National College of Ireland

Configuration Manual

Practicum II
Msc Artificial Intelligence

Abdul Ahad Shaikh
Student ID: 233380756

School of Computing
National College of Ireland

Supervisor: Vikas Sahni

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Abdul Ahad Shaikh
Student ID:	23380756
Programme:	MSc AI
Year:	2025
Module:	Practicum II
Supervisor:	Vikas Sahni
Submission Due Date:	11/12/2025
Project Title:	Comparative Analysis of YOLOv8 and Vision Transformers for Road Damage Detection and Classification Using the RDD2022 Dataset
Word Count:	475
Page Count:	13

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:	Abdul Ahad Shaikh
Date:	9th December 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

Abdul Ahad Shaikh
23380756

1 Introduction

1.1 Purpose of the Configuration Manual

This configuration manual offers a clear guide for setting up and running the road damage detection system created for this research project. The system employs YOLOv8 for object detection and a Vision Transformer (ViT) for classifying image patches based on the RDD2022 dataset. This manual aims to ensure that all results can be reproduced by outlining the necessary environment, software installations, dataset preparation, model training process, and evaluation methods.

1.2 Scope of the Document

This manual includes:

- Installation of the required deep learning libraries
- Configuration of the project environment
- Preprocessing of the RDD2022 dataset, including annotation conversion and cleaning
- Training of YOLOv8 for bounding box detection
- Cropping detections and training of ViT for classification
- Evaluation setup for both models

The document does not discuss theoretical background, model architecture details, or in-depth research analysis. Those topics are addressed in other chapters of the dissertation.

1.3 Target Audience

This document is meant for:

- ML and AI researchers replicating the study
- Academic examiners assessing the project
- Practitioners implementing road inspection models
- Students learning about object detection workflows

1.4 System Overview

The system evaluates two different deep learning approaches:

- YOLOv8, a modern anchor-free object detector that identifies road damage and generates bounding boxes
- Vision Transformer (ViT), an attention-based classifier that sorts cropped damage regions into one of five classes in the RDD2022 dataset

The RDD2022 dataset is the main dataset used. It includes 47,420 images from six countries with annotations for:

- Alligator crack
- Longitudinal crack
- Transverse crack
- Pothole
- Other damage

Evaluation metrics consist of mAP@50, mAP@50-95, accuracy, recall, and confusion matrices.

2 System prerequisites

Certain hardware and software requirements must be met for implementation to be successful. These specifications guarantee effective training and reliable experiment replication.

2.1 Hardware Needs

The system was created on a non-GPU computer to mimic low-resource environments found in the real world.

Hardware Employed in this Project: Device: 2024 13-inch Apple MacBook Air Processor: Apple M3 CPU RAM (memory): 16 GB SSD with 256 GB of disk space

Figure 1: ***Basic Specification Requirements***

This hardware configuration was sufficient for:

- YOLOv8 CPU-based training (50 epochs)
- ViT training on cropped images (5 epochs)
- Dataset preprocessing and augmentation

2.2 Software Requirements

The following software and libraries are required to run the complete pipeline:
Operating System

- macOS Sonoma / Sequoia (Apple Silicon optimised)

Programming Environment

- Python 3.12.7

Core Libraries
Tools

- Jupyter Notebook
- VSCode (optional)
- Kaggle account (optional, for dataset download)

Library	Version	Purpose
PyTorch	2.9.1 (CPU)	Deep learning backend
Ultralytics YOLO	8.3.228	YOLOv8 training and inference
Transformers	4.x	Vision Transformer model
NumPy	1.26.x	Numerical processing
OpenCV	4.10.x	Image loading & resizing
Matplotlib	3.7.x	Visualisation
Seaborn	0.12.x	Confusion matrices
Pandas	2.2.x	Data handling

2.3 Network Requirements

An internet connection is required for:

- Downloading the RDD2022 dataset
- Pulling Ultralytics, PyTorch, or Transformers libraries
- Storing notebooks or results in cloud repositories (optional)

Once installed, all experiments can be run fully offline.

3 Installation Instructions

3.1 Prerequisites

Before installation, ensure the following: macOS 13+ installed, Python 3.12 installed, pip updated to latest version, and Sufficient disk space (20 GB recommended for dataset + results)

3.2 Installation Steps

Step 1: Create a virtual environment

```
python3 -m venv rdd_env
source rdd_env/bin/activate
```

Step 2: Install torch for Apple Silicon (CPU support)

```
pip install torch torchvision torchaudio
```

Step 3: Install Ultralytics YOLOv8

```
pip install ultralytics==8.3.228
```

Step 4: Install remaining libraries

```
pip install opencv-python pandas matplotlib seaborn transformers numpy
```

Step 5: Verify installations

```
import torch
import ultralytics
import cv2
import transformers
print("Environment ready!")
```

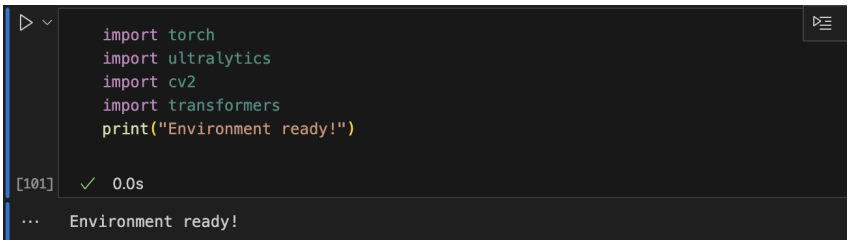

Figure 2: ***Ready to use Environment***

4 Dataset Setup and Preprocessing

4.1 Download RDD2022

The RDD2022 dataset was downloaded from the official Road Damage Dataset (RDD) repository. Link: https://www.kaggle.com/datasets/aliabdelmenam/rdd-2022

Figure 3: ***RDD2022 Dataset Folder Structure***

4.2 Convert Annotations to YOLO Format

RDD2022 provides annotations in Pascal VOC XML format. YOLO requires annotations in the format:

`class x_center y_center width height`

All values are normalised to the range $[0, 1]$.
A custom Python script was used to:

- Parse XML files
- Fix incorrect annotations (duplicate bounding boxes, negative coordinates)
- Output cleaned YOLO `.txt` label files

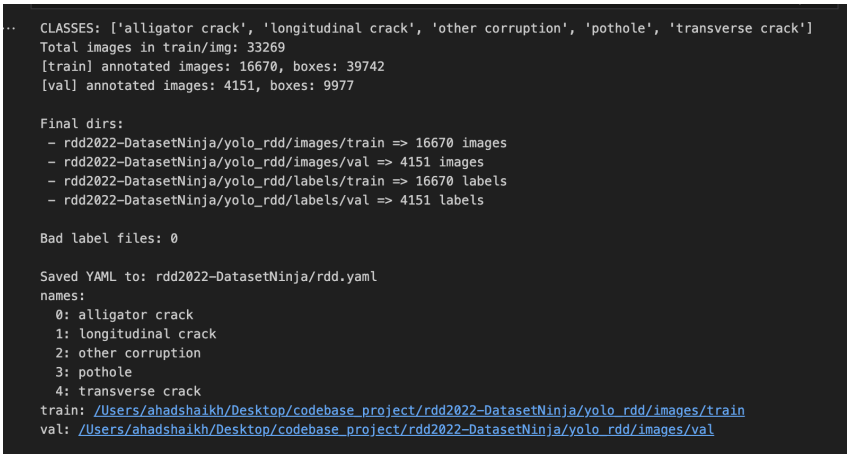

Figure 4: ***Annotation Conversion Script Output***

4.3 Build Final YOLO Dataset Structure

The final dataset directory was organised as follows:

```
yolo_rdd/
    images/train
    images/val
    labels/train
    labels/val
    rdd.yaml
```

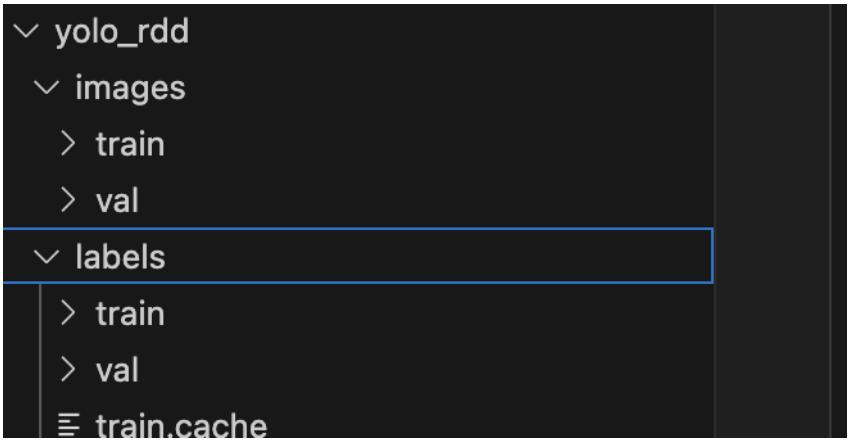

Figure 5: ***Built Final Dataset Structure***

5 Model Training

5.1 YOLOv8 Training

Training configuration

- Epochs: 50
- Image size: 640
- Batch size: 16
- Device: CPU
- Loss: Default YOLOv8 loss
- Optimiser: SGD

Run training:

```
from ultralytics import YOLO
model = YOLO("yolov8n.pt")
model.train(data="rdd.yaml", epochs=50, imgsz=640, device='cpu')
```

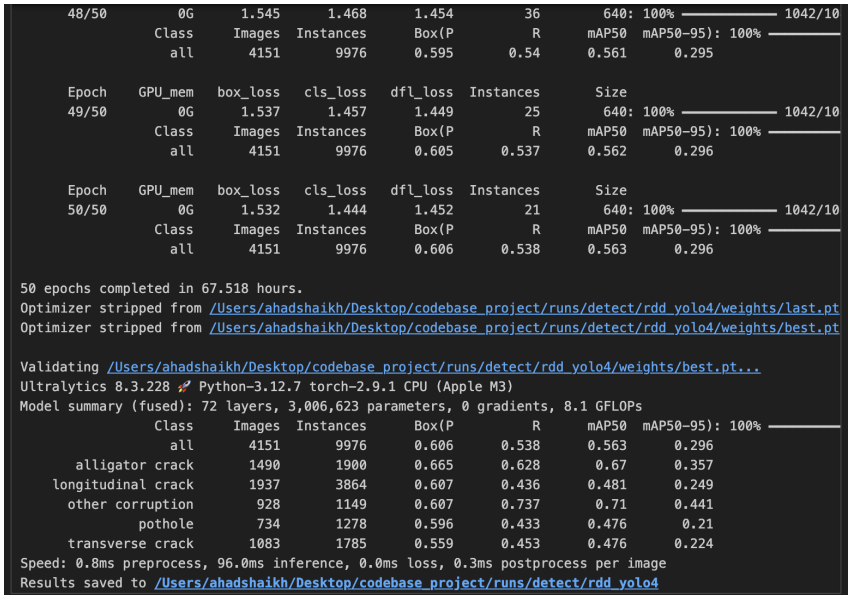

Figure 6: ***YOLOv8 Training Output Logs***

5.2 ViT Training

YOLO detections were used to generate cropped images with the following directory structure:

```
crops/
    alligator/
    longitudinal/
```

```
transverse/
pothole/
other/
```

Training configuration:

- Epochs: 5
- Optimiser: Adam
- Patch size: 16
- Input size: 224×224

```
Epoch 1/5
  Train loss: 0.9896, acc: 0.6137
  Val   loss: 0.8226, acc: 0.6819
  ✅ Saved new best model (val_acc=0.6819)

Epoch 2/5
  Train loss: 0.7727, acc: 0.7027
  Val   loss: 0.7714, acc: 0.7056
  ✅ Saved new best model (val_acc=0.7056)

Epoch 3/5
  Train loss: 0.6677, acc: 0.7435
  Val   loss: 0.7645, acc: 0.7054

Epoch 4/5
  Train loss: 0.5899, acc: 0.7771
  Val   loss: 0.7600, acc: 0.7148
  ✅ Saved new best model (val_acc=0.7148)

Epoch 5/5
  Train loss: 0.4835, acc: 0.8156
  Val   loss: 0.8626, acc: 0.6821
Best val acc: 0.714765906362545
```

Figure 7: ***ViT Training Output Logs***

6 Running the Models

After all steps are completed:

- YOLOv8 detects road damage and outputs bounding boxes.
- Cropped detections are classified using ViT.

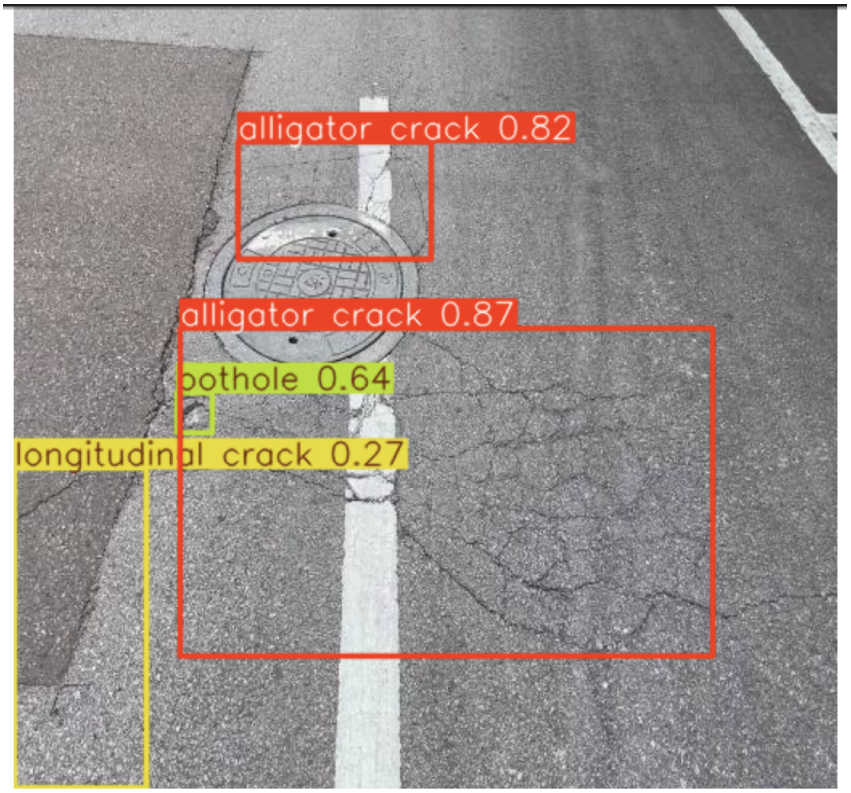

Figure 8: ***YOLOv8 Training Example***

Figure 9: ***ViT Training Example***

7 Results Overview

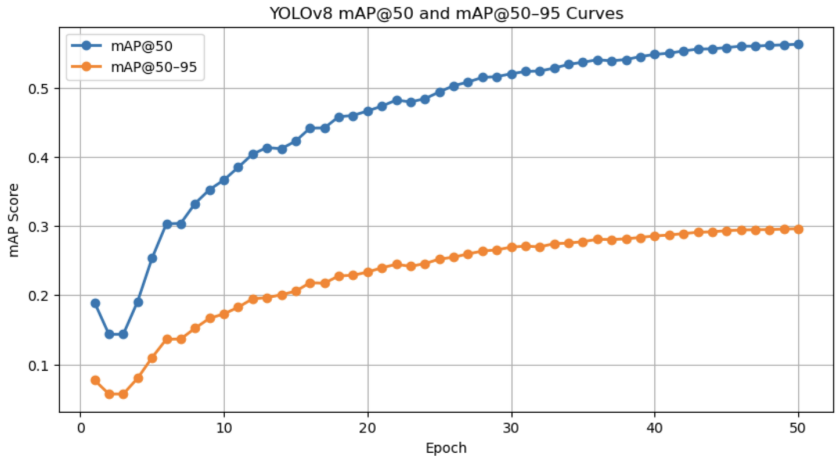

Figure 10: ***YOLOv8 mAP@50 amd mAP@50-96 Curves***

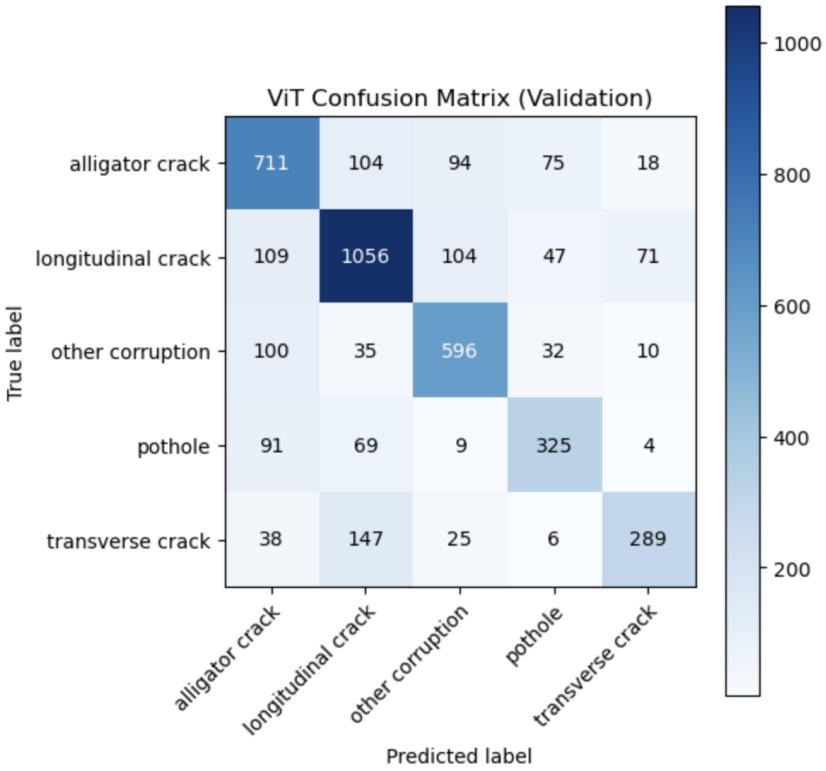

Figure 11: ***ViT Confusion Matrix***

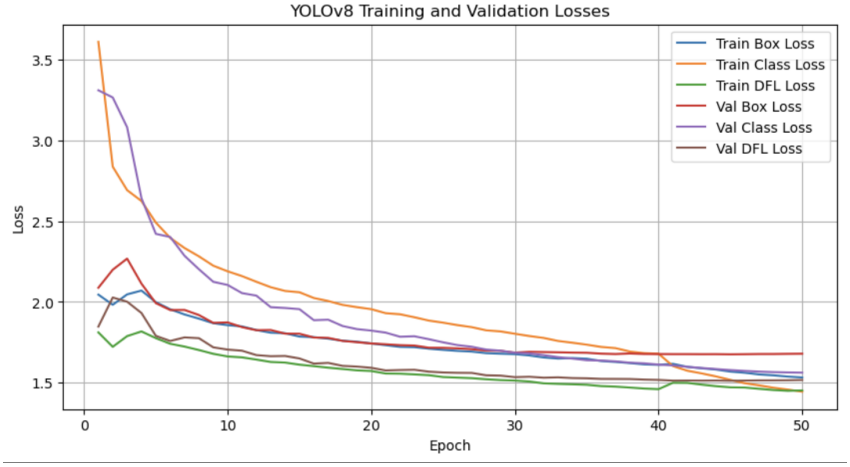

Figure 12: ***Training Validation losses***

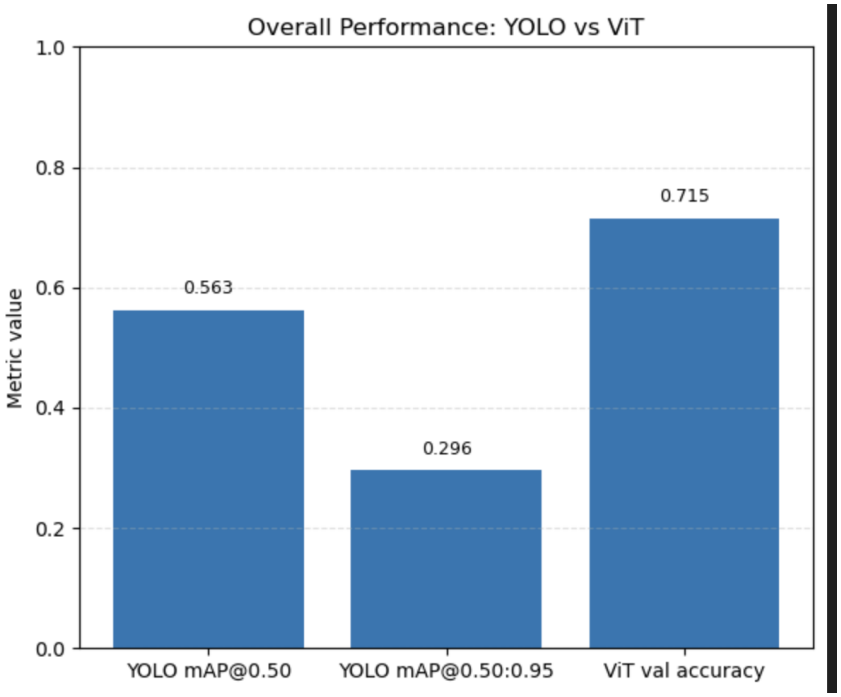

Figure 13: ***YOLOv8 vs ViT Accuracy***

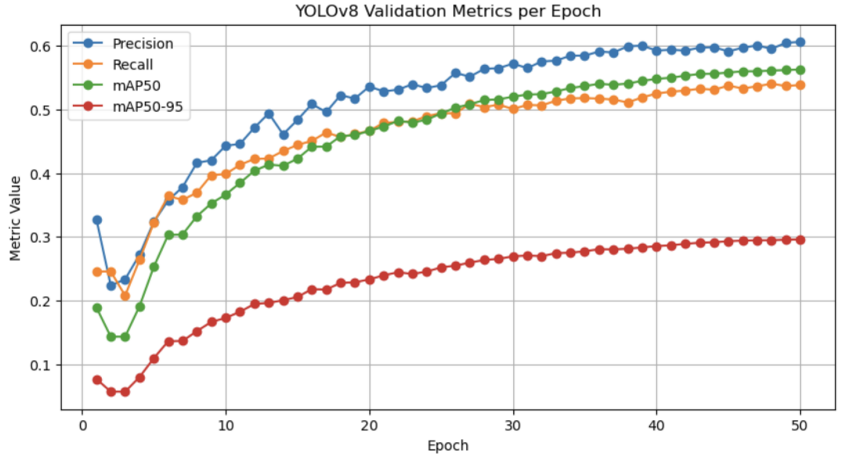

Figure 14: ***Validation Metrics Per Epoch***

8 References

Ultralytics YOLOv8 Ultralytics (2024). Ultralytics YOLOv8 Documentation. Available at: https://docs.ultralytics.com

PyTorch Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., et al. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. Advances in Neural Information Processing Systems (NeurIPS). https://pytorch.org

Vision Transformer (ViT) Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., et al. (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ICLR. https://arxiv.org/abs/2010.11929

Transformers Library (Hugging Face) Wolf, T., Debut, L., Sanh, V., Chaumond,

J., et al. (2020). Transformers: State-of-the-Art Natural Language Processing. EMNLP. https://arxiv.org/abs/1910.03771

RDD2022 Dataset Arya, D., Maeda, H., Ghosh, S., Toshiyuki, M., Oguchi, T. (2022). RDD2022: A Multi-Country Benchmark Dataset for Road Damage Detection. https://github.com/sekilab/RoadDamageDetector

Pascal VOC Annotation Format Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., Zisserman, A. (2010). The Pascal Visual Object Classes (VOC) Challenge. International Journal of Computer Vision. https://link.springer.com/article/10.1007/s11263-009-0275-4

OpenCV Bradski, G. (2000). The OpenCV Library. Dr. Dobb's Journal of Software Tools. https://opencv.org

Matplotlib Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. Computing in Science Engineering. https://matplotlib.org

Seaborn Waskom, M. (2021). Seaborn: Statistical Data Visualization. Journal of Open Source Software. https://seaborn.pydata.org

Pandas McKinney, W. (2010). Data Structures for Statistical Computing in Python. Proceedings of the 9th Python in Science Conference. https://pandas.pydata.org

NumPy Harris, C. R., Millman, K. J., van der Walt, S. J., et al. (2020). Array Programming with NumPy. Nature. https://numpy.org

YOLOv3–v5 Original Background Reference Redmon, J. Farhadi, A. (2018). YOLOv3: An Incremental Improvement. https://arxiv.org/abs/1804.02767

Attention Mechanisms Background (for ViT Theory Section) Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., et al. (2017). Attention Is All You Need. NeurIPS. https://arxiv.org/abs/1706.03762

Albumentations (Augmentation Library Reference) Buslaev, A., Iglovikov, V. I., Khvedchenya, E., Parinov, A., Druzhinin, M., Kalinin, A. (2020). Albumentations: Fast and Flexible Image Augmentations. Information. https://www.mdpi.com/2078-2489/11/2/125

Scikit-Learn Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research. https://scikit-learn.org

YOLOv8 State-of-the-Art Computer Vision Model. yolov8.com

Vision Transformers https://en.wikipedia.org/wiki/Vision$_t ransformer$