EFFICIENT CONVOLUTIONAL NEURAL NETWORKS FOR BIOACOUSTICS IN TINY HARDWARE

Technical Report

Christian Walter

Computational Medicine, University of Veterinary Medicine, Vienna, Austria

ABSTRACT

Convolutional neural networks with small computational footprint are deployed on a tiny hardware for a binary classification task of "yellowhammer" bird sounds. We use a simple network structure to minimize the amount of parameters and computations as much as possible. Our submission model for this task achieved a classification accuracy of 91.35% on the validation set. Further, the inference time of the deployed model on the tiny hardware including feature extraction was $20.34ms$.

***Index Terms*—** Bioacoustics, convolutional neural networks, small computational footprint, tiny hardware, microcontroller

1. INTRODUCTION

The 3rd task of the BioDCASE Challenge 2025 requests participants to find small footprint neural networks able to classify the sounds of the bird "yellowhammer" over background noise [1]. Further, the model should be deployable on the "ESP32-S3-Korvo-2 development board" microcontroller system. A baseline [2] was provided to give functionality to a complete machine learning pipeline plus deployment on the target microcontroller system.

The main task was to decrease the size and computational footprint of the neural network while at the same time keeping an acceptable classification performance.

2. METHODS

From the baseline system we only used the code for the deployment on the hardware but the training and testing of the neural network was done in our own framework based on `Pytorch`. Our best performing model was then converted to a `.tflite` model and inserted into the toolchain for the embedding on the microcontroller.

We only used the provided dataset [1] which includes "Yellowhammer" and "Negative" classes. The "Negative" samples correspond to different environment sounds and are higher in number than the "Yellowhammer" samples, which makes the data set slightly unbalanced.

We used the log-mel spectrogram to extract features from the audio files of the dataset. We found that for our system it was advantageous to excessively decrease the feature extraction both on time and frequency axis. For the submission model, we chose a window size of 2048, a hop size of 1024, and 16 mel filter bands.

Our neural network model was a simple convolutional neural network with two convolutional layers followed by two linear layers. The first convolutional layer uses 4 output channels, a kernel size of $(8, 8)$ with stride $(1, 1)$, and a ReLU activation function. The second convolutional layer uses 1 output channel, a kernel size of $(9, 9)$ with stride $(1, 1)$, a dropout mechanism followed by a max pool layer with kernel size of $(1, 8)$ and stride $(1, 8)$, and a ReLu activation function. The two linear layers project the output of the convolutional layers to the two output nodes of the classes. The first linear layer uses a dropout mechanism and ReLU activation, while the second linear layer only uses a softmax output.

A basic training scheme with early stopping was applied and included an "Adam" optimizer starting with a learning rate of 10^{-4} and refinement during training. After deploying the model on the microcontroller system, we extract the inference time from the onboard profiler provided by the baseline system [2].

3. RESULTS

Different models of varying sizes were evaluated but most models reached a classification accuracy of around 90% regardless of a higher or lower computational footprint (within some limits) of a similar network structure. Therefore, we reduced the computational structure of the model and the feature extraction to a reasonable small scale and achieved an average precision of 91.35% with our submission model on the validation set (confusion matrix shown in Figure 1).

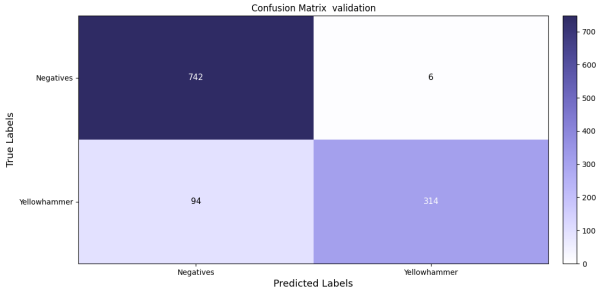

Figure 1: Confusion Matrix of the validation set classified by the submission model.

A sample of the learned weights of the first convolutional layer is shown in Figure 2, illustrating how only very few filters are required in this classification task. The submission model and feature extraction inference time resulted in $16.45ms$ and $1.63ms$, respectively. By adding the allocation time, the total inference time of the microcontroller system resulted in $20.34ms$.

We only submitted one model to the challenge with results shown in Table 1. Also we did not submit a `keras` (.5h) model but a `.onnx` model because of our `PyTorch` framework and hope

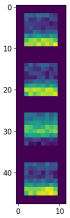

Figure 2: A sample of learned weight of the first convolutional layer of the submission model.

future versions of this challenge will include the embedding of more model formats.

Table 1: Results on the validation set and model profile.

Summission Name:	cnn_micro_v1_logmel16_n2048
Average Precision:	91.35%
Model Size [kB]:	7.02
Inference Time [ms]:	20.341

4. CONCLUSION

In our experiments, it was remarkable to observe, how much a convolutional neural network can be reduced in size and computations while at the same time similar accuracies are achieved. The model struggled mainly in the classification of the "Yellowhammer" class and learned the "Negative" class with a stronger bias, which could originate from the unbalanced dataset. Therefore, a contrastive learning approach would have been more fitting and should be evaluated in future research. The baseline framework provided a great start into model embedding on the microcontroller system once it was understood. However, much more time need to be spent to carefully evaluate the model deployment, since only the performance profiler did actually run on it. All in all, our submission model performed well while at the same time the inference time on the hardware could be decreased compared to the baseline model.

5. REFERENCES

[1] I. Morandi, P. Linhart, M. Kwak, and T. Petrusková, "Biodcase 2025 task 3: Bioacoustics for tiny hardware development set," 2025. [Online]. Available: https://doi.org/10.5281/zenodo.15228365

[2] G. Carmantini, F. Förstner, C. Isik, and S. Kahl, "Biodcase-tiny 2025: A framework for bird species recognition on resource-constrained hardware," https://github.com/birdnet-team/BioDCASE-Tiny-2025, 2025.