted work, so the csv file was submitted[6] on Kaggle and evaluated using the padded cmAP metric, "a derivative of the macro-averaged average precision score as implemented by scikit-learn". To account for certain species and minimize the impact of limited positive labels, submissions and solutions are padded with five rows of true positives prior to scoring, so even a basic submission will receive a relatively high score. Following the submission, our private score was 0.60254 (calculated using around 80% the test data), whereas our public score achieved was 0.71745 (calculated using around 20% the test data) , as shown in the figure below.



Screenshot: BirdCLEF2023 Submission result (Scores)

[6] Due to the competition deadline being already in May, we were not able to submit our results in time to officially participate, but we could submit our results anyway to receive a score for our notebook.

Unfortunately, we could not compute the performance metrics like accuracy, precision, recall or provide the confusion matrix of our model due to the lack of the test data. As stated before, the competition's test data is kept hidden as the evaluation is done through submitting the notebook in the Kaggle platform.

Discussion & Conclusion

Let’s discuss our achieved results: Generally, we can happily say that the developed classifier does what it is supposed to do. Although we would not have won the competition, even if we had submitted our notebook more timely, the performance is not bad according to the scores we obtained.

In comparison with the scores of other notebooks we would have ranked average at best with our submission. But considering that we submitted predicted results after training the model on only two epochs, and that we were competing against 1188 other teams all over the world, often consisting of people who have been working in Data Science for years and have a lot more experience than we do as students, it is not surprising that our score is lower than the winning one. Of course, the model should actually be trained over more epochs than just the two epochs we did, but for time and capacity reasons we had to cut short on that matter.

Additionally, after closing of the competition and evaluation of the submissions, it showed that although the competitors stated the data provided would be sufficient, the limited amount of training data provided was still relevant: The winner of the competition did the extra work to gather more samples (e.g. from similar competitions in previous years) – which might have been his deciduous advantage over other strong competitors. So to improve our results, we could just do the same and gather more training data.

The model we used is still relatively simple in terms of its architecture, so it may not capture all complex patterns and relationships in the recordings (and their spectrograms). Therefore, exploring deeper architectures like ResNet could be worth a try *(Hershey et al., 2017)* to further improve the outcomes. Also, using a model pre-trained on a large-scale audio dataset might improve the model's performance, and reduce training time – which definitely was a pain in our work. Furthermore, we did not use any regularization

techniques. In potential follow-up works, it should be tested if adding regularization techniques helps to prevent overfitting and improve the model's generalization on unseen data and thereby enhance the performance.

Trivially, it should be kept in mind that this classifier only works for the specific bird species which were recorded on the provided sample recordings, so in the current state it can only be used to monitor those species covered. To make the classifier more widely usable, the model needs to be trained on a broader range of species; to re-use the model to classify different birds for monitoring biodiversity in another geographical area, it needs to be trained with recordings of the local species.

All in all, we can see that classifying audio data using image representations of it to utilize image classifying models such as CNNs is a performant approach, and that it is indeed possible to train a model like this one with the limited training data we had and get good enough results – although the amount of data still makes a difference.

To conclude, our trained audio classifier could be utilized in the real-world scenario to identify Eastern African bird species by sound out of passive acoustic monitoring recordings – it does the required job and performs well enough. It shouldn't be, though, as within the competition there were developed significantly better solutions to the problem than ours – so of course, to help advance ongoing efforts to protect avian biodiversity in Africa the best possible way, we recommend using the best available solution instead of this one.

References

Borovec, J. BirdCLEF Convert Spectrograms & Reduce Noise. Retrieved from https://www.kaggle.com/code/jirkaborovec/birdclef-convert-spectrograms-reduce-noise

Chapin III, F. S., Zavaleta, E. S., Eviner, V. T., Naylor, R. L., Vitousek, P. M., Reynolds, H. L., Hooper, D. U., Lavorel, S., Sala, O. E., Hobbie, S. E., Mack, M. C., & Díaz, S. (2000). Consequences of changing biodiversity. Nature, 405(6783), 234–242. https://doi.org/10.1038/35012241

Culliton, P. Inferring Birds with Kaggle Models. Retrieved from https://www.kaggle.com/code/philculliton/inferring-birds-with-kaggle-models

Díaz, S., Fargione, J., Chapin, F. S., & Tilman, D. (2006). Biodiversity loss threatens human well-being. PLoS Biology, 4(8), e277. https://doi.org/10.1371/journal.pbio.0040277

Fujita, A. 4th Place Solution Inference Kernel. Retrieved from https://www.kaggle.com/code/atsunorifujita/4th-place-solution-inference-kernel

Furugori, K. 8th Solution Preprocess Audio & Image. Retrieved from https://www.kaggle.com/code/kunihikofurugori/8th-solution-preprocess-audio-image

Hann, L. BirdCLEF 21 - 2nd Place Model Submit. Retrieved from https://www.kaggle.com/code/leehann/birdclef-21-2nd-place-model-submit

Hershey, S., Chaudhuri, S., Ellis, D. P. W., Gemmeke, J. F., Jansen, A., Moore, R. C., Plakal, M., Platt, D., Saurous, R. A., Seybold, B., Slaney, M., Weiss, R. J., & Wilson, K. (2017, March). CNN architectures for large-scale audio classification. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). http://dx.doi.org/10.1109/icassp.2017.7952132

Kaggle. Bird vocalization classifier. Retrieved from https://www.kaggle.com/models/google/bird-vocalization-classifier/frameworks/TensorFlow2/variations/bird-vocalization-classifier/versions/1

Palanisamy, K., Singhania, D., & Yao, A. (2020, July 22). Rethinking CNN models for audio classification. arXiv.Org. https://arxiv.org/abs/2007.11154

Pushkar, K. BirdCLEF 2023 Inference. Retrieved from https://www.kaggle.com/code/pushkar007/birdclef-2023-inference

Saberlin, M. BC23 3rd Place Solution Refactored. Retrieved from https://www.kaggle.com/code/mariotsaberlin/bc23-3rd-place-solution-refactored

Sura, H. (2020, January 20). Audio Classification using Librosa and Pytorch - Hasith Sura. Medium. https://medium.com/@hasithsura/audio-classification-d37a82d6715

VSydorskyy. (2023). BirdCLEF_2023_1st_place. GitHub Repository. Retrieved from https://github.com/VSydorskyy/BirdCLEF_2023_1st_place

Contributions

Responsibilities	Team Member
Introduction	Aziza & Johanna
Data	Aziza & Johanna
Methods	Aziza & Johanna
Results	Aziza
Discussion and conclusions	Johanna
Python notebook	Aziza