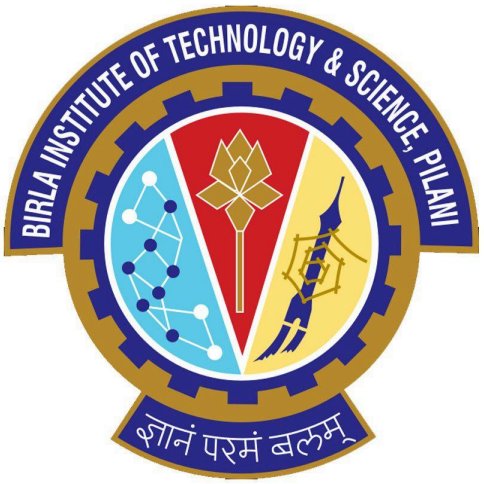
Project Proposal

EEE G513
Machine Learning for EE



By
Lovenya Jain - 2021B4AA1732P
Param Gupta - 2021A3PS0236P

1. Problem Statement

With the boom in IoT and integrated networks in today's world, we see specialised devices with minimal compute and memory pervade almost all aspects of everyday life. As the capabilities of artificial intelligence increase, it becomes increasingly important to develop techniques that enable the implementation of such useful techniques on edge devices. This project addresses the challenge of **low-latency keyword spotting** in live speech on edge devices with constrained memory and computational resources. The goal is to develop a lightweight model capable of performing efficient, real-time keyword recognition directly on devices like Arduino Nano BLE Sense, minimizing the need for cloud-based processing.

2. Techniques Employed

The project will use **supervised learning** as the primary ML technique. Specifically, we will explore architectures like **TinyML** - inspired neural networks designed for **real-time keyword spotting.** There will be a large focus on implementing efficient **digital signal processing ideas** to preprocess audio signals (discrete fourier transforms**/fast fourier transforms**), reduce the effect of auditory noise, manage the variability produced due to accent of different speakers, and so on. Research will focus on optimizing inference speed, memory usage, and accuracy to achieve low latency in live environments. The project's future plans can be extended further to add more keywords, reduce real time inference speed even further.

3. Dataset Availability

We will use datasets that contain keywords instead of full speech data. Some examples include **Google's Speech Commands** dataset, etc. Additionally, we may generate a small custom dataset of keywords. The dataset will require processing to ensure compatibility with edge-device limitations.

4. Prior Works and Basis for Improvement

Few research papers have explored keyword spotting with optimized models for low-resource devices, using techniques such as **quantized neural networks**, **pruned layers**, and **memory-efficient architectures**. Our approach will leverage these ideas, with a focus on pushing the boundaries of low-latency inference, targeting latency of less than 100 milliseconds. Improvements will focus on:

- Reducing inference time by optimizing model architecture.
- Minimizing memory footprint for compatibility with limited hardware.
- Maintaining high accuracy despite these optimizations.

5. Hypothesis and Techniques for Improvement

The hypothesis is that by incorporating specific **model compression** techniques (e.g., quantization and pruning) and efficient **lightweight architectures** (e.g., Depthwise Separable CNNs), we can achieve near-instantaneous keyword detection with a small memory footprint, maintaining accuracy above **80%** (90% being an ambitious but not that far-fetched, subjected to experimentation) for common keywords.

6. Performance Metrics

The key performance metrics will be:

- **Latency**: Time from audio input to output (goal: <100ms).
- **Accuracy**: Detection accuracy of keywords (goal: >80%)
- **Model Size**: Memory footprint (goal: <1MB).
- **Power Consumption**: Energy efficiency on edge devices.

7. Hardware Testing Plan

Yes, if feasible, the ASR model will be deployed and tested directly on the Arduino Nano BLE Sense (as it has a built-in microphone) to measure real-world performance metrics. We will test:

- **Latency** in responding to commands.
- **Accuracy** across different environments (quiet vs. noisy).
- **Consistency** in real-time operation under power constraints.