Lab 1: Keyword Spotting

## THE CONVULUTED

Course Number: ECE-597

**November 1st, 2024**

# Objective

# The objective of this lab assignment is to deploy a keyword spotting (KWS) model on the Arduino Nano BLE 33 using Edge Impulse, building skills in model training, deployment, and optimization for tinyML applications. This involves setting up Edge Impulse for custom data collection, configuring the model to recognize keywords like “yes” and “no,” and experimenting with impulse creation, MFCC feature extraction, and neural network training. The lab emphasizes evaluating model performance through featured platform metrics like confusion matrices, accuracy scores, and feature distribution, aiming to improve keyword recognition accuracy. Lastly, the lab requires a report documenting the model’s memory usage, inference time, for deployment insights on constrained hardware which the platform mimics our Arduino Nano BLE 33.

# Method/Model/Architecture/Dataset

|  |
| --- |
|  |
|  |
|  |
|  |

# Results

|  |
| --- |
|  |
|  |
|  |
|  |

# Takeaway

This lab taught how to use Edge Impulse as well as how to train and test models on the Arduino microcontroller. Compared to running experiments through Python notebooks and code, Edge Impulse provides a much simpler interface on which to train, test, and experiment. It was very easy to collect data from a device, process it, and build and test a model using the provided toolset. While the constructed model did not perform well with personally-collected data, the system made collecting it a rather seamless process. The hardest part of this lab was connecting the individual device to the cloud console of the Edge Impulse Lab. It was difficult to connect the TinyML Arduino kit to the lab through the daemon, but once connected, limited technical difficulties remained.

Overall, the model performed quite well, achieving 97% on weighted average precision, recall, and F1-score. However, the results for the particular keyword, “lemon”, and the “silence” category were significantly less remarkable. “Lemon” was only successful 14.3% of the time, otherwise classified as “unknown. The cases of “silence” were never correctly classified. These results are all detailed on the confusion matrix. These poor results could easily be a result of class imbalance, small sample sizes, and low sample diversity through the methods that the samples were collected. Had more data been collected or had it been collected in a better, more systematic way, the model might have yielded better results.