Nama: Rizki Hidayat

NIM: 1103202131

Laporan

1.Pendahuluan

Tujuan dari modul ini ini adalah untuk memperkenalkan pendekatan modular dalam pengembangan model PyTorch untuk klasifikasi gambar. Pendekatan modular memisahkan kode ke dalam modul-modul terpisah, memudahkan pemeliharaan, pemahaman, dan pengembangan lebih lanjut

2.Penjelasan

2.1. Import Libraries dan Pendefinisian Path Data

- Import library PyTorch dan modul-modul yang diperlukan.
- Tentukan path data dan path gambar.
- Download dan unzip dataset (pizza, steak, sushi) jika belum ada.

2.2. Data Preparation (data_setup.py)

- Tentukan transformasi data menggunakan torchvision.transforms.
- Gunakan data_setup.create_dataloaders untuk membuat DataLoader untuk pelatihan dan pengujian.
- Dataloader digunakan untuk memuat data secara efisien selama pelatihan dan pengujian.

2.3. Model Architecture (model_builder.py)

- Gunakan model_builder.TinyVGG untuk membuat arsitektur TinyVGG.
- TinyVGG adalah model Convolutional Neural Network (CNN) sederhana dengan dua blok konvolusi diikuti oleh lapisan klasifikasi.

2.4. Training dan Testing Engine (engine.py)

- Buat fungsi train_step untuk satu epoch pelatihan, yang melibatkan langkah-langkah forward pass, perhitungan loss, dan optimizer step.
- Buat fungsi test_step untuk satu epoch pengujian, yang melibatkan forward pass dan perhitungan loss.
- Buat fungsi train untuk menggabungkan pelatihan dan pengujian selama beberapa epoch.
- Selama pelatihan, cetak dan catat metrik pelatihan dan pengujian seperti loss dan akurasi.

2.5. Menyimpan Model (utils.py)

- Buat fungsi save_model untuk menyimpan model PyTorch ke direktori target.
- Gunakan torch.save untuk menyimpan state_dict() model.

2.6. Implementasi Pelatihan dan Simpan Model (train.py)

- Tentukan hyperparameter seperti jumlah epoch, ukuran batch, unit tersembunyi, dan laju pembelajaran.
- Tentukan path untuk data pelatihan dan pengujian.
- Pilih perangkat target (cuda atau cpu).
- Buat transformasi data menggunakan torchvision.transforms.
- Buat DataLoader untuk pelatihan dan pengujian menggunakan data_setup.create_dataloaders.
- Inisialisasikan model TinyVGG menggunakan model_builder.TinyVGG.
- Tentukan fungsi loss (CrossEntropyLoss) dan optimizer (Adam).
- Mulai pelatihan menggunakan engine.train.
- Simpan model yang telah dilatih menggunakan utils.save_model