CIFAR-10 Image Classification

Objective

Build and train deep learning models using TensorFlow or PyTorch. Apply best practices in data handling, model architecture, training loops, and evaluation.

Part 1 — Core Coding Task

Build and train a neural network to classify images from the CIFAR-10 dataset.

Requirements

1. **Data Pipeline**
    - Load CIFAR-10 dataset
    - Apply standard preprocessing (normalization, augmentations, batching)
    - Demonstrate efficient data loading (tf.data or torch.utils.data.DataLoader)
2. **Model Architecture**
    - Implement a small CNN **from scratch** (not a pre-trained model), including:
        - Convolutional layers
        - BatchNorm
        - Dropout
        - A final classification head
3. **Training**
    - Implement a clear training loop:
        - Loss computation
        - Accuracy metrics
        - Learning rate scheduling
    - Save checkpoints
4. **Evaluation**
    - Plot loss & accuracy curves
    - Compute test accuracy
    - Show sample misclassified images
5. **Reproducibility**
    - Set random seeds
    - Provide environment/requirements file

Deliverables

- Source code (Python scripts or notebook)

- README explaining:
  - Project structure
  - Training setup
  - Model design decisions
  - How to run the code

Part 2 — Model Improvement Challenge

After the baseline model is working, implement **one** of the following improvements:

Option A — Add a Residual Block
Create a simplified version of a ResNet block and integrate it into your CNN.

Option B — Add a Custom Callback / Training Utility
Example:
- TensorBoard (TF)
- Custom PyTorch callback or logger
- Early stopping
- Gradient clipping

Option C — Hyperparameter Tuning
Perform tuning on:
- Learning rate
- Batch size
- Optimizer
- Dropout rate

Provide results comparing baseline vs improved model.

Optional Extensions
- Implement mixed-precision training
- Use distributed training (Multi-GPU / TPU / DDP)
- Deploy the trained model with FastAPI, Flask, or TensorFlow Serving
- Convert model to ONNX and perform inference
- Add unit tests for model components