National College of Ireland

Configuration Manual

MSc Research Project
MSCAI1B

Raj Nadigottu
Student ID: 23319658

School of Computing
National College of Ireland

Supervisor: Lavish Thomas

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Raj Nadigottu
Student ID:	X23319658
Programme:	MSCAI1B **Year:** 2024-25
Module:	Practicum
Lecturer:	Lavish Thomas
Submission Due Date:	11/08/2025
Project Title:	**Enhancing Plant Disease Classification through Unsupervised Domain Adaptation (UDA)**
Word Count:	560 **Page Count:** 3

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Raj Nadigottu

Date: 11/08/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

(Raj Nadigottu – x23319658)

---

1. Purpose

This document outlines the software environment, dependencies, and configurations required to replicate the experimental setup for the Unsupervised Domain Adaptation (UDA) for Plant Disease Classification project. It is based on the actual implementation across all provided code notebooks, from data preparation to final evaluation.

2. Hardware Requirements

- **CPU:** Quad-core or higher (Intel i5/i7 or AMD equivalent)
- **GPU:** NVIDIA GPU with at least 6GB VRAM (recommended for deep learning model training)
- **RAM:** Minimum 16GB
- **Storage:** At least 50GB free space for datasets, models, and outputs

3. Software Requirements

3.1 Operating System

- Windows 10/11 (tested)
- Linux Ubuntu 20.04+ (optional, for faster training)
- macOS (optional, but GPU acceleration limited)

3.2 Python Environment

- **Python Version:** 3.9.x
- Recommended to use **Anaconda** or **Miniconda** for environment management.

4. Dependencies

Install the following Python packages:

```
pip install tensorflow==2.12.0
pip install keras==2.12.0
pip install numpy==1.23.5
pip install pandas==1.5.3
pip install scikit-learn==1.2.2
pip install matplotlib==3.7.1
pip install seaborn==0.12.2
pip install opencv-python==4.7.0.72
pip install pillow==9.5.0
pip install tqdm==4.65.0
pip install jupyterlab==3.6.3
```

5. Dataset Configuration

5.1 Source Dataset

PlantVillage Dataset – used as the labeled source domain for baseline training and CycleGAN translation. Images stored in class-specific folders.

5.2 Target Dataset

PlantDoc Dataset – used as the unlabeled target domain for evaluation, SimCLR feature learning, and pseudo-labeling. Images stored in class-specific folders.

5.3 Folder Structure

```
dataset/
├── PlantVillage/
│   ├── Class1/
│   ├── Class2/
│   └── ...
├── PlantDoc/
│   ├── Class1/
│   ├── Class2/
│   └── ...
```

Before training, all images should be resized and normalized as per the preprocessing steps in Phase0_Data_Preparation_and_Cleaning.ipynb. This ensures consistency across models and prevents shape mismatch errors.

6. Notebook Execution Order

1. **Phase0_Data_Preparation_and_Cleaning.ipynb** - Load datasets, align classes, resize, normalize images.
2. **Phase1_Baseline_Classifier_Training.ipynb** - Train MobileNetV2 baseline on PlantVillage, evaluate on PlantDoc.
3. **Phase 2_ CycleGAN Training & Translation.ipynb** - Train CycleGAN and generate PlantDoc-style images.
4. **Phase_3_UDA_Training.ipynb** - Train classifier on original + translated images.
5. **Phase4_SimCLR_Pretraining_and_Finetuning.ipynb** - Pretrain SimCLR on Translated Plantvillage images, fine-tune classifier.
6. **Phase4b_SimCLR_Pretrained_Evaluation.ipynb** – Evaluate SimCLR-trained classifier.
7. **phase5.ipynb** – Apply pseudo-labeling with high-confidence predictions.
8. **phase6.ipynb** – Final training with all data sources (original, translated, pseudo-labeled).

7. GPU Configuration

If using GPU acceleration:

- Install **CUDA 11.8** and **cuDNN 8.6** (compatible with TensorFlow 2.12.0).
- Verify GPU detection:

```
import tensorflow as tf
print("Num GPUs Available:", len(tf.config.list_physical_devices('GPU')))
```

8. Random Seed Control

For reproducibility, it is recommended to set a consistent random seed in all notebooks before running experiments. The following snippet can be added at the start of each notebook:

```
import tensorflow as tf
import numpy as np
import random

SEED = 42
tf.random.set_seed(SEED)
np.random.seed(SEED)
random.seed(SEED)
```

To ensure complete reproducibility, it is recommended to include this snippet at the start of every notebook before loading data or training models.

9. Output Storage

Models are saved in .h5 format or SavedModel format depending on the training phase.
Saved outputs include:

- Baseline MobileNetV2 model
- CycleGAN generators
- SimCLR encoder
- Final UDA classifier

10. How to Run the Project

1. **Prepare Environment** - Install all dependencies and set up GPU drivers if applicable.
2. **Download Datasets** - Place PlantVillage and PlantDoc datasets in the folder structure described above.
3. **Run Phase 0** - Prepare datasets, align classes, resize, and normalize.
4. **Run Phase 1** - Train and evaluate baseline model.
5. **Run Phase 2** - Train CycleGAN and generate translated images.
6. **Run Phase 3** - Train classifier with original + CycleGAN images.
7. **Run Phase 4 & 4b** - Pretrain SimCLR, fine-tune classifier, evaluate.
8. **Run Phase 5** - Apply pseudo-labeling to PlantDoc images.
9. **Run Phase 6** - Final model training with all datasets combined.
10. **Save & Backup** – Store all .h5 models and evaluation results for reporting.

11. References

- TensorFlow Documentation: https://www.tensorflow.org/api_docs
- Keras API: https://keras.io/api/
- PlantVillage Dataset: https://github.com/spMohanty/PlantVillage-Dataset
- PlantDoc Dataset: https://github.com/pratikkayal/PlantDoc-Dataset

11. **Access to code + Dataset + whole Pipeline**
    https://drive.google.com/drive/folders/1n8-QDTLoLOJrRZnpNrIBQFjhoNtaHdCl?usp=sharing