# Model Deployment on Embedded Systems

https://github.com/as-budi/Embedded_AI.git

# 1. Apa itu EloquentTinyML?

- **EloquentTinyML** adalah library Arduino/C++ yang memungkinkan kita untuk menjalankan model machine learning berukuran kecil di **mikrokontroler** seperti **ESP32**, **ESP8266**, **Arduino Uno**, **Nano**, dll.
- Library ini membungkus **TensorFlow Lite for Microcontrollers (TFLite Micro)** dan membuat penggunaannya **jauh lebih simpel**.
- Tujuan utamanya: membuat proses deployment **ringan**, **cepat**, dan **tidak memerlukan TensorFlow kompleksitas penuh**.

## 2. Alur Umum Deployment ML ke ESP32

- **Latih Model** di komputer/laptop (menggunakan TensorFlow/Keras/Sklearn)
- **Pruning & Kuantisasi** model jika diperlukan
- **Ekspor Model** menjadi array C-byte menggunakan **everywhereml**
- **Gunakan EloquentTinyML** untuk memuat model ke ESP32
- **Inferensi**: berikan input sensor/data, dapatkan output prediksi

# ✨ A. Training Model

- Model dilatih di PC/laptop menggunakan Keras atau Tensorflow.
- Biasanya modelnya sederhana: **MLP (Multilayer Perceptron)**, **Decision Tree**, **SVM**, **Random Forest**, atau bahkan **small CNN**.
- Contoh training model untuk dataset iris: https://github.com/as-budi/Embedded_AI/blob/main/09_Model_Deployment_on_Embedded_Systems/09_codes/train-tensorflow-model-for-arduino.ipynb

## ✨ B. Pruning & Kuantisasi (Opsional tapi Penting)

- Pruning & Kuantisasi membuat model lebih kecil, lebih cepat, dan kompatibel dengan memori terbatas.
- **Lihat Materi sebelumnya tentang Pruning dan Kuantisasi!**

# ✨ C. Ekspor Model

- Model hasil training perlu diubah menjadi array C-byte agar bisa di-include ke dalam program Arduino.
- Untuk mengekspor model ini dibutuhkan library `everywhereml`.

```
pip install "everywhereml>=0.2.32"
```

- contoh baris code yang digunakan untuk mengekspor model:

```
from everywhereml.code_generators.tensorflow import convert_model
c_header = convert_model(model, X, y, model_name='irisModel')
print(c_header)
```

- Hasilnya dapat disalin dan disimpan dalam file `model.h`:

```
#pragma once
...
const unsigned char irisModel[] DATA_ALIGN_ATTRIBUTE = { 0x1c, 0x00, 0x00, 0x00,
...
0x54, 0x46, 0x4c, 0x33, 0x14, 0x00, 0x00, 0x09 };
```

- Bagian yang disalin adalah mulai dari `#pragma once` sampai dengan akhir dari isi variabel `const unsigned char irisModel[] DATA_ALIGN_ATTRIBUTE`.

# ✨ D. Deployment dan inferensi di ESP32

- Untuk melakukan deployment dan inferensi di ESP32 melalui Arduino IDE, perlu diinstal library:
  - EloquentTinyML (https://github.com/eloquentarduino/EloquentTinyML)
  - tflm_esp32 (https://github.com/eloquentarduino/tflm_esp32)
- Di Arduino IDE, include `EloquentTinyML.h`.

# Contoh basic code:

```
#include "irisModel.h"
#include <tflm_esp32.h>
#include <eloquent_tinyml.h>
#define ARENA_SIZE 2000

Eloquent::TF::Sequential<TF_NUM_OPS, ARENA_SIZE> tf;

void setup() {
   Serial.begin(115200);
   delay(3000);
   Serial.println("__TENSORFLOW IRIS__");
   tf.setNumInputs(4);
   tf.setNumOutputs(3);

   tf.resolver.AddFullyConnected();
   tf.resolver.AddSoftmax();

   while (!tf.begin(irisModel).isOk())
      Serial.println(tf.exception.toString());
}


void loop() {
  // classify class 0
  if (!tf.predict(x0).isOk()) {
      Serial.println(tf.exception.toString());
      return;
  }

  Serial.print("expcted class 0, predicted class ");
  Serial.println(tf.classification);

  // classify class 1
  if (!tf.predict(x1).isOk()) {
      Serial.println(tf.exception.toString());
      return;
  }

  Serial.print("expcted class 1, predicted class ");
  Serial.println(tf.classification);

  // classify class 2
  if (!tf.predict(x2).isOk()) {
      Serial.println(tf.exception.toString());
      return;
  }

  Serial.print("expcted class 2, predicted class ");
  Serial.println(tf.classification);

  // how long does it take to run a single prediction?
  Serial.print("It takes ");
  Serial.print(tf.benchmark.microseconds());
  Serial.println("us for a single prediction");

  delay(1000);
}
```

- Arduino project dapat diakses di: https://github.com/as-budi/Embedded_AI/tree/main/09_Model_Deployment_on_Embedded_Systems/09_codes/IrisExample