Vishnuvaradhan Moganarengam (VXM210090)

EEDG6302: Microprocessor and Embedded Systems

Wednesday Lab Report

Project 3 Lab 2: Audio Classification

Aim:

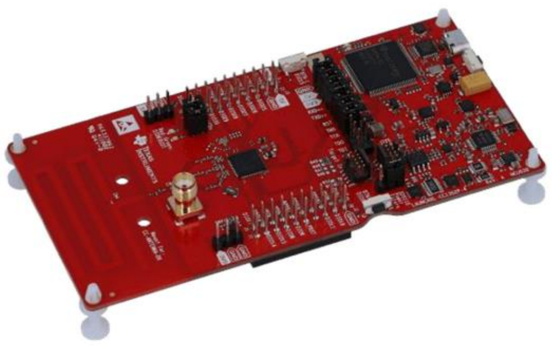
• A brief understanding about ML algorithms
• A brief understanding about TI Launchpad (CC1352P) and Booster Sensors
• Use machine learning to build a system that can recognize when a particular sound is happening—a task known as audio classification.

Introduction:

Machine learning is a branch of artificial intelligence that involves machines imitating intelligent human behavior to perform complex tasks. Tiny machine learning, or TinyML, is a newly emerging field that combines machine learning and embedded systems. Embedded systems are small computing devices that operate with extremely low power and can run for extended periods on small batteries. TinyML involves shrinking deep learning networks to fit on tiny hardware, making it possible to bring artificial intelligence to intelligent devices.
Edge computing, on the other hand, involves bringing computation and data storage closer to the source of data. Most edge devices in IoT ecosystems are designed to collect sensor data and transmit it to local or remote cloud platforms.

Texas Instruments CC1352P LaunchPad:

This LaunchPad speeds development on devices with integrated power amplifier and multi-band radio support for concurrent Sub-1Ghz and 2.4-GHz operation. Protocols supported include Bluetooth Low Energy, Sub-1 GHz, Thread, Zigbee, 802.15.4, and proprietary RF with the compatible CC13x2-CC26x2 SDK. It has Broad band antenna support for Sub-1 GHz (868 MHz / 915 MHz / 433 MHz) and 2.4 GHz frequency bands.



Edge Impulse

Edge Impulse is a cloud-based platform that enables the development of machine learning models for TinyML devices. The platform supports automated machine learning (AutoML) processing for edge devices and offers support for various boards, including smartphones, to deploy machine learning models on these devices. With Edge Impulse, developers can create and train machine learning models using data from sensors, audio, and other sources, and then deploy those models on edge devices for real-time processing.

Sampling New Data

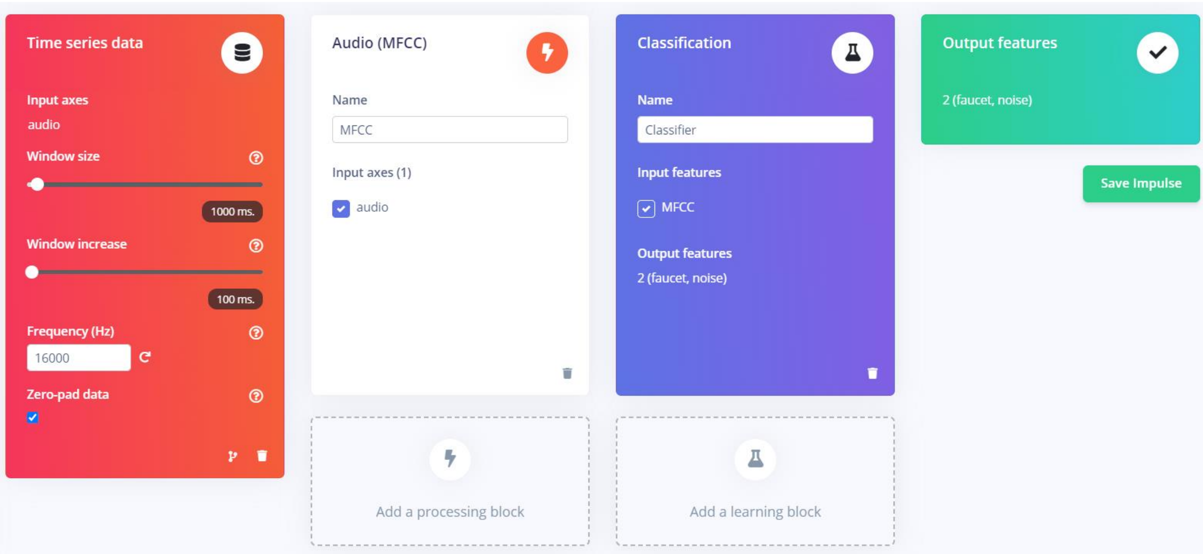
Machine learning works best with lots of data, so a single sample won't cut it. Now is the time to start building your own dataset. The device will capture a second of audio and transmit it to Edge Impulse:

Noise - 5 minutes of background noise
Faucet - 5 minutes of running faucet

Designing an Impulse

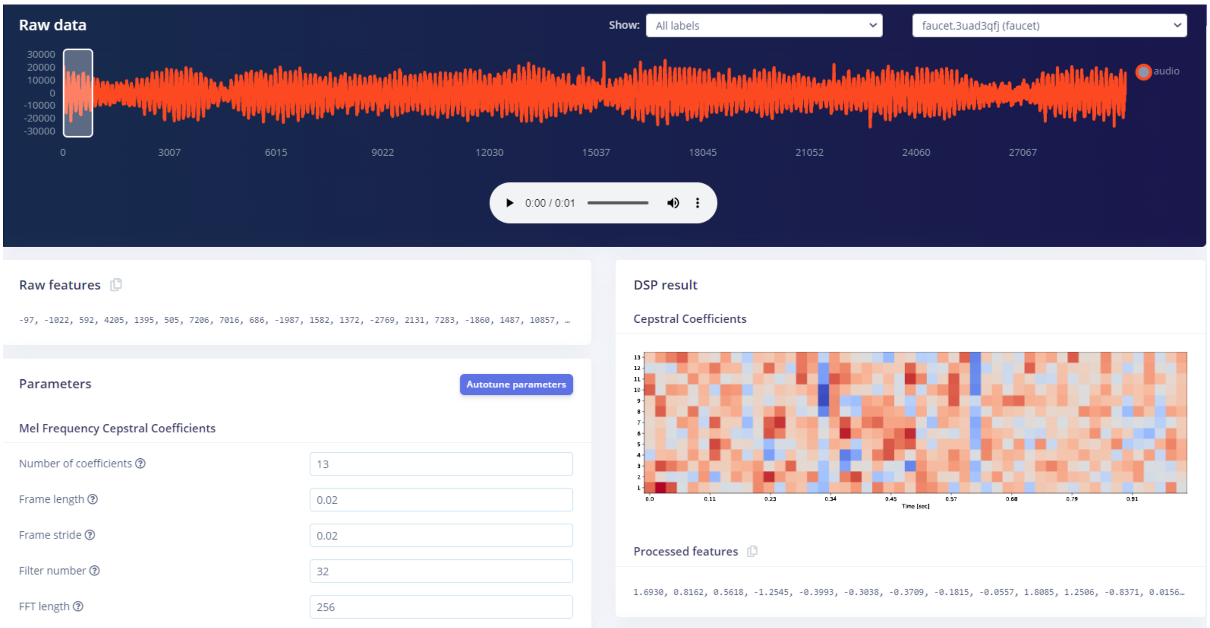
After setting up the training dataset, an impulse is created in Edge Impulse. The impulse takes the raw data and divides it into smaller windows, which are then processed by signal processing blocks to extract relevant features. These blocks always produce the same output for the same input and are used to simplify the raw data for processing. Next, a learning block is used to classify new data based on what has been learned from the training dataset. Learning blocks improve with experience by continuously learning from past examples. By combining signal processing and learning blocks, impulses are able to accurately classify new data in real-time, even on resource-constrained edge devices.

For this project we'll use the 'Spectral analysis' signal processing block. This block applies a filter, performs spectral analysis on the signal, and extracts frequency and spectral power data. Then we'll use a 'Neural Network' learning block, that takes these spectral features and learns to distinguish between the two (Noise, Faucet) classes.



Configuring the MFE block

The MFE (Mel Frequency Energies) block in audio processing takes a window of audio data and transforms it into a table of data where each row represents a specific range of frequencies, and each column represents a specific time span. The value in each cell of the table corresponds to the amplitude of the frequencies in that range during that time span. When visualized as a spectrogram, each cell in the table is represented by a colored block whose intensity varies based on the amplitude of the frequencies. This allows for a visual representation of the audio data that can be used for further analysis and processing.



Configuring the Neural Network

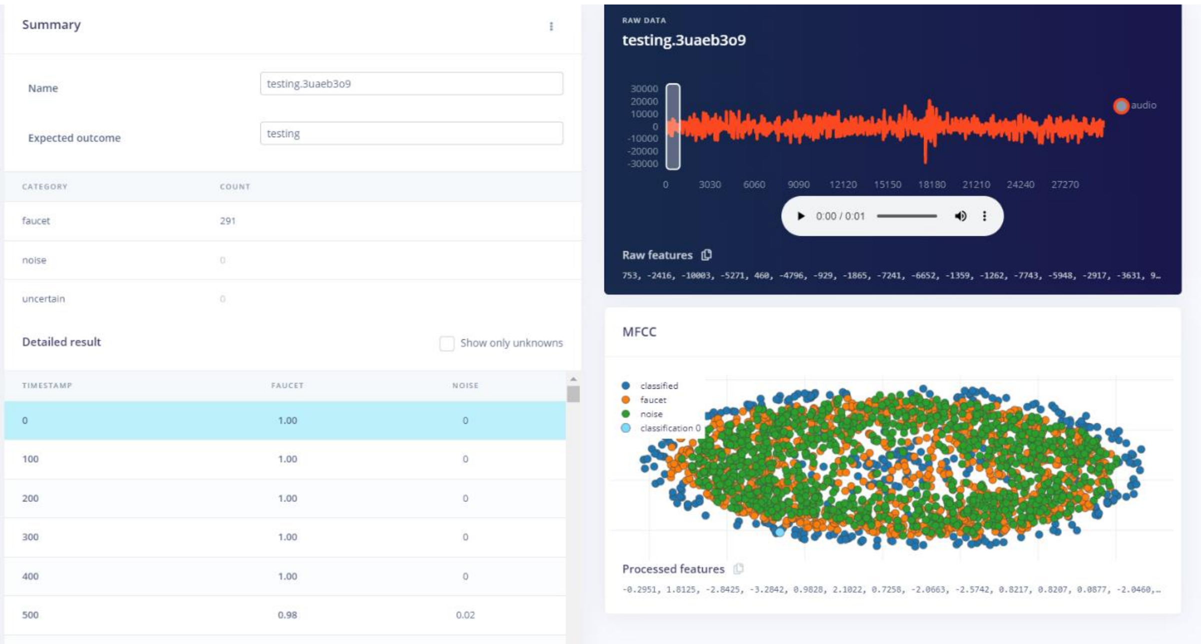
Neural networks are a group of algorithms that are loosely based on the structure and function of the human brain, and are designed to identify patterns in data. In this particular case, the neural network we are training will take the signal processing data as input, and attempt to map it to one of four classes. By analyzing the input data and recognizing patterns, the neural network can learn to accurately classify new data based on what it has learned during the training process.

A neural network is composed of interconnected layers of neurons, where each connection between neurons has a weight assigned to it. In the case of this specific neural network, there would be a neuron in the input layer that corresponds to the height of the first peak of the X-axis data obtained from the signal processing block, and a neuron in the output layer that corresponds to each of the four classes, including "wave". Initially, when defining the neural network, all the connections between the neurons are assigned random weights. As a result, the neural network will make random predictions until it is trained using a labeled dataset to adjust the weights of its connections and improve its accuracy in making predictions.

During the training process, the neural network is presented with the raw data and asked to make a prediction based on the current weights of its connections. The outcome of the prediction is then compared to the true label of the data, and the weights of the connections are adjusted slightly based on the error between the predicted and true labels. This process is repeated iteratively with all the training data, gradually improving the accuracy of the network's predictions over time. This is why it is important to have labeled data during the training process, as it allows the network to learn from its mistakes and adjust its weights to better classify new, unseen data.

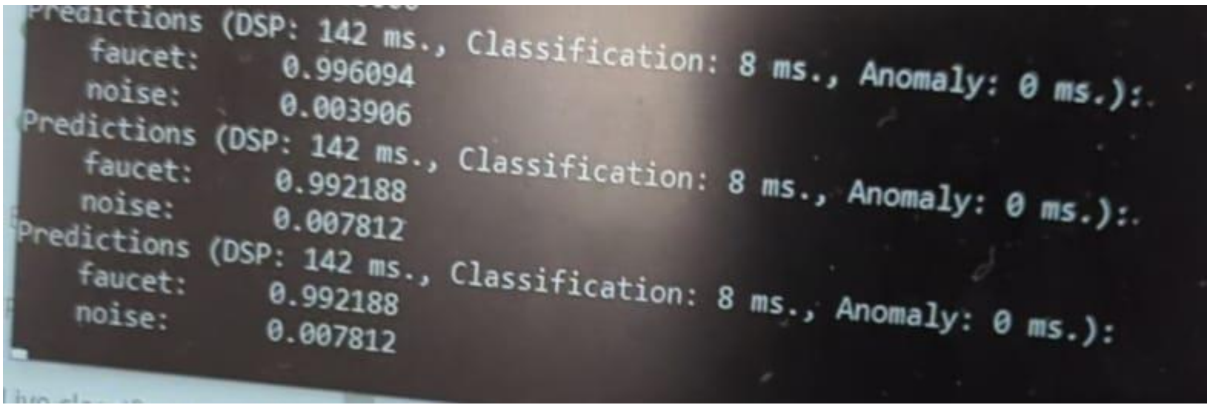
Classifying New Data

Neural networks are given more data to learn patterns in datasets by classifying the data properly, classifying new data is fairly simple by increasing the testing data to train neural networks.



Deploying Back to Device

Once the impulse is designed, trained, and verified, it can be deployed back to the target device. This allows the model to run locally without the need for an internet connection, which minimizes latency and reduces power consumption. To accomplish this, Edge Impulse can package the complete impulse, which includes the signal processing code, neural network weights, and classification code, into a single C++ library. This library can be included in the embedded software on the device, allowing it to run the machine learning model with minimal overhead.



The output shows the prediction of the sample as the 99% probability of being faucet.