National College of Ireland

Configuration Manual

MSc Research Project
Practicum Part 2

Nachiket Mehendale
Student ID: X23272473

School of Computing
National College of Ireland

Supervisor: Prof. Ade Fajemisin

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Nachiket Mehendale
Student ID:	X23272473
Programme:	Practicum 2
Year:	2024-25
Module:	MSc Research Project
Supervisor:	Prof. Ade Fajemisin
Submission Due Date:	11/08/2025
Project Title:	Configuration Manual
Word Count:	759
Page Count:	6

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:	
Date:	7th Aug 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

Nachiket Mehendale
X23272473

1 Introduction

This configuration manual provides step-by-step instructions for setting up and executing the Enhanced Crop Weed Detection project using an optimized Swin Transformer architecture. The manual covers environment setup, dataset configuration, code execution procedures, and result interpretation to enable complete reproduction of the research findings. The document ensures that researchers can replicate the experimental setup and validate the proposed optimization techniques. All necessary components including datasets, code structure, and execution guidelines are detailed for seamless project reproduction.

2 System Requirements

Platform: Kaggle Notebooks
Hardware: NVIDIA Tesla P100 GPU, 13GB RAM
Software: Python 3.11, PyTorch 2.0, CUDA 11.8

Required Libraries:

```
import torch
import torch.nn as nn
import torch.optim as optim
import torchvision
import numpy as np
from sklearn.metrics import precision_score, recall_score, f1_score
import os
import time
import gc
```

Most libraries are pre-installed in Kaggle. Install, only if required:

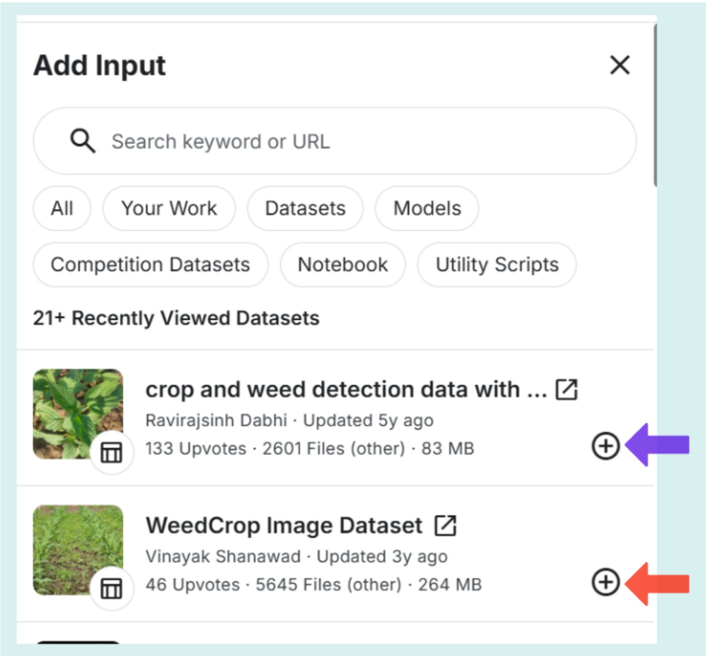
```
!pip install torch torchvision scikit-learn
```

3 Dataset Setup

3.1 Add Datasets to Kaggle Notebook

Make sure the following datasets are present in the executable Kaggle notebook and if not: Search for these datasets in the Add Input section of Kaggle Notebook and click on + symbols. This section is located on the top right corner in the Kaggle notebook.

1) **WeedCrop Image Dataset** --------*Primary dataset*
2) **crop and weed detection data with bounding boxes** -------*Secondary dataset*



Primary and secondary datasets can be found in Kaggle at following locations:

1) https://www.kaggle.com/datasets/vinayakshanawad/weedcrop-image-dataset
2) https://www.kaggle.com/datasets/ravirajsinh45/crop-and-weed-detection-data-with-bounding-boxes/data

Note: Once you have added them or if they are already present in the notebook, **+** symbol becomes **–**

3.2 Verify Dataset Paths

In the executable Kaggle notebook file, make sure the dataset paths are set as follows:

```
# Primary dataset - used in Dataset Setup and Conversion module
source_dir = '/kaggle/input/weedcrop-image-dataset/WeedCrop.v1i.yolov5pytorch'
# Secondary dataset - used in transfer learning module
transfer_path = '/kaggle/input/crop-and-weed-detection-data-with-bounding-boxes/agri_data/data'
```

4 Code Execution

4.1 Notebook Structure

The code implementation is organized and consists of key sections covering:

- Library imports and dependencies
- Dataset conversion and preprocessing
- Model architecture definitions
- Optimization techniques implementation
- Training and evaluation functions
- Transfer learning setup
- Main execution pipeline

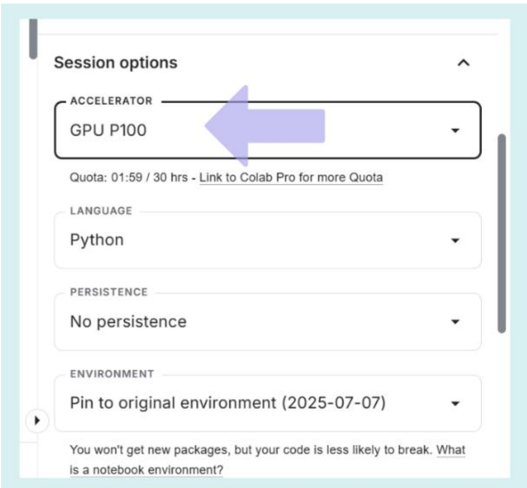
4.2 Execution Steps

- **Import Implementation:** The python implementation file is located at: https://github.com/Nachiket-AM/X23272473_MSCAI1_Research_Practicum_Part_2 Open/import it in Kaggle platform. Or you can directly access Kaggle notebook at: https://www.kaggle.com/code/cdnmehen/swin-transformer-optimization-x23272473-mscai1/edit or access it from OneDrive link as follows : x23272473_Swin_Transformer_Optimization.ipynb (Ctrl + Click to follow the link)

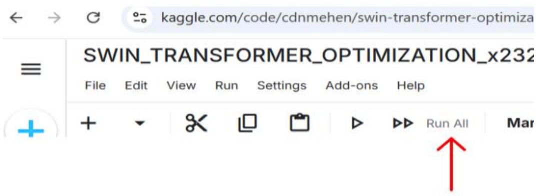
- **Dataset Configuration:** Make sure datasets are added, and path is defined in the executable Kaggle notebook file as mentioned in point no 3.1 and 3.2

- **Enable GPU:** In Kaggle notebook settings, select Accelerator → GPU P100

  (*This setting is present under the "Session options" section located on the right bottom side of the notebook*)



- Select "**Run All**" cells option from the executable Kaggle notebook



5 Model Architectures

In this project we have implemented 4 deep learning models:

Model	Description
Default Swin Transformer	Standard hierarchical Swin architecture
Optimized Swin Transformer	Enhanced version with novel optimization techniques
CNN Model	Traditional convolutional neural network baseline
Vision Transformer	Standard ViT implementation

Optimization Techniques in Optimized Swin:

- DTC: Dynamic Token Clustering - reduces computational complexity
- APS: Adaptive Patch Splitting - optimizes patch size selection
- SCSA: Selective Cross-Scale Attention - efficient multi-scale processing

6 Training Configuration

Hyperparameters (identical for all models):

- Epochs: 5
- Batch Size: 6
- Learning Rate: 0.001
- Optimizer: AdamW
- Loss: CrossEntropyLoss

7 Results

7.1 All Models Performance Comparison

All models achieved comparable accuracy on the test dataset:

Model	**Test Accuracy (%)**	**Precision**	**Recall**	**F1 Score**
Default Swin	82.29	0.823	0.823	0.823
Optimized Swin	83.33	0.833	0.833	0.833
CNN	81.25	0.813	0.812	0.812
Vision Transformer	83.33	0.836	0.833	0.833

7.2 Computational Efficiency Analysis (Default Vs Optimized SWIN)

Metric	**Default Swin**	**Optimized Swin**	**Improvement**
Parameters (M)	60.46	0.27	99.60%
Model Size (MB)	231.5	1	99.60%
Inference Time (ms)	10.4	22.1	-113.00%
FLOPs (G)	409.91	0.69	99.80%
Attention Efficiency	1	0	100.00%
Accuracy/Param (M)	1.36	311.32	22772.30%

The optimized Swin model achieved a 99.6% decrease in parameters or weights (from 60.46M to 0.27M) and model size (from 231.5MB to 1MB), while maintaining similar accuracy but with increased inference time from 10.4ms to 22.1ms. The efficiency gains resulted in a remarkable 22,772% improvement in accuracy per parameter, though at the cost of eliminated attention mechanisms and slower inference speed.

7.3 Transfer Learning Analysis

Secondary dataset performance:

Metric	**Value**
Final Transfer Training Accuracy	78.85%
Final Transfer Validation Accuracy	79.17%
Transfer Learning Epochs	5
Feature Extraction	Partially Frozen (head + last 2 stages trained)
Transfer Training Accuracy	78.85%
Transfer Validation Accuracy	79.17%

8 Conclusion

This configuration supports recreation of the optimized Swin Transformer research. The entire execution shows notable computational performance without losing crop-weed classification performance. The documented setup procedures and sequential execution approach enables reliable reproduction of the study results.