ReadME File

Name: Priyanshu Ranka (NUID: 002305396)

Semester: Spring 2025

Subject / Professor: PRCV / Prof. Bruce Maxwell

OS and Software: Windows 11 / VS Code

3 of 8 time travel days were used for this project. 5 more days are left.

Project Overview:

This project outlines a comprehensive exploration of deep learning techniques applied to image recognition, specifically focusing on the MNIST digit dataset and transfer learning to classify Greek letters. It's structured into five main tasks, each building upon the previous one, to provide a thorough understanding of network construction, training, analysis, and adaptation.

Task 1 involves building and training a convolutional neural network (CNN) to recognize handwritten digits from the MNIST dataset. The code emphasizes good Python practices, including modular design with classes and functions, and saving the trained model for later use. This task serves as the foundation, ensuring a well-performing model is available for subsequent tasks.

Task 2 focuses on analyzing the trained network from Task 1. It involves visualizing the filters of the first convolutional layer to understand what features the network has learned. Additionally, it examines the effect of these filters on an example image, providing insights into the network's internal workings.

Task 3 introduces transfer learning, where the pre-trained MNIST model is adapted to classify Greek letters. This demonstrates the flexibility and reusability of neural networks. The task requires modifying the output layer of the pre-trained model and training it on a new dataset of Greek letter images, while also testing its performance on user-provided images.

Task 4 encourages experimentation and optimization. Students are tasked with designing and executing their own experiments to explore the impact of various network architecture and training parameters on performance. This involves automating the experimentation process and systematically evaluating different configurations to find the most efficient and effective model.

Overall, this project provides a practical and in-depth learning experience in deep learning, covering fundamental concepts like CNN architecture, training, evaluation, and transfer learning, while also emphasizing good coding practices and experimental design.

Folder Structure:

The ZIP file contains the following directories and files:

- **Code Files/**
  - Task_1_ABCD: Codes for accomplishing Task 1 – A, B, C, D parts.
  - Task_1_E: Codes for accomplishing Task 1-E
  - Task_1_F: Codes for accomplishing Task 1-F
  - Task_2: Codes for accomplishing Task 2
  - Task_3: Codes for accomplishing Task 3

    - Task_4: Codes for accomplishing Task 4
    - Task_4_Plotting: Code to plot Task 4
    - Extension_1: Codes for accomplishing Extension 1
    - Extension_2: Code for accomplishing Extension 2
    - Extension_2_Plotting: Code to plot Extension 2
- **CSV Files/**
    - Includes CSV files containing metadata, training logs, or extracted features.
    - fashion_mnist_experiment_results: Task 4 csv file
    - mnist_experiment_results: Extension csv file
- **Datasets/**
    - Contains the dataset used for training and evaluation, including MNIST or other relevant data.
    - data/ : MNIST dataset
    - dataFashion/ : MNIST Fashion dataset
    - Greek_train/ : Greek letter dataset
    - Paint_test_images/ : Digit written in Paint
    - Written_test_images/ : Handwritten digits
- **Models/**
    - Stores trained neural network models and checkpoints for future inference or fine-tuning.
- **Outputs/**
    - Contains results such as performance metrics, logs, and any visual outputs generated during model evaluation.
- **PRCV_Project_5_Report.pdf**
    - The final project report detailing the methodology, implementation, results, and conclusions.

Links:

Output Folder: Outputs

Models: Models

Requirements: Usage:

Libraries: Torch, Torchvision, Matplotlib, Numpy, Pandas and Opencv for python.

They can be directly installed from the command line using:
pip install torch torchvision matplotlib numpy pandas opencv-python

Usage:

The code files can be directly run. Make sure to extract all the .csv files in the same folder as the code files for Task 4 and Extension_2 to run it smoothly.