EE P 596 - TinyML - Assignment 2

Total Points: 100
Spring Quarter, 2025
Department of Electrical and Computer Engineering
University of Washington, Seattle, WA 98195

Due: 11:59 pm (PST) on May 25 (Sun), 2025 via Canvas

Note:

- This homework contains both programming questions (marked as [**Pro**]) and discussion questions (marked as [**Dis**]).
- You may use and adapt Python functions and code provided in Lab 6 and in the course materials.
- Your homework submission must include: (i) a `.ipynb` file containing Python code for the programming questions; (ii) a `.pdf` file with discussion answers and key figures or screenshots; (iii) Arduino deployment code files (`.ino`, `.h`, `.bat` or any other executables) used in real-time testing.
- Name your files as follows:
"*#_$_EEP596_HW2.ipynb*" and "*#_$_EEP596_HW2.pdf*" where # and $ are your first and last names.

1. **[Pro] (Fine-Tuning Pretrained Model for Digit Classification, 50 points)**

   You will fine-tune the final layer of the pretrained CNN model from Lab 6 ("Magic Wand" gesture classifier) using a student-collected dataset of digit gestures (0–9). You may refer to and reuse code from Lab 6.

   (a) **Data Collection (10 points)**
   Follow the data collection procedure in Lab 6. If you did not collect your own gesture data previously, collect it now for digits 0 through 9. Store your recordings in a Google Drive folder and include the link in your submission.

   (b) **Dataset Preparation and Visualization (10 points)**
   From the shared Google Drive dataset (link will be posted), create a dataset with 30 examples per digit (0–9), totaling 300 samples. Rasterize each sample and randomly select 1 example per class (0–9) to visualize (10 plots total).

   (c) **Fine-Tuning and Classification Report (15 points)**
   Load the model `saved_model.keras` from Lab 6 and freeze all layers except the final Dense layer. Fine-tune the model on the dataset from part (b). Print the classification report (precision, recall, F1-score, accuracy) on the dataset from part (b) after fine-tuning.

   (d) **Deployment and Real-Time Testing (10 points)**
   Deploy the fine-tuned model to your Arduino Nano 33 BLE Sense. Submit one screenshot of real-time inference results for each digit (10 total) from the Serial Monitor. Accuracy is not required—just show that the system runs and attempts inference.

   (e) **Discussion (5 points)**
   In a short write-up, reflect on whether real-time results improved after fine-tuning. Why do you think the model works or does not work well? Suggest one or two improvements (e.g., more training data, dropout tuning, etc.).

2. **[Pro] (Audio Classification with Edge Impulse, 50 points)**

You will build and deploy a TinyML audio classifier using Edge Impulse and the Arduino Nano 33 BLE Sense.

(a) **Data Collection (10 points)**
Record 2–3 second audio samples for each class: **clap**, **knock**, and **silence**. Collect at least 50 samples per class using Edge Impulse's data collection tool or mobile app.

(b) **Feature Extraction (10 points)**
Use the **MFCC** processing block in Edge Impulse Studio to extract features. Submit screenshots or brief summaries showing the feature extraction step and MFCC settings used.

(c) **Model Training (10 points)**
Use Edge Impulse's **Transfer Learning** feature with **MobileNet 0.1** or a similar small CNN model. Train the model on your dataset and show the test accuracy and confusion matrix.

(d) **Model Optimization (10 points)**
Apply int8 quantization and EON compiler optimizations to reduce model size and improve latency. Submit evidence of model size reduction and show the test accuracy and confusion matrix for the compressed model.

(e) **Deployment and Results (10 points)**
Deploy the model to the Arduino Nano 33 BLE Sense using either the Arduino IDE (with TensorFlow Lite for Microcontrollers) or EON compiler method. Submit a screenshot of live inference results from your Arduino Nano 33 BLE Sense for each class (e.g., **clap**, **knock**, and **silence**) from the Serial Monitor or command line.

Submission Instructions for Q2:

- Submit a public sharing link to your Edge Impulse project.
- Include a ZIP file with deployment code generated from Edge Impulse.
- Include a PDF with model performance screenshots, Arduino output screenshots, and any comments.