Edge Deep Learning for Bird Species Recognition

Marlon Müller
marlon.mueller@tum.de

Shao Jie Hu Chen
shaojie.hu@tum.de

Ahmed Kaddah
ahmed.kaddah@tum.de

ABSTRACT

Our project proposes a solution for monitoring bird populations in remote ecosystems, emphasizing energy efficiency and minimal human intervention. The hardware configuration includes ESP32-equipped edge nodes with microphones and GPS modules, powered by batteries. Leveraging the low-energy microcontrollers, a Convolutional Neural Network (CNN) is employed to classify bird species using sound. The classification results are transmitted to a central Raspberry Pi via LoRa. Training will be done utilizing the Xeno-canto dataset merged with broader environmental sound datasets.

1 INTRODUCTION

Birds play an important role in the ecosystem; they serve as an indicator for environmental health and ecosystem vitality. If bird populations and diversity decline, important ecosystem processes, particularly decomposition, pollination, and seed dispersal, will likely decline as a result as investigated by Şekercioğlu et al. [12]. Studying bird populations and behaviors in natural habitats, such as forests and mountains, provides valuable insights into the state of these ecosystems [16].

This project would enable biodiversity monitoring to assess habitat health, identify endangered species, and implement targeted conservation efforts [10]. Additionally, the project promotes non-intrusive research, respecting natural wildlife behavior while gathering accurate data. Its use of cost-effective low-energy microcontrollers expands access to bird research, encouraging scalability and comprehensive studies across diverse environments.

Employing low-energy micro-controllers is essential for ensuring efficient bird monitoring with minimal human intervention, aligning with the low-to-none energy constraints of these remote ecosystems. We utilize such micro-controllers to classify bird species via sounds.

For an accurate classification, deep learning approaches for environmental sound recognition gained popularity in the research community for their capacity to extract features from raw data and more precise results [9][7]. We therefore propose to apply a CNN, for the bird species recognition, thereby contributing to non-intrusive research that respects natural wildlife behavior while collecting precise data.

The resulting classifications are then transmitted to a central node via LoRa. By classifying the sound directly on the edge node, only the event notification has to be transmitted with an extremely low bandwidth occupation. The central node aggregates the data, making it readily available for researchers and other parties.

2 LITERATURE

In our initial literature research, we came across several papers discussing state-of-the-art approaches for similar use cases. Tsompos et al. [15] present a CNN architecture designed for IoT audio bird detection with a parameter count of approximately 73,000. However, the study lacks insights into the practical implementation on an edge device, particularly concerning aspects such as memory utilization. Disabato et al. [5] explore the application feasibility of a CNN architecture on the STM32 Nucleo H7 board, again for bird detection. Their investigation encompasses considerations of memory constraints, computational speed, and power consumption. The complete implementation of their model exhibits a memory footprint of approximately 750 kB.

Some applications tailored for edge deployment with more comparable hardware requisites but divergent application include, e.g., Höchst et al. [6]. The authors propose a system architecture that consists of different ESP32s, which are connected to an NVIDIA Jetson Nano. The ESP32s transmit captured data to the central node, tasked with processing the raw data. Such setups pose a drawback due to the central node being a computing bottleneck. Moreover, this setup requires high-data-rate transmissions and thus the use of less energy-efficient protocols like Wi-Fi.

The authors of [4] employ a lightweight CNN to detect illegal tree-cutting activities. The proposed system involves multiple nodes connected to LoRa gateways, and inference is performed on an ARM-Cortex M4F-powered device with 256 kB of RAM. Another similar approach is presented in Singh et al. [13], where the proposed architecture, for an use case of forest's monitoring, consists of edge devices interconnected using LoRa.

3 DATASETS

In our initial literature review, we have identified various datasets suitable for training deep learning models for bird species classification. We aim to avoid custom dataset creation, which can be time-consuming and limit broader applicability. One frequently utilized dataset for similar applications is the Xeno-canto dataset [1]. It focuses on bird sounds and offers the flexibility to query based on, e.g., location or bird species, rendering it an ideal choice as the primary dataset. However, the training data must also encompass non-bird sounds like environmental noises in order to be meaningful.

We came across serveral datasets that focus solely on bird sound occurrences without distinguishing between bird species, as in the case of freefield1010 (Stowell and Plumbley [14]). Other datasets, like BirdVox-DCASE-20k (Lostanlen et al. [8]), are specific to individual bird species or locations. This makes such datasets less suitable as primary datasets but still valuable as supplementary data. Therefore, our current plan involves merging the Xeno-canto data with one or a selection of broader environmental sound datasets to create a suitable training dataset.

4 ARCHITECTURE

Taking into consideration the requirements and constraints we have in our use case, as well as the advantages and disadvantages of the different architectures found in state-of-the-art approaches, we propose the architecture as shown in figure 1. Choices regarding

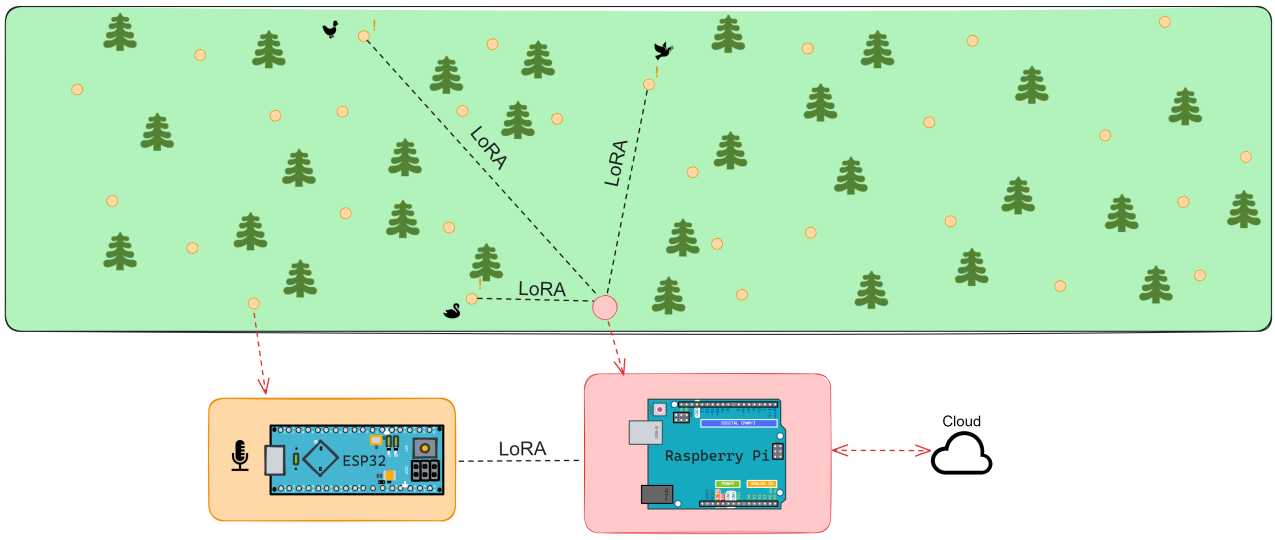

Figure 1: An overview of the project's structure, featuring multiple ESP32 devices deployed in remote locations. These ESP32 devices communicate with a central node, a Raspberry Pi, via the LoRa protocol. The Raspberry Pi, in turn, forwards the collected data to the cloud.

the hardware, software and communication protocols used in our architecture shall be explained in detail in the following subsections.

4.1 Hardware

The architecture consists of ESP32s in edge nodes, each one of them equipped with a microphone used to record environmental sounds, as well as a GPS module to obtain geolocation information. The ESP32 are connected to a Raspberry Pi (central node) using the LoRa protocol.

The edge nodes are powered using batteries, whereas the central node may be strategically positioned for access to the power grid. As an additional feature to our system, the ESP32 could also be powered using solar panels (in order to improve the lifespan).

Compared to the state-of-the-art, we have used ESP32 as in Höchst et al. [6], as this is one of the lowest power consumption microcontrollers that we have found with enough memory to assumingly fit a neural network inside it and with a dedicated library to introduce a pretrained neural network within it.

4.2 Communication

An important choice is the utilized communication protocol. In this scenario, various microcontrollers are strategically positioned in nature, tasked with transmitting information to a central node.

Due to network inference on the edge device, the edge nodes are only required to transmit relatively small amount of data. This bandwidth efficiency enables the adoption of a LPWAN communication protocol such as LoRa. As motivated in Singh et al. [13] and in Andreadis et al. [4], LoRa seems a suitable protocol for this project, due to its long-range and low-power consumption. We thus expect that the energy consumption of our architecture is lower than, e.g., in Höchst et al. [6], where Wi-Fi is used.

4.3 Software

The objective is to employ the datasets outlined in section 3 to train a deep learning model using high-level libraries such as PyTorch [11]. This process may include tasks like clipping crowd-sourced audio files, potentially resampling them to a target sampling rate. Additionally, sound preprocessing is also done on the ESP32.

Our strategy for feature extraction involves the implementation of mel frequency cepstral coefficients (MFCC), a widely utilized method for similar tasks [9]. Incorporating an additional discrete cosine transform, which is comparatively resource inexpensive, serves to further diminish the model size and reduce inference time compared to processing a mel-scale spectrogram [4], the second common approach identified.

Following model training, the model will be converted into ONNX format [3]. Given the power constraints, particularly for the deep learning model, we anticipate the necessity for low-level optimizations to fit the network onto the microcontroller while minimizing performance loss. Addressing this challenge, we plan to employ the ESP-DL library [2]. This library transforms the ONNX model into ESP32-friendly C++ code. Notably, ESP-DL only supports very established neural network layers, e.g., 2D convolution and max pooling. However, this aligns with the prevalent choices in the literature surveyed in Section 2. Initial experiments are promising, but further testing is needed for full validation. Key parameters like clip length, sampling rate, STFT window size, the number of filter banks, and the specific network architecture must be carefully selected and analyzed to align with available hardware resources. Should deploying on an ESP involve excessive deep learning optimization or oversimplification to the extent that deep learning offers no advantage, we will opt for more traditional audio processing methods.

Once a bird species event is inferred, it will be transmitted to the LoRa gateway along with a timestamp. Depending on the final configuration, additional data such as a sensor ID (assuming multiple gateways in a general network) or location may also be transmitted, potentially less frequently. Regardless, the transmitted data should be kept minimal. From the gateway onward, the available software options are diverse. An initial objective is to visualize the detected bird species on a map using a simple frontend.

5 DEMONSTRATION

In order to implement the architectural framework outlined in section 4, the equipment listed in table 1 is required to build a simplified demonstration of the project.

Units	Device
×3	ESP32-S3FN8 (†)
×3	Batteries for ESP32
×3	Microphones for ESP32
×3	GPS modules for ESP32
×3	LoRa Transceiver modules for ESP32
×3	Waterproof and impact-resistant boxes for ESP32
×1	Raspberry Pi (any version)
×1	Power Supply for Raspberry Pi
×1	LoRa module for Raspberry Pi
×1	Waterproof and impact-resistant box for Raspberry Pi

Table 1: Equipment list for the demo.

For (†), other alternatives may be used (any ESP32, ESP32-S2, ESP32-S3, or ESP32-C3 with max flash memory). The selection is based on ESP-DL compatibility. ESPs with built-in LoRa support (or similar specs) are preferred.

For the demonstration, the proposed architecture will only consists of three ESP32s. To summarize, a Raspberry PI acts as a central node and LoRa gateway. The edge nodes are three ESP32s, each equipped with a microphone, a GPS module, and a LoRa transceiver. The sensor nodes run on battery and are protected from the environment using waterproof and impact-resistant boxes.

In addition, as we are interested in a quantitative analysis of our approach, different measurement equipment will be needed. In particular, we are interested in, e.g., the energy consumption of our demo setup and the bandwidth and latency in the network. For this, certain measurement tools like multimeters and oscilloscopes and specialized software like Wireshark will be utilized. Additional measurements may become necessary throughout the development. These will be further addressed once we have successfully established a functional demo of the project.

REFERENCES

[1] 2005. *xeno-canto - Sharing wildlife sounds from around the world.* Retrieved November 9, 2023 from https://xeno-canto.org

[2] 2019. *Espressif deep-learning library for AIoT applications.* Retrieved November 9, 2021 from https://github.com/espressif/esp-dl

[3] 2019. *Open Neural Network Exchange (ONNX).* Retrieved November 9, 2023 from https://onnx.ai

[4] Alessandro Andreadis, Giovanni Giambene, and Riccardo Zambon. 2021. Monitoring illegal tree cutting through ultra-low-power smart iot devices. *Sensors* 21, 22 (2021), 7593.

[5] Simone Disabato, Giuseppe Canonaco, Paul G Flikkema, Manuel Roveri, and Cesare Alippi. 2021. Birdsong detection at the edge with deep learning. In *2021 IEEE International Conference on Smart Computing (SMARTCOMP)*. IEEE, 9–16.

[6] Jonas Höchst, Hicham Bellafkir, Patrick Lampe, Markus Vogelbacher, Markus Mühling, Daniel Schneider, Kim Lindner, Sascha Rösner, Dana G Schabo, Nina Farwig, et al. 2022. Bird@ Edge: Bird Species Recognition at the Edge. In *International Conference on Networked Systems*. Springer, 69–86.

[7] Stefan Kahl, Connor M Wood, Maximilian Eibl, and Holger Klinck. 2021. BirdNET: A deep learning solution for avian diversity monitoring. *Ecological Informatics* 61 (2021), 101236.

[8] Vincent Lostanlen, Justin Salamon, Andrew Farnsworth, Steve Kelling, and Juan Pablo Bello. 2018. BirdVox-full-night: a dataset and benchmark for avian flight call detection. In *Proc. IEEE ICASSP*.

[9] Dulani Meedeniya, Isuru Ariyarathne, Meelan Bandara, Roshinie Jayasundara, and Charith Perera. 2023. A Survey on Deep Learning Based Forest Environment Sound Classification at the Edge. *Comput. Surveys* 56, 3 (2023), 1–36.

[10] Sefi Mekonen. 2017. Birds as biodiversity and environmental indicator. *Indicator* 7, 21 (2017).

[11] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).

[12] Çağan H Şekercioğlu, Gretchen C Daily, and Paul R Ehrlich. 2004. Ecosystem consequences of bird declines. *Proceedings of the National Academy of Sciences* 101, 52 (2004), 18042–18047.

[13] Rajesh Singh, Anita Gehlot, Shaik Vaseem Akram, Amit Kumar Thakur, Dharam Buddhi, and Prabin Kumar Das. 2022. Forest 4.0: Digitalization of forest using the Internet of Things (IoT). *Journal of King Saud University-Computer and Information Sciences* 34, 8 (2022), 5587–5601.

[14] Dan Stowell and Mark D Plumbley. 2013. An open dataset for research on audio field recording archives: freefield1010. *arXiv preprint arXiv:1309.5275* (2013).

[15] Christos Tsompos, Vasilis F Pavlidis, and Kostas Siozios. 2022. Designing a Lightweight Convolutional Neural Network for Bird Audio Detection. In *2022 Panhellenic Conference on Electronics & Telecommunications (PACET)*. IEEE, 1–5.

[16] Mohamed Zakaria, Puan Chong Leong, and Muhammad Ezhar Yusuf. 2005. Comparison of species composition in three forest types: Towards using bird as indicator of forest ecosystem health. *Journal of biological sciences* 5, 6 (2005), 734–737.