Weekly Report 1: Image Classification and Identification

Team Members:
Noshin Nawar – 2221507042
Md. Arebi Sarker – 2221362042
Mubashshira Kaisar – 2221070642
Md Zidan Khan – 2231413642
Date: 18 June, 2025

***Abstract*—In the first week of development, our team focused on laying the technical foundation for a modular and scalable image-based machine learning system. We implemented a unified command-line interface (CLI) and designed a folder structure to support the integration of multiple image processing tasks such as classification, detection, segmentation, and more. A detailed documentation file (`README.md`) was also created to support future users and contributors.**

I. Introduction

The project titled *Image Classification and Identification* aims to integrate a range of image processing and computer vision tasks—such as handwritten digit recognition, bone fracture detection, and document scanning—under a unified Python-based interface. A command-line approach was selected to ensure modularity and ease of testing across all tasks.

II. This Week's Progress

- Followed the required folder structure: `codeP/`, `codeB/`, `DATA/`, `RESULTS/`
- Created `main.py` to handle the tasks via the instructed style of command-line using `argparse`
- Implemented test-ready placeholder modules for each task.
- Verified CLI execution by printing messages.
- Developed `bash_command.sh` inside `codeB/` to automate CLI testing
- Used dummy test dataset (`exampledata/`) inside `DATA/` to simulate input handling
- Tested command structure in PowerShell and Bash successfully

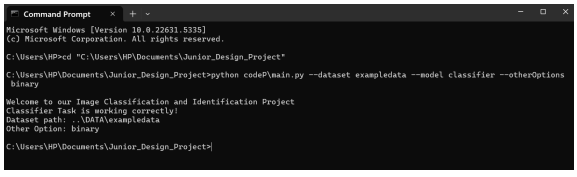

Fig. 1. Successful run of command

III. Challenges Faced

- Initially built the CLI without aligning to the instructor's expected folder layout and command syntax.
- • Managing consistent paths across different folders required structural reorganization.
- Ensured documentation and task setup was understandable for future contributors.

IV. Solutions and Decisions

- Adopted a consistent folder structure and modular approach for each task.
- Used placeholder task logic to allow early testing of CLI structure.
- Ensured documentation and task setup was understandable for future contributors.

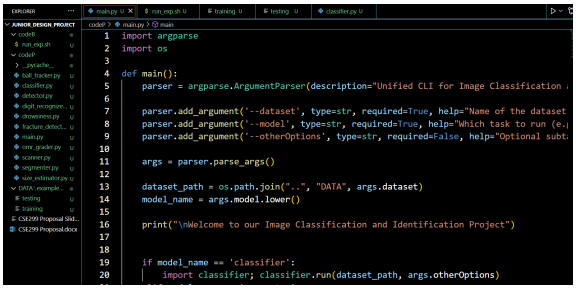

Fig. 2. Code of main.py

V. Next Week Plan

- Begin implementation of the tasks using a custom-trained model.
- Set up model loading/saving utilities.
- Prepare sample datasets and preprocessing pipeline for classification and detection tasks.
- Optionally implement logging or result-saving utilities for CLI outputs.