Car Whispering: the AI Mechanic
TinyML Audio Event Detection

Eoin Jordan¹, Dr Martin Serrano², and Pearse Gough³

¹ Data Science Institute, National University of Ireland Galway, Ireland
² Data Science Institute, National University of Ireland Galway, Ireland
³ Google, Ireland
My Background

- Software & Electronics Engineer 10+ years
- NUIG DSI (PhD Researcher on semantic technologies supervised by Dr Martin Serrano) 4 Months
- CoderDojo (Mentor) 4+ years
- Galway National Park City (Maker Champion) 2+ years
- RethinkWaste.ie (Co-Founder, Voluntary Director and Product Designer) 3+ Years
- Driver of 15+ year old cars for longer than I care to say.
Introduction

• Starting my journey with ML and have a history in Electronics

• To make peace with my limitations as a mechanic “Knowing what you don't know”

• Mysterious sounds coming from my car I cannot identify

• Fault Codes were not helping for certain issues

• Advent of TinyML via the TinyML Foundation and the contributions to the movement by TensorFlow Lite (Google), and newer companies lowering the bar to entry such as Edge Impulse.

https://semiengineering.com/why-tinyml-is-such-a-big-deal/
Aim

• To generate interest in our AI Mechanic project.
• To demonstrate the low barrier into TinyML.
• To also demonstrate an easy route to building prototypes with sensors.
• To gather feedback from the audience on this topic, and how to best start it as an open source hardware software project.
• A little knowledge is a dangerous thing. (a recurring theme today). Lets try to grow ours.

• After fixing an oil pressure issue in my car I introduced a new self oiling feature to the system, and the only error I had was not associated with oiling what I didn’t know.
The Solution?

Record the desired state and wait for the next issue to arise (fortunately my car is old enough to provide useful data)

- Allow the professionals to do the work and try to learn from them.
- Capture audio from a car with an issue get it fixed and record the audio of the fixed car.
- Label the data.
Considering the options

• Cloud based connected via phone hotspot
• TinyML[1] finally brings independent Intelligence to the Edge of the Network

Why on a Microcontroller?

• Inexpensive
• Low Power
• Handheld
• No Connectivity Required
• Accessible and easy to deploy to
• Privacy* ...When not connected to a network

[2]https://www.tensorflow.org/lite/microcontrollers#why_microcontrollers_are_important
Edge Intelligence Main Challenges\textsuperscript{[3]}

- Open Architecture
- Modelling and Performance Analysis
- Heterogenous Wireless Networking
- Resource Allocation and Energy Efficiency
- QoS and QoE Provisioning
- Security and Privacy Concerns
- Federation and Cross Platform Service Supply

Tiny machine learning is broadly defined as a fast growing field of machine learning technologies and applications including hardware, algorithms and software capable of performing on-device sensor data analytics at extremely low power, typically in the mW range and below, and hence enabling a variety of always-on use-cases and targeting battery operated devices. ”[5]

“TinyML, it’s both a concept and an organization — and it has acquired significant momentum over the last year or two. ”[4]

Options Considered:

TensorFlow Lite

Automated Workflow for Generation of Models (Edge Impulse[6] as it has the Apache 2.0 license)
Selecting the Hardware

The following development boards are supported by Tensorflow Lite Officially[6]:

- Arduino Nano 33 BLE Sense[7]
- SparkFun Edge
- STM32F746 Discovery kit
- Adafruit EdgeBadge
- Adafruit TensorFlow Lite for Microcontrollers Kit
- Adafruit Circuit Playground Bluefruit
- Espressif ESP32-DevKitC
- Espressif ESP-EYE
- Wio Terminal: ATSAMD51
- Himax WE-I Plus EVB Endpoint AI Development Board
- Synopsys DesignWare ARC EM Software Development Platform
- Sony Spresense

I chose to use:

- Arduino Tiny Machine Learning Kit (Arduino Nano 33 BLE Sense)
- M5 Echo Atom (ESP32)[8]

[8] https://docs.m5stack.com/en/atom/atomecho
YAMNet and AudioSet

- YAMNet is an audio event classifier suggested in the audio event example on TensorFlow's tutorial, which samples a given audio waveform and makes predictions the given Audio scenarios are described by the AudioSet ontology. [10]

- This is the most relevant area for my research but may not be of interest widely in the audience. I see myself contributing mainly on this part of the project. Engine Starting, Engine Knocking. Heavy Engine “We estimate this class has low quality” [12].

http://research.google.com/audioset/

http://research.google.com/audioset/dataset/heavy_engine_low_frequency.html

http://www.jordipons.me/apps/audioset/

[10] https://tfhub.dev/google/yamnet/1
Tensorflow Lite Micro

- TensorFlow Lite for Microcontrollers is a port of TensorFlow Lite, designed to run machine learning models on DSPs, microcontrollers and other devices with limited memory.[14]

[14]https://www.tensorflow.org/lite/microcontrollers#why_microcontrollers_are_important
Capturing some Data

• Recording the good and bad states and annotating them. I used Edge Impulses collection app (which is also open source Apache License 2.0)

[https://smartphone.edgeimpulse.com/classifier.html](https://smartphone.edgeimpulse.com/classifier.html)
[https://studio.edgeimpulse.com/studio/69300/learning/keras/7](https://studio.edgeimpulse.com/studio/69300/learning/keras/7)
Training the neural network

[Image of a neural network training interface with details on settings and performance metrics.]
Training the Model

Retrain model with known parameters

- MFE
- Spectrogram
- NN Classifier

Build output

Scheduling job in cluster...
Job started
Reducing dimensions for visualizations...
UMAP(n_components=1, verbose=True)
Construct fuzzy simplicial set
Sat Feb 5 23:19:38 2022 Finding Nearest Neighbors
Still running...
Sat Feb 5 23:19:42 2022 Finished Nearest Neighbor Search
Sat Feb 5 23:19:44 2022 Construct embedding
Still running...
completed 0 / 500 epochs
completed 98 / 500 epochs
FOSDEM '22

Accuracy

Training output

Model

Last training performance (validation set)

<table>
<thead>
<tr>
<th>Model</th>
<th>Version:</th>
<th>Quantized (int8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model version:</td>
<td></td>
<td>Quantized (int8)</td>
</tr>
</tbody>
</table>

Last training performance (validation set)

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Accuracy</strong></td>
<td>61.4%</td>
</tr>
<tr>
<td><strong>Loss</strong></td>
<td>1.23</td>
</tr>
</tbody>
</table>

Confusion matrix (validation set)

<table>
<thead>
<tr>
<th></th>
<th>BACKGROUND</th>
<th>AIR LEAK</th>
<th>AIR LEAK ENG</th>
<th>BACKGROUND</th>
<th>IDLING</th>
<th>NORMAL ENG</th>
<th>OIL CAP OFF</th>
</tr>
</thead>
<tbody>
<tr>
<td>BACKGROUND NO</td>
<td>61.3%</td>
<td>0%</td>
<td>35.0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>AIR LEAK</td>
<td>0%</td>
<td>88.3%</td>
<td>11.1%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>AIR LEAK ENGINE</td>
<td>0%</td>
<td>0%</td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>BACKGROUND NO</td>
<td>0%</td>
<td>96.7%</td>
<td>0%</td>
<td>2.2%</td>
<td>1.1%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>IDLING</td>
<td>0%</td>
<td>32.1%</td>
<td>42.0%</td>
<td>0%</td>
<td>25%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>NORMAL ENGINE</td>
<td>0%</td>
<td>0%</td>
<td>42.1%</td>
<td>0%</td>
<td>0%</td>
<td>53.4%</td>
<td>3.8%</td>
</tr>
<tr>
<td>OIL CAP OFF ENGI</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>100%</td>
</tr>
<tr>
<td>F1 Score</td>
<td>0.76</td>
<td>0.70</td>
<td>0.68</td>
<td>0.04</td>
<td>0.39</td>
<td>0.70</td>
<td>0.96</td>
</tr>
</tbody>
</table>

Feature explorer (full training set)

The feature explorer is only supported when you have a single DSP block.

On-device performance

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Inferencing Time</strong></td>
<td>38 ms.</td>
</tr>
<tr>
<td><strong>Peak RAM Usage</strong></td>
<td>11.8K</td>
</tr>
<tr>
<td><strong>Flash Usage</strong></td>
<td>221.7K</td>
</tr>
</tbody>
</table>
Implementing the Code

```cpp
#include <pwm.h>
#include <oled.h>

void led_lcd()
{
    // Audio buffers, pointers and selectors */
    typedef struct {
        uint8_t *buffer;
        uint8_t buf_ready;
        uint8_t buf_count;
        uint32_t n_samples;
    } inference_t;

    static inference_t inference;
    static signed short sampleBuffer[2048];
    static bool debug_on = false; // Set this to true to see e.g. features generated from the raw signal

    @brief Arduino setup function
    void setup()
    {
        // Initialize the digital pins as an output
        pinMode(LED_PIN, OUTPUT);
        pinMode(LED_PIN, INPUT);
        pinMode(LED_PIN, OUTPUT);
        // Put your setup code here to run once:
        Serial.begin(115200);
        // Set up the LCD's number of columns and rows:
        lcd.begin(16, 1);
        // Print a message to the LCD.
        lcd.print("Hello, FOSDEM!");

        // Summary of inference settings (from model_data.h)
        es_printf("Inference Settings: \n");
        es_printf("\nInterval: %d ms \n", (long)(CLASSIFIER_INTERVAL_MS);
        es_printf("Frame size: %d\n", CLASSIFIER_FRAME_SIZE);
        es_printf("Sample length: %d\n", CLASSIFIER_RAW_SAMPLES_COUNT / 16);
        es_printf("Classes: %d\n", sizeof(CLLS_CLASSIFIER_INFERENCING_CATEGORIES) / sizeof(CLLS_CLASSIFIER_INFERENCING_CATEGORIES));
        if (microphone_inference_started(0, CLASSIFIER_RAW_SAMPLES_COUNT) == false) {
            return;
        }
    }
```
Sharing the Model
Deploying the Model

**Grove:** “Grove is a modular, standardized connector prototyping system. Grove takes a building block approach to assembling electronics.

Compared to the jumper or solder based system, it is easier to connect, experiment and build and simplifies the learning system, but not to the point where it becomes dumbed down.

Some of the other prototype systems out there takes the level down to building blocks.

Good stuff to be learned that way, but the Grove system allows you to build real systems.

It requires some learning and expertise to hook things up.”[19]

Seeed Studio have great resources and library support for their models with useful examples.

Supply Chain withstanding you may need to search in multiple stores for your parts regardles.

I recommend avoiding cloned devices for your comfort building as drivers can cause issues on windows. Use reputable electronics stores (See references, please suggest any you may know). To avoid supporting potentially dangerous work environments for their producers, and help future developments.

3.3v compatible grove displays: [https://www.mouser.ie/c/?q=grove%20display](https://www.mouser.ie/c/?q=grove%20display)

FOSDEM '22

Edge Intelligence - Live Demo..
FOSDEM '22

Testing Via Edge Impulse

[21]https://docs.edgeimpulse.com/docs/running-your-impulse-arduino
Future Work

- Replace the display with a 3.3v one, parts ordered already.
- Combine Error Codes from OBD scanner to tag audio collected in real time by collecting via a streaming service.
- Share collected events via a network of users, interested in suggestions of how to implement that??
- Extend the AudioSet Ontology to include more mechanical terms. (Relevant to my research)
- Perform reasoning on the minified AI Mechanic Ontology
- Monitor Device Battery Performance
- Seeking contributors: https://github.com/eoinjordan/CarWhispering
- More target MCUs: BBC:Micro(My sons), ESP32-C3 (RISC-V), and of course the RP2040:
Future Work on i3-MARKET

- Automotive Data is yet a potential area for further exploration.
- Automotive Data Owners may want to share the collected data if there are incentives to do so.
- Data Sharing model by selling data sets via Data Marketplaces can act as an incentive.
- Creation and Support of Automotive Data Space is yet a requirement we will explore.
- Perform Data Cleaning and Define Data Formats for Sharing Automotive Collected Data is important.
- AI Edge Intelligence will benefit from more Available End User Automotive Data.
- Seeking contributors: https://github.com/eoinjordan/CarWhispering

FOSDEM ’22
Future Work on INFINITECH

- Detect a coercive event via a **microphone** and embedded model on an **MCU**
- Call an api via the **Modbus** or other secure wired system communication on the ATM that signals an investigation needs to take place on that persons account for possible human trafficking
- TAH external system is also triggered for **redflag indicators**[22] in the area.

https://www.howitworksdaily.com/how-do-cash-machines-work/[23]
https://www.traffikanalysis.org/[22]
https://www.infinitech-h2020.eu/[24]
Conclusions

- AI has become more accessible
- AI at the Edge or Edge Intelligence make for makers with TinyML (TensorFlow Lite), and everyone via intuitive tools such as Edge Impulse is a possibility
- Offline enhancements to existing erroring systems are now possible to implement cheaply.
- It is now possible to build a device that can give us a pocket car whispering mechanic. Very exciting!
Questions and Comments

• Thank you so much for your attention!

• Accepting contributors from all areas:

• Mechanics, automation, build, electronics, graphics, documentation, test, development, ML, no qualifications necessary, beginners, novice drivers or otherwise all are welcomed with open arms:

  • https://github.com/eoinjordan/CarWhispering
References


[2] https://www.tensorflow.org/lite/microcontrollers#why_microcontrollers_are_important


[8] https://docs.m5stack.com/en/atom/atomecho


[10] https://tfhub.dev/google/yamnet/1


[14] https://www.tensorflow.org/lite/microcontrollers#why_microcontrollers_are_important


[18] https://github.com/eoinjordan/CarWhispering


[21] https://docs.edgeimpulse.com/docs/running-your-impulse-arduino
Transfer Learning

• Audio events detected in other models can help identify similar types in another e.g. engine sounds used in the type of vehicle may help identify my idling one.

https://colab.research.google.com/github/tensorflow/tensorflow/blob/master/tensorflow/lite/g3doc/tutorials/model_maker_audio_classification.ipynb?authuser=2#scrollTo=wbMc4vHjaYdQ

https://www.tensorflow.org/tutorials/audio/transfer_learning_audio?authuser=2