

# 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

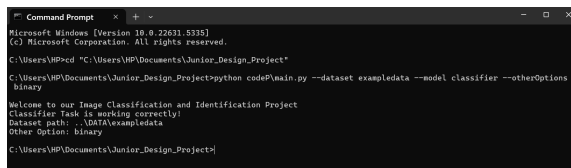
**Abstract**—In the first week of development, our team focused on laying the technical foundation for a modular and scalable image-based machine learning system. We implemented a unified command-line interface (CLI) and designed a folder structure to support the integration of multiple image processing 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



```
Microsoft Windows [Version 10.0.22631.5335]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HPD>cd "C:\Users\HPD\Documents\Junior_Design_Project"
C:\Users\HPD\Documents\Junior_Design_Project>python codeP/main.py --dataset exampledata --model classifier --otherOptions
binary
Welcome to our Image Classification and Identification Project
Classifier Task is working correctly!
Dataset path : DATA\exampledata
Other Option: binary
C:\Users\HPD\Documents\Junior_Design_Project>
```

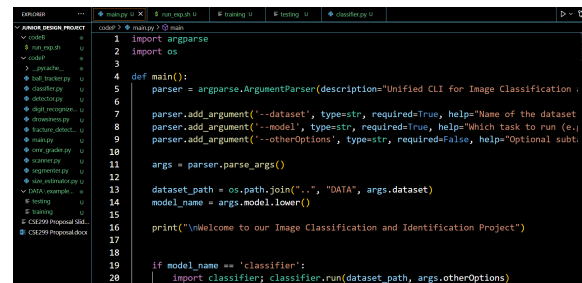
Fig. 1. Successful run of command

## 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.



```
1 import argparse
2 import os
3
4 def main():
5     parser = argparse.ArgumentParser(description='Unified CLI for Image Classification')
6
7     parser.add_argument('--dataset', type=str, required=True, help='Name of the dataset')
8     parser.add_argument('--model', type=str, required=True, help='Which task to run (e.g.)')
9     parser.add_argument('--otherOptions', type=str, required=False, help='Optional sub')
10
11     args = parser.parse_args()
12
13     dataset_path = os.path.join(".", "DATA", args.dataset)
14     model_name = args.model.lower()
15
16     print(f"Welcome to our Image Classification and Identification Project")
17
18
19     if model_name == 'classifier':
20         import classifier; classifier.run(dataset_path, args.otherOptions)
```

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.