

Nama : Eko Putra Nugraha

NIM : 1103213212

### **1. Struktur dan Organisasi Kode**

Kode ini dirancang dengan sangat terorganisir. Ada penggunaan modul PyTorch Lightning untuk menyederhanakan pengelolaan model, pelatihan, dan evaluasi. Framework ini membantu mengabstraksi proses seperti loop pelatihan, validasi, dan pengujian, sehingga kode menjadi lebih modular dan mudah dikelola. Selain itu, library tambahan seperti matplotlib, seaborn, dan torchvision digunakan untuk visualisasi data dan transformasi dataset.

### **2. Pengolahan Dataset**

Dataset CIFAR-10 digunakan sebagai basis pelatihan. Dataset ini dinormalisasi menggunakan mean dan standar deviasi yang dihitung dari data latih. Normalisasi ini penting untuk meningkatkan efisiensi pelatihan model. Selain itu, transformasi data seperti horizontal flipping dan random resizing digunakan untuk data augmentation, yang bertujuan mengurangi overfitting dan meningkatkan kemampuan generalisasi model.

### **3. Implementasi PyTorch Lightning**

PyTorch Lightning digunakan untuk mendefinisikan pipeline pelatihan melalui kelas CIFARModule. Modul ini mengorganisir komponen utama seperti:

- Inisialisasi model dan fungsi loss (CrossEntropyLoss).
- Konfigurasi optimizer (dengan dukungan untuk Adam dan SGD) dan scheduler untuk penyesuaian learning rate.
- Definisi langkah pelatihan, validasi, dan pengujian.

Kode ini juga menggunakan callback seperti LearningRateMonitor untuk memantau learning rate dan ModelCheckpoint untuk menyimpan model terbaik berdasarkan akurasi validasi.

### **4. Modularitas Model**

Pendekatan modular digunakan untuk mendukung berbagai arsitektur model seperti ResNet, DenseNet, dan GoogleNet. Sebuah dictionary (model\_dict) digunakan untuk memetakan nama model ke implementasi aktualnya. Hal ini memungkinkan fleksibilitas untuk menjalankan berbagai model dengan pipeline yang sama.

### **5. Peningkatan Generalisasi**

Strategi seperti data augmentation, penggunaan transformasi normalisasi, dan mekanisme checkpointing mendukung pelatihan yang lebih stabil dan generalisasi yang lebih baik. Selain itu, penyesuaian hyperparameter seperti learning rate dan weight decay dapat dilakukan dengan mudah.

Secara keseluruhan, kode ini adalah implementasi yang efisien dan fleksibel untuk eksperimen pembelajaran mendalam pada CIFAR-10. Struktur modular dan penggunaan framework PyTorch Lightning memberikan fleksibilitas dan kemudahan dalam menambahkan model atau fitur baru.