CNN Class Assessment
Image Classification
March 19, 2025

==========Instructions==========

1. **Submission Format:** Two zip files with the name **CNN_CA_<RollNo>.zip** and **<Roll_no>_saved_models.zip** containing the directory structure as explained in **Submission Guidelines**.
2. Plagiarism will lead to **100% penalty** to all the involved parties. You are neither allowed to discuss among each other nor share code artifacts among each other.
3. Scoring will be automated script based, so maintain the proper file naming. Try to make the best-effort submission.
4. **Please** make sure to follow the correct file requirements in terms of command line arguments and proper file naming conventions.

==========

Objective

The goal of this assignment is to train a **CNN** model on the **CIFAR10 dataset** and explore model performance using different CNN architectures taught (AlexNet).

File 1 - <Roll_No>_train.py -> for Task 1 and 2

Task 1: Dataset Preparation

1. Load the CIFAR10 dataset and apply necessary **normalization and augmentation techniques** to improve model generalization.
2. Split the dataset into **training (80%)** and **testing (20%)** sets using **random seed as 10**.
3. Use the **torchvision.datasets** module to load the dataset and apply necessary **transformations**.

Task 2: Train AlexNet on CIFAR-10

1. Load a **AlexNet** model from **torchvision.models** and modify the final fully connected layer to classify **10 categories** in CIFAR10 Dataset. (**Note:- you may use a pretrained model or train from scratch**)
2. Fine-tune the model by training it on the dataset using:
    - **Adam optimizer** or **SGD with momentum**
    - **Cross-Entropy Loss**
    - Learning rate scheduling for better convergence
3. Implement **early stopping** to prevent overfitting.
4. Save the **trained models**. (alexnet.pth)

File 2 - <Roll_No>_evaluate.py -> for Task 3

Take the trained_model path and the name of the model as command line arguments.

Task 3: Model Evaluation

1. Evaluate the model using **accuracy, precision, recall, and F1-score**.
2. Generate a **confusion matrix** to analyze class-wise performance.
3. Submit a report <roll_no>_<model_name>_report.pdf containing the point number 2 and 3.

Submission Guidelines

- Submission Link - https://forms.gle/4vB5RcteRC937FuM7
- **Deadline - 19/03/2025 5:30pm**
- Submit two **Python scripts (.py)** with well-documented code.

Submission Directory Structure for **code**:-

CNN_CA_<Roll_no>
|
|---<roll>_train.py
|
|---<roll>_evaluate.py

Submission Directory Structure for **model**:-

<Roll_no>_saved_model
|
|---alexnet.pth
