E1 201: Hardware Acceleration and Optimization for Machine Learning

Project: Bird and Animal Call Detection on Edge Devices

Advisor :

Chetan Singh Thakur
Associate Professor, Department of Electronic Systems Engineering, Indian Institute of Science, Bangalore-560012

Team :

Kirteyman Singh Rajput*
Paul Antony*
Shubham Datta*

*M.Tech., Department of Electronic Systems Engineering, Indian Institute of Science (IISc), Bangalore

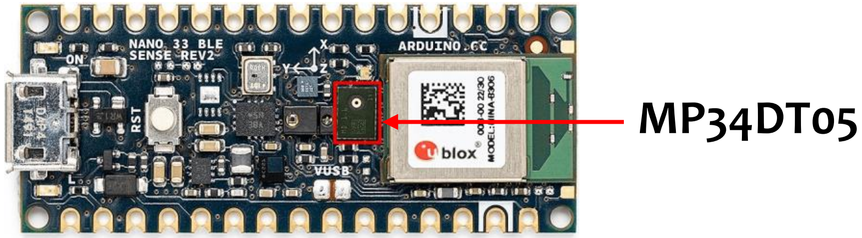
Introduction

Title: Bird and Animal Call Detection

Objective: Detect and classify different bird and animal calls using audio data and machine learning (CNN).

Classes:
The dataset consists of audio files in .wav format.
Eight classes:
- Bird1 (osprey) : 100 wav files (80% training and 20% testing)
- Bird2 (norcar): 100 wav files (80% training and 20% testing)
- Bird3 (ruckin): 100 wav files (80% training and 20% testing)
- Bird4 (pasfly): 100 wav files (80% training and 20% testing)
- Bird5 (rewbla): 100 wav files (80% training and 20% testing)
- Bird6 (rebwoo): 100 wav files (80% training and 20% testing)
- Cat: 164 wav files (80% training and 20% testing)
- Dog: 113 wav files (80% training and 20% testing)

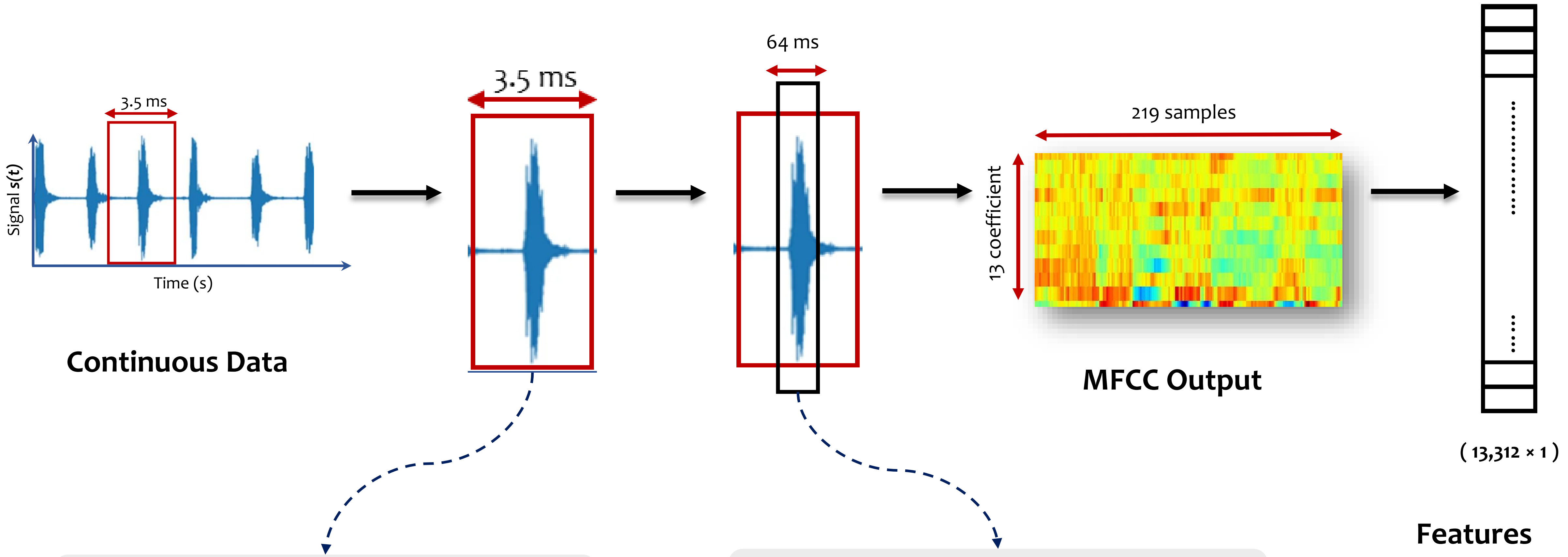
Hardware :
Microcontroller Board: **Arduino Nano 33 BLE Sense**
ML can be used in Arduino boards using **TensorFlow Lite** libraries
Sensor: Digital Microphone (MP34DT05)



Datasets:

1) https://www.kaggle.com/datasets/mmoreaux/audio-cats-and-dogs
2) https://www.kaggle.com/datasets/luisblanche/birdcall-singing-3?select=norcar

Data and Preprocessing

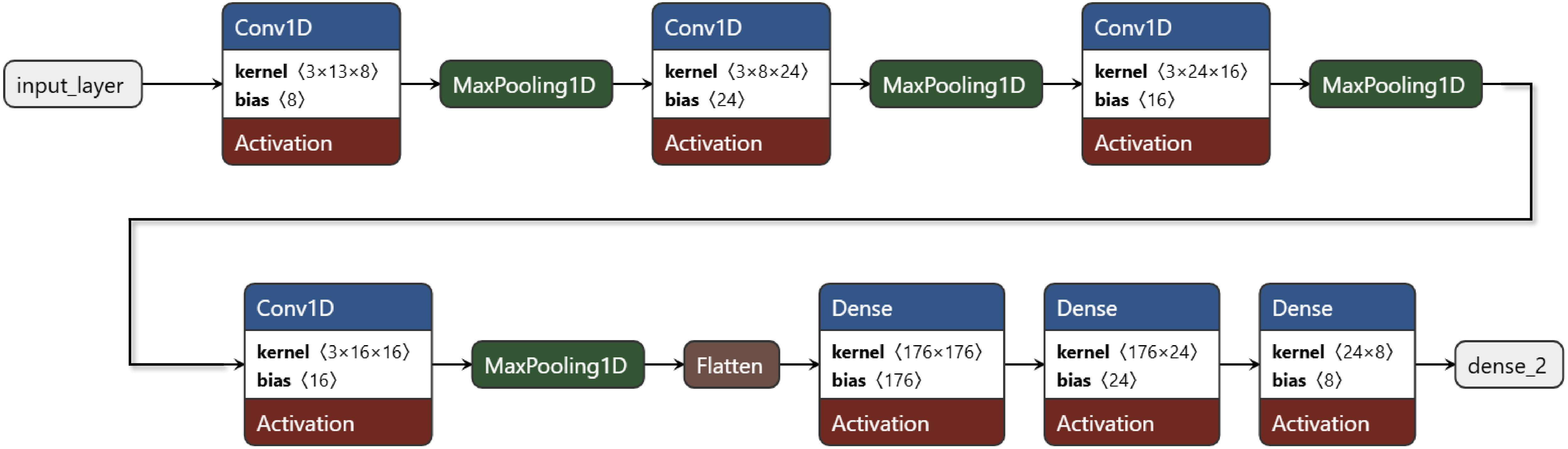


Window size = 3.5 sec
Number of samples = 3.5 × f_s
= 3.5 × 16,000
= 56,000 samples

Frame size = 64 ms => 1024 samples
Frame stride = 16 ms => 256 samples
Mel- Coefficient = 13

Model Architecture

INPUT SIZE: 219 × 13 × 1
Batch Size: 16



OUTPUT SIZE: 8 Classes
(Bird1, Bird2, Bird3, Bird4, Bird5, Bird6, Cat, Dog)

*Code: https://github.com/kirteymanr/Bird-and-Animal-Call-Detection-on-Edge-Devices

Hardware Implementation

Challenge

	Microprocessor	>	Microcontroller
Compute	1GHz – 4GHz	~10x	1MHz – 400MHz
Memory	512MB – 64GB	~10000x	2KB – 512KB
Storage	64GB – 4TB	~100000x	32KB - 2MB
Power	30W – 300W	~1000x	150uW – 23.5mW

TensorFlow Lite was essential for our project as it enables running deep learning models on resource-limited devices **by reducing the library size** from ~400 MB (standard TensorFlow) to ~1 MB, allowing models to operate within strict memory and storage limits.

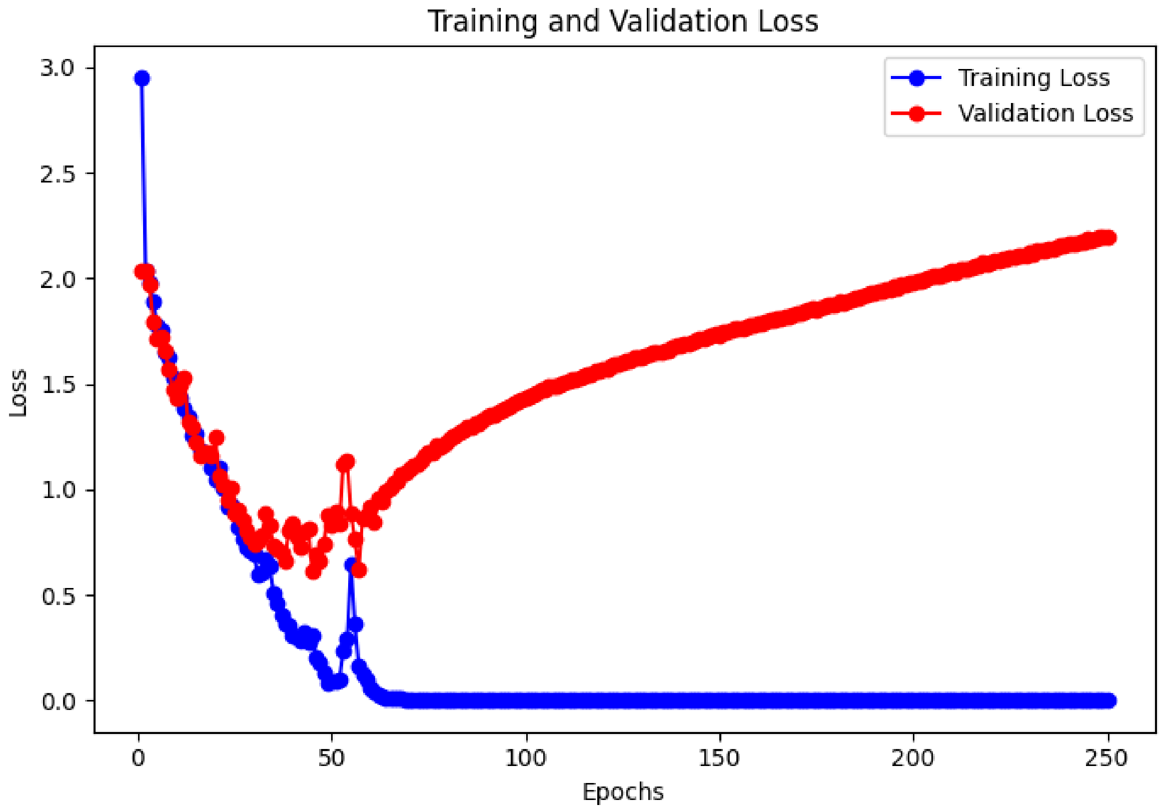
This lightweight framework made it feasible to **deploy models on mobile devices and embedded systems** without exceeding available resources, unlike the full TensorFlow library, which can demand over 1 GB of RAM to run even a 200 MB model.

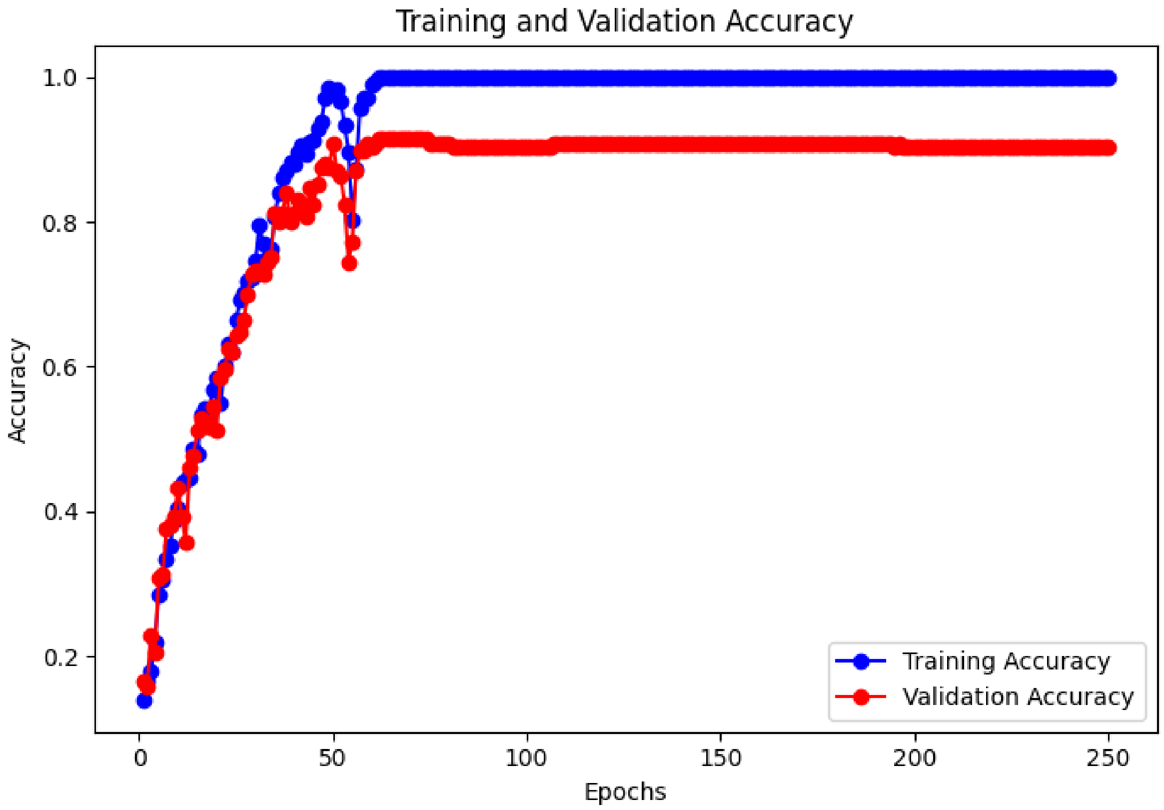
Deployment

Collect Data → Preprocess → Develop Model → Training → Evaluation and Optimize → Convert Model → Deploy Model → Make Inferences

TensorFlow | **TensorFlow Lite**

Results





Final Training Loss: 8.21863523015054e-06
Final Validation Loss: 2.19516658782959

Final Training Accuracy: 1.0
Final Validation Accuracy: 0.9034090638160706

Conclusion and Future Work

Conclusion:

This project successfully developed a model for detecting and classifying bird and animal calls using a 1D CNN on MFCC features. Integrating hardware (Arduino Nano 33 Ble) and software achieved reliable real-time classification, showcasing the potential for practical applications in wildlife monitoring and ecological studies.

Future Work:

Future efforts will focus on enhancing the model's accuracy by:

- **Expanding the Dataset:** Incorporating additional species to improve classification diversity and model robustness.
- **Optimizing Hardware:** Exploring energy-efficient hardware solutions to enable longer deployment in field conditions.
- **Model Enhancements:** Investigating advanced techniques like transfer learning or hybrid models to improve classification accuracy and computational efficiency.

Thank you