

Problem Selection and Motivation

Advances in machine learning have made it possible to implement TensorFlow models on microcontrollers, paving the way for real-time object detection and decision-making in compact, resource-constrained environments. The goal of this project is to develop a modular architecture to scale a large TensorFlow model down to TensorFlow Lite Micro for deployment on a SparkFun Thing Plus RP2040 microcontroller with an attached ARDUCAM.

Dataset Selection

The dataset chosen for this project is the Car Object Detection dataset from Kaggle: [Car Object Detection Dataset](#). This dataset contains labeled images of cars, making it suitable for training an object detection model. Given its size and diversity, it should be sufficient for training a deep network capable of recognizing cars in various conditions.

Network Selection and Customization

The network will start as a standard convolutional neural network (CNN) but will eventually be quantized and possibly pruned to be deployed as a TensorFlow Lite Micro model for the SparkFun Thing Plus RP2040.

Framework Selection

TensorFlow will be used for initial model training and optimization due to its robust ecosystem and support for TensorFlow Lite conversion. TensorFlow Lite will be used for model inference on the RP2040 microcontroller.

Reference Materials

- [Pico TensorFlow Lite Micro GitHub](#)
- [TensorFlow Lite for Microcontrollers](#)
- [TinyML Audio for Everyone](#)
- [SparkFun ThingPlus RP2040](#)
- [ARDUCAM](#)

Performance Metrics

- **Accuracy**
- **Latency**
- **Memory Footprint**
- **Power Consumption**

Project Schedule

- **March 27 - April 3: Proposal Submission & Initial Setup**
 - Finalize project proposal and submit via GitHub.
 - Set up project repository and structure (Proposal, Code, Reports, etc.).
 - Verify dataset availability and pre-process data if needed.
- **April 4 - April 14: Model Selection & Initial Training**
 - Select a deep learning model suitable for object detection.
 - Implement the model using TensorFlow and train it on the chosen dataset.
 - Conduct preliminary training runs and analyze initial performance.
- **April 15 - April 22: Optimization & Model Compression**
 - Apply techniques such as quantization and pruning to convert the model to TensorFlow Lite.
 - Test the compressed model on a local device for validation.
 - Adjust hyperparameters to balance performance and efficiency.
- **April 23 - April 28: Deployment & Testing on RP2040**
 - Deploy the TensorFlow Lite model to the SparkFun Thing Plus RP2040 microcontroller.
 - Run real-world tests to evaluate inference speed, accuracy, and resource usage.
 - Debug issues and refine deployment strategy.
- **April 29 - May 1: Final Presentation Preparation & Delivery**
 - Prepare a 15-20 minute presentation, including results, challenges, and future work.
 - Present the project on the assigned date.
- **May 2 - May 6: Final Report Submission**
 - Write and finalize the individual report.
 - Submit all deliverables (reports, code, and documentation) via GitHub by May 6.