**GROUP MEMBERS:**

**Sharis Stanley Rebeiro - 100862101**

**Shreyas Shrihari Manjula - 100813324**

**Pooja Atulkumar Modi - 100816838**

**Ankitha S - 100861588**  
  
  
ADVANCED SPEECH COMMAND RECOGNITION SYSTEM

# Abstract

This project focuses on the development of a high-performance, noise-robust speech command recognition system that operates in real time on resource-constrained devices such as Raspberry Pi. The system addresses challenges like handling diverse background noise and maintaining low latency for real-world usability. By leveraging the Google Speech Commands Dataset, augmented with noise variants, and integrating advanced machine learning techniques, the project achieves competitive recognition accuracy. Furthermore, the system explores multi-language support, extending its application to French and Spanish commands. Optimizations such as quantization and pruning enable deployment on edge devices with minimal resource consumption.

# Introduction

Speech command recognition is a foundational component in many voice-activated systems, enabling hands-free control for smart home devices, mobile applications, and industrial automation. However, challenges such as handling noisy environments, achieving real-time processing, and adapting to resource-limited hardware require robust solutions.

Deploying such systems on edge devices like Raspberry Pi provides significant advantages, including improved response times, reduced reliance on cloud infrastructure, and enhanced user privacy. This is critical in applications where internet connectivity is intermittent or where latency-sensitive operations are needed.

This project aims to develop a speech recognition system that is:  
1. Noise-robust for effective command recognition in varying environments.  
2. Optimized for real-time performance on edge devices.  
3. Extended to support multiple languages for wider applicability.

# Dataset Preparation

The Google Speech Commands Dataset, containing one-second long audio clips of 30 common commands, served as the foundation for the project. To improve robustness, noise variants were generated using techniques such as adding white noise, background chatter, and environmental sounds using Librosa. Additional datasets in French and Spanish were integrated, mapping commands to equivalent phrases in these languages.

Noise augmentation ensures that the model can handle real-world scenarios where background noise is prevalent. For example, noise injection simulates environments like busy streets or crowded rooms. Augmenting the dataset also increases the diversity of the training data, making the model more generalizable.

# Feature Engineering

To enhance recognition accuracy, advanced audio features were extracted, including Mel-Frequency Cepstral Coefficients (MFCCs), Chroma Features, and Mel-Spectrograms. These features capture the temporal and spectral properties of the audio, which are critical for distinguishing between different commands.

Preprocessing steps involved resampling audio to 16 kHz for uniformity, normalizing signals to reduce variability, and applying data augmentation techniques like pitch shifting and time stretching. These techniques enhance the model's robustness by exposing it to diverse variations of the same audio commands.

# Model Architecture

A CNN-RNN hybrid model was designed to combine the strengths of both convolutional and recurrent layers. CNNs were used to extract local time-frequency patterns from audio signals, while RNN layers, specifically LSTM or GRU, captured long-term dependencies. An attention mechanism was included to focus on critical audio segments, enhancing the model's ability to prioritize relevant parts of the input.

The architecture also includes dropout layers to reduce overfitting and fully connected layers for final classification. The combination of CNN and RNN layers ensures that both spatial and temporal features are effectively learned.

# Training and Optimization

The model was trained using the AdamW optimizer with weight decay, alongside techniques like early stopping and learning rate scheduling to prevent overfitting. Data augmentation was extensively used during training to make the model robust against various noise conditions.

Quantization and pruning methods were applied to optimize the model for edge deployment. Quantization reduces the precision of the model's weights, while pruning removes less significant connections in the network. These optimizations result in a smaller and faster model without significant loss in accuracy.

# Results and Evaluation

The system achieved an accuracy of ~95% in clean environments and maintained a Word Error Rate (WER) of 8% in clean conditions and 15% in noisy conditions. Multi-language support was successfully implemented, achieving an accuracy of 85%+ for French and Spanish commands.

Evaluation metrics included confusion matrices and classification reports, which showed high precision and recall for most commands. The model's robustness was further validated by testing it on noisy environments, where it consistently outperformed baseline models.

# Deployment

The model was converted to TensorFlow Lite for deployment on a Raspberry Pi. Real-time audio processing was implemented, achieving inference latency below 100ms per command. Dependencies were set up on the Raspberry Pi, and a lightweight microphone setup was integrated.

Deployment involved several steps, including installing TensorFlow Lite runtime, configuring the Raspberry Pi hardware, and integrating the microphone. Real-time inference was achieved using low-latency audio processing libraries.

# Challenges and Resolutions

Challenges included handling background noise and meeting the hardware constraints of edge devices. Robust data augmentation techniques and model compression methods were used to address these challenges. For example, noise injection helped the model generalize better, while quantization reduced the model's memory footprint.

Another challenge was achieving low latency during real-time inference. This was resolved by optimizing the audio processing pipeline and ensuring efficient implementation on the Raspberry Pi.

# Conclusion and Future Work

The project successfully developed a noise-robust speech command recognition system deployable on resource-constrained devices. Multi-language support and real-time inference were achieved, showcasing the system's potential for IoT applications. Future work includes expanding language support to other widely spoken languages and further optimizing the system for ultra-low-power devices.

Additionally, integrating this system with other IoT devices could create a seamless ecosystem for voice-controlled applications.