Title page: EEDG/CE 6302 Lab Report 4
CE6302, Embedded Systems, 302 Laboratory - 83204
Lab Topic: - TinyML - Audio Classification
Yuyang Hsieh, YXH230019
Lab partner name: Xiongtao Zhang
Group Number 4
Instructor name: Tooraj Nikoubin
TA name: Seyed Saeed
Date: 9/21/2024

1. Objective:

Purpose: The purpose of this lab is to gain an understanding of TinyML and machine learning algorithms by building an audio classification system. Using the TI Launchpad (CC1352P) and CC3200 Audio BoosterPack, the system will recognize specific sounds, such as running water, amidst background noise. This lab provides practical experience in data collection, signal processing, and deploying machine learning models on embedded devices for real-time classification.
Methods used: Lab 4 uses TinyML to implement machine learning based audio detection into embedded devices (CC1352P and CC3200). TinyML is a field of study in Machine Learning and Embedded Systems that explores machine learning on small, low-powered microcontrollers, enabling secure, low-latency, low-power and low-bandwidth machine learning inferencing on edge devices. To demonstrate a completed flow of this, lab 4 includes data acquisition and analysis, model training and evaluation, device deployment and audio detection performing.
Results: After deploying the trained model into the launchpad, it can detect the audio in real time and use machine learning to generate classification results. When playing the audio of the faucet, it gave the prediction of the faucet with 0.99, and when playing the audio of other noise or detect ambient sound, it gave the prediction of noise with 0.99. This result shows the device works properly on detecting and distinguishing the faucet sound and other noise.
Hardware used: TI Launchpad (CC1352P), CC3200 Audio BoosterPack. Software used: Edge Impulse CLI, Texas Instruments UniFlash, and Edge Impulse.
Major Conclusions: By collecting data, analyzing data, extracting features, training model, evaluating model and deploying model, the device can detect and distinguish the sound of faucet and noise with high accuracy.

2. Introduction:

2.1 Hardware and Software Background Information
Hardware Background Information:TI Launchpad (CC1352P):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. Available variations feature different RF matching networks and power levels.
CC3200 Audio BoosterPack: Enables the evaluation and development with the digital audio peripheral [I2S] present on the SimpleLink™ Wi-Fi® CC3200 device. It contains a Class-D power amplifier to drive speakers and an ultra-low power audio codec, TLV320AIC3254, supporting programmable audio processing. Speakers, headsets, and microphones are sold separately.

Software Background Information: Edge Impulse is ushering in the future of embedded machine learning by empowering developers to create and optimize solutions with real-world data. It makes the process of

building, deploying, and scaling embedded ML applications easier and faster than ever, unlocking massive value across every industry, with millions of developers making billions of devices smarter.

2.2 Purpose of the Experiment

The purpose of lab 4 is using TinyML to perform audio detection tasks on microcontrollers based on machine learning. The lab demonstrates a completed workflow of building model. It includes collecting data from embedded device and upload to cloud platform, Edge Impulse, through serial ports; Processing signals and distinguishing between the two sounds (faucet and noise) with designing an impulse on cloud; Generating features based on previous data and training neural network model; After that, classifying the new Data with trained model to evaluate its performance; Finally, deploying the model back to device.

2.3 Summary of the Experiment

In lab 4, a audio classification task is performed on embedded device with low-power consumption and in real-time. TinyML is the core tool to achieve these targets. From data collection, signal processing to generating features, training model, validating the model, and deployment, the procedures of applying TinyML is completed. By applying TinyML, computation-intensive and memory-intensive applications are performed on the cloud, models and related applications are optimized to be able to run on the microcontrollers and embedded systems.

2.4 Findings of the experiment

By analyzing collected audio data, we learned to process MFCC audio signal and extract features from it. In model evaluation, we learned to use confusion matrix to measure the performance of the trained model; We also used test dataset to validate the model, as well as live data.

3. Explanation:

3.1 Experiment procedure

To set up the CC3200-LAUNCHXL with the CC3200AUDBOOST audio booster pack, begin by addressing the pin conflict on header J3. Pins 26-30 on J3 conflict with the CC3200AUDBOOST and need to be disconnected. Texas Instruments recommends bending these pins down to prevent interference. Since Edge Impulse-supported booster packs do not use these pins, this modification ensures proper functioning. To interface the boards, connect the following pins: P1-2 (NC) to TP13, P3-2 (GND) to P3-9 (DIN), P3-3 (NC) to P3-10 (DOUT), P3-6 (NC) to P4-9 (NC), and P3-4 (NC) to P3-7 (FSYNC). These connections will power the audio booster pack and allow access to the I2C lines on the CC3200-LAUNCHXL.

Next, open a terminal and run the edge-impulse-daemon command, ensuring the correct COM port is selected for the Launchpad. For data acquisition, use Edge Impulse and configure the microphone sensor to capture a 10-second sample with the label "noise." Collect 2 minutes of background noise labeled as "noise" and 2 minutes of running faucet noise labeled as "faucet," ensuring at least 48 seconds of test data. Real-world noise often contains background sounds such as TV, kids playing, or cars passing by, while running faucet sounds may be accompanied by dishes or other kitchen activities. Capturing these variations helps improve model performance.

To design the impulse, start with time series data and set the window size to 1000. The processing block should be a feature extractor, using MFCC (Mel-frequency cepstral coefficients) for audio signal processing. This digital signal processing technique extracts important features from the audio, which the model will learn from. Following this, generate features using the MFCC block.

For the learning block, configure the neural network to train the model and recognize patterns in the audio data. Set the number of training cycles to 300 and the learning rate to 0.005 for effective learning. Once training is complete, test the model by performing live classification with a sample length of 1000ms. Finally, deploy the model by flashing the firmware onto the CC3200-LAUNCHXL, then run the model on the device using the command edge-impulse-run-impulse --continuous. This will allow the device to classify sounds in real-time based on the trained model.

3.2 Experiment Images

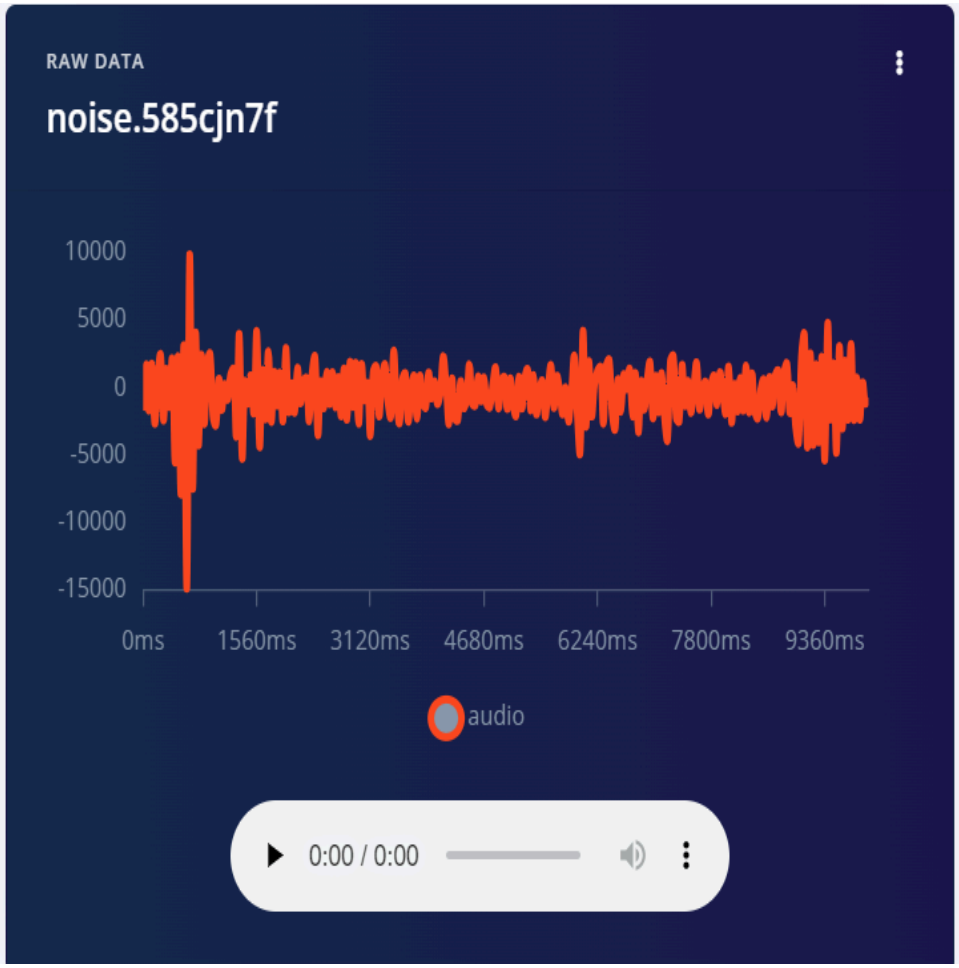

Image 3.2-1: : Raw data graph for noise

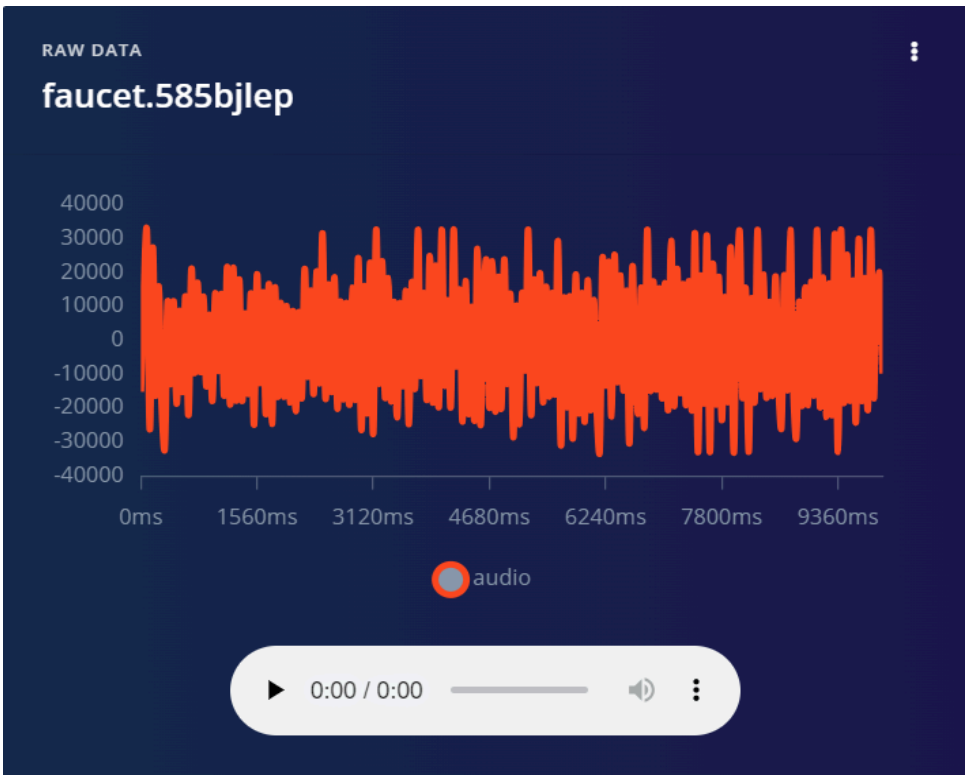

Image 3.2-2: : Raw data graph for faucet

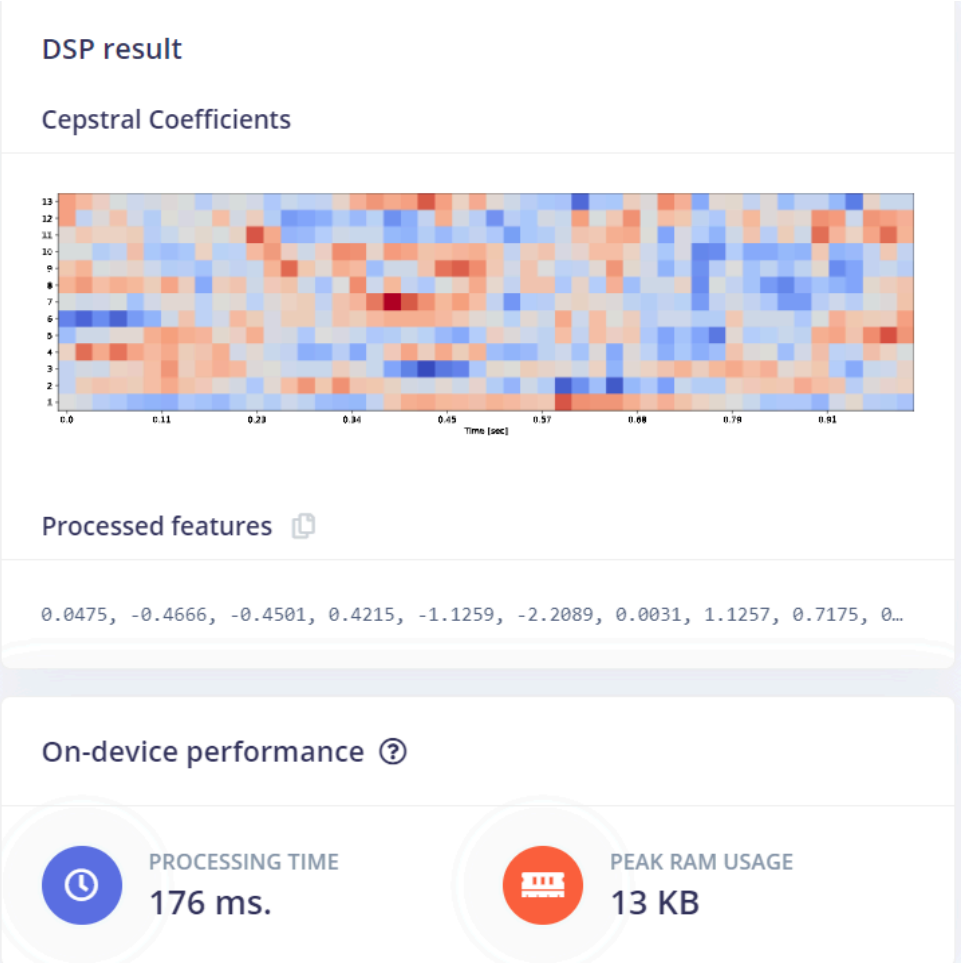

Image 3.2-3: Spectrogram of background noise

Image 3.2-4: Spectrogram of running faucet

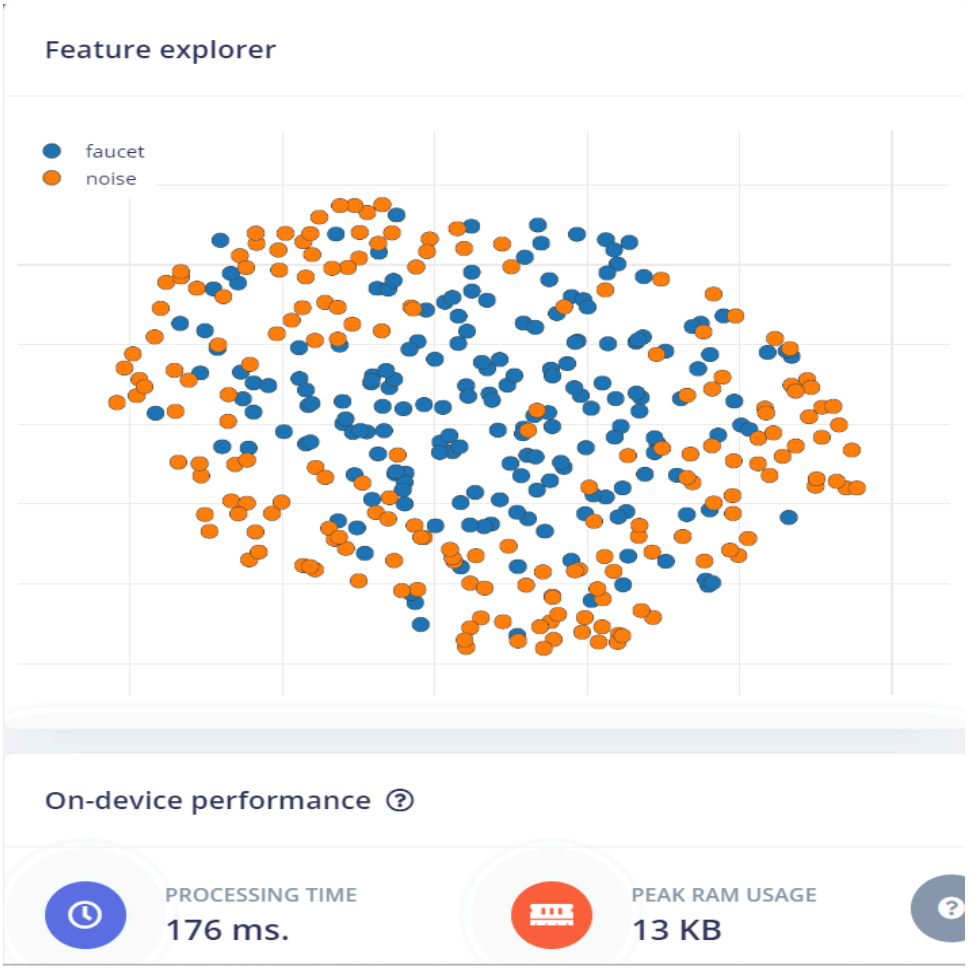

Image 3.2-5: Feature generation proce

Image 3.2-6: The Model Panel for Neural Network

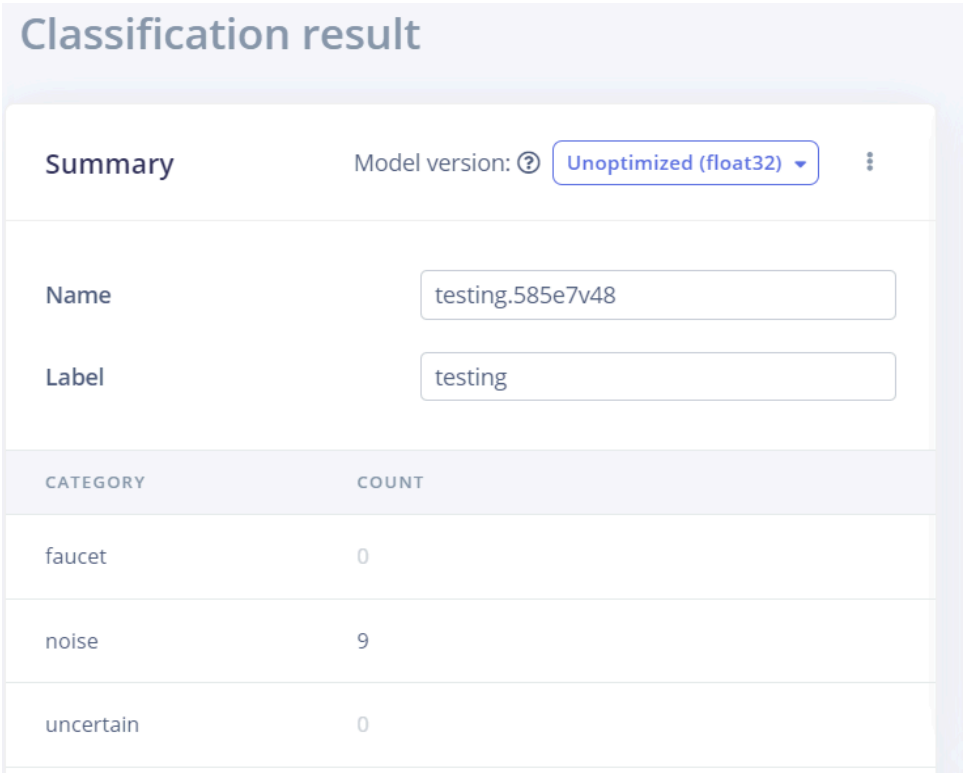

Image 3.2-7: live classification for noise

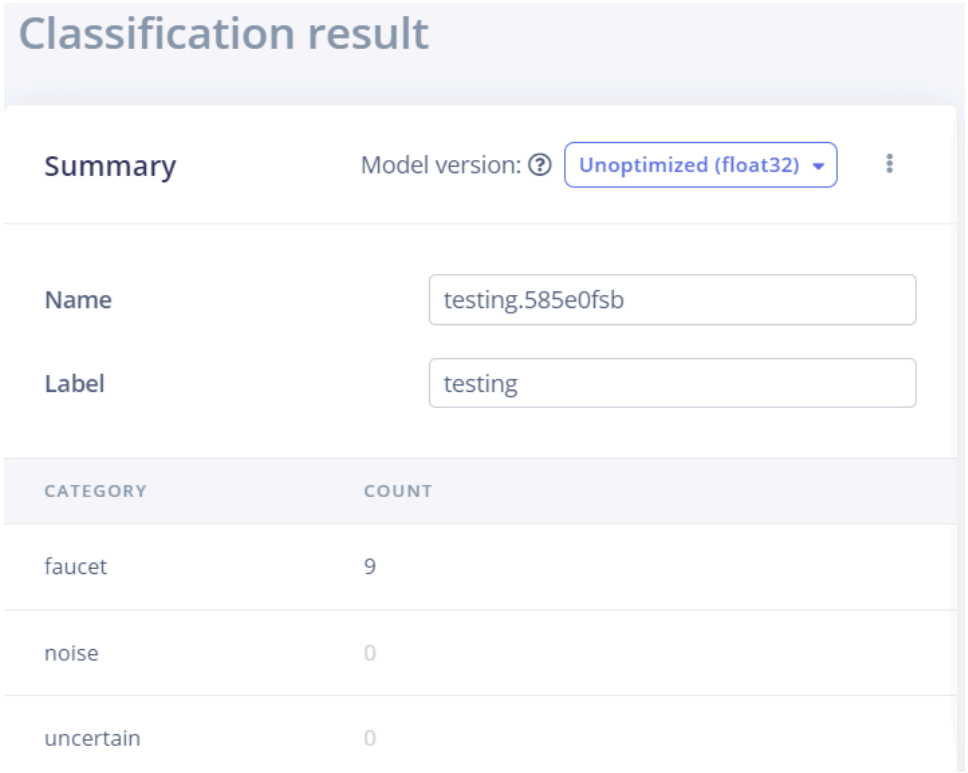

Image 3.2-8: live classification for faucet

Image 3.2-9: model testing including test data

```
    noise: 0.996094
Predictions (DSP: 148 ms., Classification: 9 ms., Anomaly: 0 ms.):
#Classification results:
    faucet: 0.000000
    noise: 0.996094
Predictions (DSP: 148 ms., Classification: 9 ms., Anomaly: 0 ms.):
#Classification results:
    faucet: 0.000000
    noise: 0.996094
Predictions (DSP: 148 ms., Classification: 9 ms., Anomaly: 0 ms.):
#Classification results:
    faucet: 0.000000
    noise: 0.996094
Predictions (DSP: 148 ms., Classification: 9 ms., Anomaly: 0 ms.):
#Classification results:
    faucet: 0.000000
    noise: 0.996094
Predictions (DSP: 147 ms., Classification: 9 ms., Anomaly: 0 ms.):
#Classification results:
    faucet: 0.000000
    noise: 0.996094
```

Image 3.2-10: Output for noise

```
    noise: 0.000000
Predictions (DSP: 148 ms., Classification: 8 ms., Anomaly: 0 ms.):
#Classification results:
    faucet: 0.996094
    noise: 0.000000
Predictions (DSP: 148 ms., Classification: 8 ms., Anomaly: 0 ms.):
#Classification results:
    faucet: 0.996094
    noise: 0.000000
Predictions (DSP: 148 ms., Classification: 8 ms., Anomaly: 0 ms.):
#Classification results:
    faucet: 0.996094
    noise: 0.000000
Predictions (DSP: 148 ms., Classification: 8 ms., Anomaly: 0 ms.):
#Classification results:
    faucet: 0.996094
    noise: 0.000000
Predictions (DSP: 148 ms., Classification: 8 ms., Anomaly: 0 ms.):
#Classification results:
    faucet: 0.996094
    noise: 0.000000
Predictions (DSP: 148 ms., Classification: 9 ms., Anomaly: 0 ms.):
#Classification results:
    faucet: 0.996094
    noise: 0.000000
Predictions (DSP: 148 ms., Classification: 9 ms., Anomaly: 0 ms.):
#Classification results:
    faucet: 0.996094
    noise: 0.000000
```

Image 3.2-11: Output for faucet

4. Discussions and Conclusions:

4.1 Comparison of your experimental results with the research/theory of the concept with reasoning.
The experimental results are quite consistent with the theoretical concepts of TinyML and audio classification. TinyML is designed to make it possible to run machine learning tasks, including real-time audio classification, right on low-power, resource-constrained devices, such as the TI Launchpad CC1352P. In theory, this will allow for fast, efficient processing with very minimal latency. The lab could then achieve this by successfully deploying a trained neural network onto the device that could classify things like running water and background noise in real time.

The theoretical advantages of feature extraction in improving model accuracy were further validated by the use of MFCC for signal processing in order to draw out key features in audio data. From classification results, it was assured that deep neural networks can learn to identify certain patterns within noisy conditions. In general, the experimental results verified that sound recognition running in real time on edge devices is possible and will be efficient, just as the TinyML principles state.

4.2 Learnings from the experiment
In this lab, it involves an audio classification system with the use of machine learning, where specific sounds-a perfect example would be a running water sound from a faucet-end are recognized even when there is surrounding noise. Additionally, one will get hands-on experience with different stages of the machine learning process from collecting audio data to processing a signal for the extraction of important features-such as through MFCC-to training a deep neural network over those sounds into desired classes.

The hands-on experience that this trained model is going to get deployed on an embedded device like CC3200-LAUNCHXL gives experience in flashing firmware, interfacing hardware components, and performance evaluation of the model in real time. Such a project increases the knowledge in signal processing, neural network training, and TinyML while showing the challenges when real-world data-for example, variation of background noise-is used. It highlights a few basic key ideas of embedded machine learning: how processing has to be efficient and adaptable for deployment on low-power devices.

5. References:

Texas Instruments: https://www.ti.com/
Texas Instruments, Educational BoosterPack MkII Guide: https://www.ti.com/lit/ug/slau599b/slau599b.pdf
Energia: https://energia.nu/
Texas Instruments CC1352P LaunchPad datasheet: https://www.ti.com/lit/ds/symlink/cc1352p.pdf?ts=1649268244550&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FCC1352P
Audio BP User Guide: https://www.ti.com/lit/ug/swru383a/swru383a.pdf?ts=1636125907650&ref_url=https%253A%252F%252Fwww.ti.com%252Ftool%252FCC3200AUDBOOST

Seeed Studio, TinyML: https://wiki.seeedstudio.com/tinyml_topic/