

PENERAPAN MACHINE LEARNING PADA ESP32 MENGGUNAKAN TINYML

Ifadah Aulia Muhti Sinaga
233140707111034

ifadahauliasinaga@gmail.com

ABSTRAK

Praktikum ini bertujuan untuk mengimplementasikan algoritma Machine Learning pada perangkat mikrokontroler ESP32 dengan pendekatan TinyML. Model klasifikasi bunga iris digunakan sebagai studi kasus, dengan empat fitur sebagai input utama. Proses pelatihan dilakukan menggunakan TensorFlow, kemudian model dikonversi ke dalam format .tflite agar dapat dijalankan secara lokal pada ESP32 tanpa koneksi internet. Pengujian menunjukkan bahwa ESP32 mampu menjalankan model dengan waktu inferensi yang sangat singkat dan tingkat akurasi mencapai 100% terhadap data uji.

Kata Kunci: Machine Learning, TinyML, ESP32, TensorFlow Lite, Iris Dataset

ABSTRACT

This practicum aims to implement a Machine Learning algorithm on an ESP32 microcontroller using the TinyML approach. The Iris flower classification model is used as a case study, with four primary input features. The model is trained using TensorFlow and then converted into a .tflite format to run locally on the ESP32 without an internet connection. Test results demonstrate that the ESP32 can perform inference with extremely low latency and achieve 100% accuracy on the test data.

Keywords: Machine Learning, TinyML, ESP32, TensorFlow Lite, Iris Dataset

1. PENDAHULUAN

1.1 Latar Belakang

Internet of Things (IoT) telah menghadirkan berbagai peluang baru dalam pengolahan data secara real-time di perangkat berbasis mikro. Salah satu kemajuan signifikan adalah hadirnya TinyML, yang memungkinkan model Machine Learning berukuran kecil dijalankan langsung di mikrokontroler seperti ESP32. Dengan pendekatan ini, pengambilan keputusan dapat dilakukan secara cepat tanpa ketergantungan pada cloud atau server eksternal, sehingga sangat mendukung efisiensi dan mobilitas sistem pintar.

1.2 Tujuan Praktikum

- 1) Tujuan dari praktikum ini meliputi:
- 2) Menerapkan model klasifikasi Machine Learning menggunakan dataset Iris.
- 3) Mengonversi model TensorFlow ke dalam format .tflite yang kompatibel untuk perangkat mikro.
- 4) Mengimplementasikan dan menjalankan model di ESP32 menggunakan pendekatan TinyML.
- 5) Mengevaluasi performa model saat dijalankan pada perangkat mikrokontroler.

2. METODOLOGI

2.1 Alat dan Bahan

- **Alat:**
 - Laptop atau komputer
 - Wokwi Simulator
 - Google Colab
 - Arduino IDE atau PlatformIO
- **Bahan:**
 - a) ESP32 DevKit V1
 - b) File model Machine Learning dalam format .tflite
 - c) Library Arduino: eloquent_tinyml, tfml_esp32
 - d) Dataset Iris (data latih dan uji dari TensorFlow)
 - e) Python serta library pendukung: TensorFlow, NumPy, Pandas, Scikit-learn

2.2 Langkah Implementasi

Berikut adalah langkah-langkah implementasi secara rinci dalam melakukan simulasi sistem klasifikasi bunga iris berbasis Machine Learning menggunakan ESP32 dan TinyML:

1. Pelatihan dan Konversi Model Machine Learning

- Menggunakan Google Colab atau Jupyter Notebook untuk pelatihan model.
- Dataset Iris dimuat menggunakan pandas atau sklearn.datasets.
- Dilakukan preprocessing seperti normalisasi dan one-hot encoding.
- Model klasifikasi dibangun dan dilatih dengan TensorFlow.
- Setelah pelatihan, model dikonversi ke format .tflite menggunakan TFLiteConverter.
- File .tflite diubah ke format array C++ (.h) agar dapat digunakan di Arduino IDE.

2. Implementasi di ESP32 melalui VSCode

- Membuat proyek baru di VSCode menggunakan PlatformIO atau Arduino IDE dengan board ESP32 DevKit V1.
- File iris_model.h dimasukkan ke dalam proyek sebagai representasi model.

- Kode program ditulis dengan library `eloquent_tinymml` untuk melakukan inferensi.
- Program diunggah ke ESP32 dan hasil klasifikasi diamati melalui Serial Monitor.
- Input nilai `x0`, `x1`, `x2`, dan `x3` untuk melakukan pengujian terhadap ketiga kelas spesies iris.

3. Pengujian dan Evaluasi Sistem

- Mengukur kecepatan inferensi dengan fungsi `tf.benchmark.microseconds()`.
- Memvalidasi hasil klasifikasi terhadap label aktual untuk mengukur akurasi.
- Mengevaluasi stabilitas sistem melalui pengujian berulang dan pemantauan kemungkinan crash.
- Menilai konsistensi hasil prediksi selama periode waktu tertentu.

3. HASIL DAN PEMBAHASAN

3.1 Hasil kelas data IRIS yang diuji

