BirdCLEF 2023 – Bird Call Classification

BirdCLEF 2023 – Madárhang klasszifikáció

Mihályi Márk László (EAPH9W), Kozák Balázs (DXAINN), Hegedűs László (G0J9RT)
Team: Convolutional Conviviality | Github

ABSTRACT Our chosen topic was the BirdCLEF 2023 competition from Kaggle, because that comes with a labelled dataset by default and has possibilities to use state-of-the-art techniques. Mel-spectrograms were generated from the bird sounds and then fed to the model, which is based on the Xception [1] network pretrained on the ImageNet dataset. As part of our task, we applied transfer learning by finetuning a smaller model on top of this to achieve our results.

ABSZTRAKT A választott témánk a BirdCLEF 2023-as verseny volt Kaggle-ről, mert egy előre felcímkézett adathalmaz is jár hozzá, valamint remek potenciálja van korszerű technikák próbálgatására. A madárhangokból Mel-spektrogrammokat generáltunk és azt használtuk a modell bemeneteként, ami az Xception [1] hálózat volt az ImageNet adathalmazon előtanítva. A munkánk részeként transzfer learninget alkalmaztunk aminek keretében egy kisebb modellt finetune-oltunk az alap modellre, így érve el az eredményeinket.

I. INTRODUCTION

Sound classification is the first step in understanding more advanced topics involving sound and speech in machine learning.

As human-machine interfaces evolve, one of the final things to be done before invasive technologies (e.g., NeuraLink chip) will be the proper recognition of human speech and the applications built on top of it (e.g., Whisper and home assistants). We felt that such a task would have been a little bit too much of an undertaking, this is why we choose the BirdCLEF 2023 competition. It also comes with an easy to work with labelled dataset.

The Kaggle page of the competition gives another possible, more practical motivation behind the choice (unfortunately as engineers we don't necessarily share these motivations): Previous practices on keeping accurate bird population estimates consisted mostly of observer-based surveys. This is very expensive as it requires people to go out into the forest and count birds all day. Passive acoustic monitoring (PAM) and a machine learning model could yield better results for much cheaper. We tried to train a model for this exact task. The bird counts then can be used by experts to estimate biodiversity and hence even climate change.

II. Field overview and previous solutions

Sound classification at its heart is sequence analysis, which makes RNNs at first glance a natural choice for it. Another way of analyzing sounds is to create a Mel-spectrogram from the waveforms and analyze that as it is supposed to already contain the important features of the sample. A Mel-

spectrogram can be interpreted as an image and hence techniques successful in image classification can eventually be used for our task as well.

A Mel-spectrogram is a special filter that creates a spectrogram from the sound mapped to the Mel-scale, which was designed to have equal sounding pitches at equal distance from each other. [2]

III. System Design

Xception, the model architecture we used, stands for Extreme Inception and was developed by Google in 2017. It uses depthwise separable convolutions [3]. We chose to use transfer learning, as the dataset is massive in size and with limited resources this gives probably the best results. The pretrained model is trained on the ImageNet [4] dataset and is the default from keras.applications.

As part of the finetuning, the base model is freezed, and a new, smaller model is trained on top of it. That includes three layers for us, the first being a 2D pooling layer [5], then a dropout [6] layer for regularization and finally a dense layer with soft-max activation for the output [7].

The full model has about 541 thousand finetuned parameters and almost 21 million frozen parameters, this way only about 1/40 of the parameters were trained by us, making a large save on compute power.

IV. Implementation

A. Data Acquisition and Preparation

We decided to discard the longest samples, notably the ones longer than one minute. The training set was chosen to be 70% of all samples, with the rest being the validation and test datasets.

A custom generator was written to feed the batches to the model. First it cuts the sample into 3 seconds long sections, then it returns the Mel-spectrogram for each of them (the result is extended to three dimensions for Xception).

The Keras dataset API was used to achieve better performance on batch related tasks, mainly caching and prefetching. This way the GPU is always busy and ideally it does not have to wait for the CPU after each epoch.

We planned to use data augmentation techniques to further improve accuracy and prevent overfitting, however, it proved difficult, so we chose to give up on it [8].

B. Evaluation (training, validation, and test errors)

The training was done in Google Colab, on a GPU runtime instance, the 10 epochs took roughly 4 hours. It did not come without challenges, as the free tier virtual machines come with limited amounts of system memory. This meant that at the very minimum we had to use a generator and batch training, since only about 10% of the dataset could be fit in RAM before a CUDA out-of-memory-error happened. Even with this we ran into a bug once, that made the model consume all memory.

We applied early stopping (though retrospectively this was not necessary, it is still good practice) [9] and model checkpointing in order to save the model. The loss function was categorical cross-entropy [10] (with the categorial accuracy metric), and our choice fell on the Adam optimizer [11] to train the model, as it has dynamic learning rate and overall great results in many different domains.

No explicit hyperparameter optimization was done; for validation we relied on the categorical accuracy metric.

C. Testing

The trained model achieved 39.2% accuracy on the test dataset, which is not high, but given our resources we are satisfied. (Other Kaggle participants have scored up to 70% or higher, but that is just a number without much context.)

V. Summary and Future Plans

We implemented a model capable of identifying bird calls in 39% of the cases. For this we used the Mel-spectrograms of the bird sounds and fed them to a finetuned-by-us version of the Xception model (which was pretrained on the ImageNet dataset).

Currently none of us have any future plans to continue studies or projects in this specific area of deep learning, however obviously all three of us have interests in machine learning in general. This even includes a BSc thesis.

VI. References

[1] Chollet, F., 2017. *Xception: Deep learning with depthwise separable convolutions.* In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1251-1258).

[2] Leitner, B. Z. J., & Thornton, S. (2019). *Audio Recognition using Mel Spectrograms and Convolution Neural Networks.*

[3] O'Shea, K. and Nash, R., 2015. *An introduction to convolutional neural networks*. arXiv preprint arXiv:1511.08458.

[4] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. and Berg, A.C., 2015. *Imagenet large scale visual recognition challenge. International journal of computer vision*, 115, pp.211-252.

[5] Gholamalinezhad, H. and Khosravi, H., 2020. *Pooling methods in deep neural networks, a review*. arXiv preprint arXiv:2009.07485.

[6] Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R.R., 2012. *Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580.*

[7] Nwankpa, C., Ijomah, W., Gachagan, A. and Marshall, S., 2018. *Activation functions: Comparison of trends in practice and research for deep learning.* arXiv preprint arXiv:1811.03378.

[8] Perez, L. and Wang, J., 2017. *The effectiveness of data augmentation in image classification using deep learning*. arXiv preprint arXiv:1712.04621.

[9] Shen, R., Gao, L. and Ma, Y.A., 2022. On Optimal Early Stopping: *Over-informative versus Under-informative Parametrization*. arXiv preprint arXiv:2202.09885.

[10] Mao, A., Mohri, M. and Zhong, Y., 2023. *Cross-entropy loss functions: Theoretical analysis and applications*. arXiv preprint arXiv:2304.07288.

[11] Kingma, D.P. and Ba, J., 2014. Adam: *A method for stochastic optimization*. arXiv preprint arXiv:1412.6980.