Mosquito Classification based on Aquatic signals on Microcontrollers, TinyML

Student: **Muhannad Alhumaidi**

Supervisor: **Dr. Rytis Paškauskas**

Abstract

Malaria and dengue pose urgent health challenges. This project explores real-time mosquito species classification using low-power microcontrollers. Building upon the HumBugDB dataset with custom preprocessing and segmentation, we developed an efficient C++ feature-extraction pipeline to run in ~263 ms per 1-second audio slice, deploying a lightweight CNN on the Arduino Nano 33 BLE Sense within ~98 KB SRAM. All code, datasets, and analyses are available on my GitHub [11].

1. Introduction

In 2023, Malaria, a mosquito-borne disease, took 567,000 lives in Africa [1], 76% were children under 5—that's 1,200 young lives lost every day. Dengue fever is also rising, with thousands of cases in Saudi Arabia [2]. Since 2000, 12.7 million deaths have been averted through global efforts [1]. This project aims to push it further: developing low-power, real-time detection of high-risk mosquito species on microcontrollers. Because saving lives shouldn't require a data center.

2. Related Work & Datasets

Several datasets were explored, including the HumBug DB, Zooniverse [3] which is a crowd lead paper – a very interesting read if I may add - however its problems included not clear annotations, while mosquito events where identified the species themselves were not clear.

mobile phone recordings [4] the mobile microphone will make it harder to deploy on microcontrollers … a small, annotated dataset [5], this one had segments of 300ms, which is sub-optimal to say the least, also it was only 15 minutes, which isn't even worth the hassle of merging to any actual large dataset, with 3 species.

Ultimately, the HumBugDB dataset [6] was chosen, which is known to be the state of the art on mosquitoes, offering ~13 hours of data for the mosquito classification task. We looked into many other datasets however none were usable according to our metrics.

3. Methodology

3.1 Literature Review and Model Selection

Foundational understanding of attention mechanisms was developed from *Attention Is All You Need* [9], followed by domain-specific TinyML research such as TinyChirp [7] and *Tiny Transformers for Environmental Sound Classification on the Edge* [8].

Raw-audio transformer and CNN models offer architectural simplicity but lower accuracy, while the mel-spectogram cnn always beats other architectures but with the added mel-spectogram delay which can often double the inference time, which is a problem that [7] tried to resolve…

Initial benchmarks on HumBugDB using my implementation of the TinyChirp paper's models:

- Mel-spectrogram + CNN: 61% accuracy.
- Raw-audio transformer: 57%.
- Raw-audio CNN: 55%.

Transformer deployment challenges on TensorFlow Lite Micro included lack of layer normalization and head-multiplication support — [8] couldn't even deploy it on microcontrollers, I managed to get around this by using RIOT-OS with RIOT-ML to deploy a simulated-input transformer model on Xiao ESP32-C3.

3.2 Data Refinement

A major bottleneck was false negatives in mosquito classification. Which made me manually go through the entire DB and timestamp each second where mosquitoes aren't in the recordings, I then logged the timestamps into a csv file with a very clever technique which made me listen to only 40% of the audio and classifying 100% of it. yielding a cleaner dataset and raising CNN accuracy to 74%. This refined dataset is now part of a co-authored research paper I'm conducting with my supervisor.

3.3 Signal Processing Enhancements

However when listening to the actual recordings before passing them to our model, a very strange glitching sound can be heard, after a deeper look into signal processing I concluded that downsampling aliasing errors happened, they were eliminated by testing multiple preprocessing pipelines and finalizing a **low-pass filter + downsampling** approach, the code of which is on github.

3.4.1 Feature Extraction

After deploying the model, it didn't perform nearly as well as we hoped it would, after very careful analysis, I found that the mel-spectogram of the C++ pipeline didn't match the python pipeline, which is understandable since no library mentiones matching outputs.

C++ and Python feature-extraction pipelines were implemented from scratch, including FFT, STFT, and mel-spectrogram algorithms to ensure cross-language output consistency — a limitation absent in existing libraries.

However, on the Arduino nano BLE 33 sense, it was painfully slow, 1s of audio's worth of features were extracted in more than 1 second, which obviously kills the real-time requirement of the project.

3.4.2 Embedded Optimization

We made the feature extraction optimized on any board with an ARM-Cortex processor, reduced mel-spectrogram computation time to **263 ms** on the Arduino Nano 33 BLE Sense — faster than Edge Impulse's fixed-point DSP (271 ms). This performance was achieved through:

- Matrix multiplication optimizations for triangular mel filterbanks.
- Leveraging ARM's highly optimized FFT routines.

4. Results & Discussion

Initial results showed 61% accuracy with raw audio transformers, 65% with CNN/Mel pipelines, and 74% after dataset cleaning. The final optimized separable CNN achieved 78% balanced accuracy, F1 score of 77.8%, and recall of 77%.

5. Deployment

On-device feature extraction was optimized to 263 ms per 1s audio slice on the Arduino Nano 33 BLE Sense, outperforming Edge Impulse's implementation (271 ms). The system ran within ~98 KB SRAM. Tests on ESP32-C3 and Portenta H7 confirmed cross-board compatibility.

6. Conclusion & Future Work

This project demonstrated a full pipeline for mosquito species detection on microcontrollers, achieving real-time classification with optimized DSP and lightweight CNN models. Future work will explore transformer-based models on embedded systems, investigate the bottleneck of Mel-spectrogram features, and extend the dataset. Integration with IoT protocols like LoRa/MQTT will enable real-world applications.

References

[1] World Health Organization. Malaria Report, 2023.

[2] Ministry of Health, Saudi Arabia. Dengue Updates, 2023.

[3] Kiskin et al. HUMBUG Zooniverse: A crowd-sourced acoustic mosquito dataset, 2020.

[4] Batista et al. Using Mobile Phones as Acoustic Sensors, 2011.

[5] Mukundarajan et al. Annotated dataset of mosquito bioacoustics, 2017.

[6] Kiskin et al. HumBugDB: Acoustic mosquito dataset, 2021.

[7] Han et al. TinyChirp: Efficient audio models for embedded devices, 2022.

[8] Zhang et al. Tiny Transformers for Environmental Sound Classification on the Edge, 2022.

[9] Vaswani et al. Attention is All You Need, 2017.

[10] InsectSet459: Open dataset of insect sounds for bioacoustic ML, 2023.

[11] Project GitHub Repository: https://github.com/Jack-froOost/Mosquito-species-classification-based-on-acoustic-signals-on-microcontrollers