Identifying Cyclist Safety In Urban Traffic

Using Computer Vision

Joelma Rodrigues

A Report Submitted in Partial Fulfilment

of the requirements for the

Degree of

BSc in Computing in IT (4th year)

Diagram

Description automatically generated with medium confidence

May 2025

Supervisor: Dr. Muhammad Iqbal

Abstract

As traffic in urban centres grows and global warming debates spread, amid other factors, the use of eco-friendly transport has increased in the past few years. There is a need for infrastructure investment and road users' education regarding cyclists' rights to mitigate the risks of cycling. This project aims to build a computer vision model to automatically extract structured data from visual inputs, such as bicycles, scooters, high visibility vests and helmets, to support decisions to improve cyclists' experience and safety. The project offers a background analysis of different technologies and then discusses the development phases. The methodology adopted is the standard for data mining projects, CRISP-DM. The model was developed, evaluated and tested in six phases. The data was collected from open-source datasets and then tailored to the model's needs. The model was trained using YOLOV8 pre-trained weights and fine-tuned with different hyperparameter combinations, using augmentation and regularisation techniques to help the model generalise and avoid overfitting. Then, the chosen model was evaluated using the test subset, achieving a mAP@50 of 82.7%, with a satisfactory precision-recall balance. Followed by a cycle lane environment simulation with an inference of a recorded video. Despite the challenges, the project proves that computer vision can help existing cycle lane monitoring systems automatically collect relevant data to support decision-making to incentivise cycling as a means of transport.

**Keywords:** Intelligent Transportation Systems, Computer Vision, Object Detection, YOLO, CRISP-DM

Acknowledgement

I would like to express my gratitude to Professor David McQuinn for providing the foundation required to complete this project and to Dr Muhammad Iqbal, who helped me throughout the implementation of this capstone project, offering guidance and support.

I want to thank the friends I made during these four years of graduation. They made this journey even more enjoyable.

I am deeply thankful to my mom, Jeane Rodrigues and my brother, Augusto Rodrigues, for all their care and love. Their constant support reminded me of what truly mattered and uplifted me in the most challenging times.

I extend my gratitude to those who believed in my capabilities before I could recognise them and offered unwavering support throughout this journey, especially to Carraffat, Eddie, Steve, Caitriona, and Hugh.

Lastly, I would like to acknowledge the support from CCT College Dublin for providing resources and encouragement.

Contents

[1 Introduction 6](#_Toc198927205)

[1.1 Motivation 6](#_Toc198927206)

[1.2 Problem 6](#_Toc198927207)

[1.3 Objectives 6](#_Toc198927208)

[2 Literature Review 7](#_Toc198927209)

[2.1 Computer Vision: Trasport and Safety 7](#_Toc198927210)

[2.2 CNN and Object Detection 8](#_Toc198927211)

[2.3 You Only Look Once: Real-Time Object Detection 9](#_Toc198927212)

[2.4 Methodology 10](#_Toc198927213)

[3 Methodology (CRISP-DM) 12](#_Toc198927214)

[3.1 Business Understanding 12](#_Toc198927215)

[3.2 Data Understanding 13](#_Toc198927216)

[3.3 Data Preparation 14](#_Toc198927217)

[3.4 Ethical Considerations 17](#_Toc198927218)

[3.5 Modelling 18](#_Toc198927219)

[3.6 Evaluation 21](#_Toc198927220)

[3.7 Deployment 24](#_Toc198927221)

[4 Results 25](#_Toc198927222)

[5 Conclusion 26](#_Toc198927223)

[6 References 27](#_Toc198927224)

List of Figure

[Figure 1. Illustration of AI Fields 7](#_Toc198919157)

[Figure 2. Idea of a Convolutional Neural Network Architecture 8](#_Toc198919158)

[Figure 3. Functions: Classification, Object Detection, And Segmentation (Tedrake, 2024) 9](#_Toc198919159)

[Figure 4. YOLO Detection Model (Redmon et al., 2016) 10](#_Toc198919160)

[Figure 5. CRISP-DM Diagram 11](#_Toc198919161)

[Figure 6: Dataset Sample 13](#_Toc198919162)

[Figure 7. Pre-Selection of Images with LabelImg 14](#_Toc198919163)

[Figure 8. Data Splitting Ratio 15](#_Toc198919164)

[Figure 9. Classes Distribution 15](#_Toc198919165)

[Figure 10. YOLO Annotation Format 16](#_Toc198919166)

[Figure 11. Augumentation Techniques 17](#_Toc198919167)

[Figure 12. YOLOv8 Architecture (Yaseen, 2024) 19](#_Toc198919168)

[Figure 13. Intersection over Union (IoU ) Illustration (Padilla et al. 2021) 21](#_Toc198919169)

[Figure 14. Confusion Matrices 22](#_Toc198919170)

[Figure 15. Labels vs Prediction 23](#_Toc198919171)

[Figure 16. Cycle lane Monitoring System Simulation 24](#_Toc198919172)

[Figure 17. Details of the Simulation 24](#_Toc198919173)

[Figure 18. Inference of Test Subset 25](#_Toc198919174)

List of Tables

[Table 1. Hyperparameters Tested 20](#_Toc198922767)

[Table 2. Precision and Recall Results 22](#_Toc198922768)

[Table 3. Mean Average Precision at IoU 0.5 and 0.5 – 0.95 23](#_Toc198922769)

Chapter 1

# Introduction

* 1. Motivation

Around 41000 cyclists are killed yearly when commuting [World Health Organization (2022)](#Who2022); an individual not wearing a helmet at the time of a collision is more susceptible to severe head and brain injuries, while individuals wearing a helmet have the chance of a head injury reduced by up to 88%, according to the [World Health Organization (2022)](#Who2022). Despite the evidence of the efficacy of protective equipment while riding, its use is not compulsory in most countries.

Incentivising cycling as a means of transport is key to reducing traffic congestion in urban centres, promoting active life and reducing carbon emissions. However, attracting people to cycling requires infrastructure investment and road users' education regarding cyclists' rights to mitigate the risks of cycling.

* 1. Problem

Existing cycle lane monitoring systems can analyse cyclist trends, such as direction, at a determined time and location with the help of technologies such as radars and loop detectors [(Dilek and Dener, 2023)](#Sensors_ITS2023). In urban centres, computer vision technology is already used for multiple-mode monitoring systems to detect people, motorised vehicles, and cyclists from shared spaces. Although the technology is already available and used in similar scenarios, the use of computer vision in cycle lane monitoring systems to identify users' behaviour is yet to be explored.

* 1. Objectives

Motivated cyclists' safety concerns and lack of regulation issues, this project aims to train a computer vision model capable of monitoring cycle lane users' behaviours, identifying the types of cycle lane users (bicycle or e-scooter) and the use of protective equipment (helmet and high visibility vest). The model offers a method to automatically collect structured data to support decisions about infrastructure investments, campaigns to promote cyclist safety awareness and analysis of its effectiveness, and future introduction of helmet regulations.

Chapter 2

# Literature Review

* 1. Computer Vision: Trasport and Safety

Computer Vision (CV) is a field of Artificial Intelligence where machines *learn* to recognise, track or detect objects from visual input, such as images and videos, imitating human abilities with the help of Machine Learning (ML) and Deep Learning (DL) methods to apply in real-world scenarios [(Goodfellow, Bengio and Courville, 2016)](#MIT_Deep_Learning).

A diagram of a machine learning

AI-generated content may be incorrect.

Figure 1. Illustration of AI Fields

In traffic monitoring and safety, computer vision can be used in a range of scenarios, from detecting and classifying vehicles, accidents on the roads, obstacles, pedestrians and lane line detection or even for surveillance in so-called Intelligent Transportation Systems (ITS) that helps to automate tasks and provide data to improve decision making [(Dilek and Dener, 2023)](#Sensors_ITS2023).

The above-mentioned are usually applied in multi-modal environments when the system simultaneously analyses all road users: pedestrians, cyclists, and vehicles rather than monitoring cycle lanes alone. For example, [Breum, Kostic and Szell (2022)](#cyclists_copenhagen2022) studied cyclists' trajectory in the controversial bidirectional junctions designed for Dybbølsbro, Copenhagen, and analysed road safety implementation's effectiveness using object tracking. In another example, [Thompson, Lowry and Abdel-Rahim (2024)](#TransportationConsortium2024) suggested different approaches to display the potential of CV for traffic safety analysis by calculating the distance between a pedestrian and a vehicle at intersections to flag conflict zones and identify the potential for incidents.

This project applies computer vision to build a model and suggests how it can be used in cycle lane monitoring systems to extract relevant data. Sections [2.2](#cnn_obj_detec) and [2.3](#yolo2_3)briefly explain the concepts and architectures behind the proposed model. [Chapter 3](#CHAPTER3) summarises the development phases of a practical object detection model, and [Chapter 4](#Chapter4) discusses the research results.

* 1. CNN and Object Detection

Convolutional Neural Networks, or CNNs, are deep learning algorithms that use convolutional layers to learn the most important aspects of a visual input. This mathematical operation extracts a third feature from two other inputs, which is the characteristic that differs classical Machine Learning (ML) models from CNNs [(Goodfellow, Bengio and Courville, 2016)](#MIT_Deep_Learning).

The efficiency of a CNN is due to its hierarchical architecture. This solves the problem of depth that previous models' architectures encountered when dealing with variables of 2D shapes. A CNN is a stack of Convolutional layers, activation functions, pooling layers, and fully connected layers.

A diagram of a network

AI-generated content may be incorrect.

Figure 2. Idea of a Convolutional Neural Network Architecture

* **Convolutional Layer:** The most critical layer of a CNN, the first layer, uses kernels of n-dimension to slide through the image and automatically extract relevant feature maps (filters).
* **Activation function:** Introduces non-linear relationships among the features output in the previous layer.
  + An example of a function is the **Rectified Linear Unit** (ReLU), where . When, then ; and when , .
* **Pooling layer:** Reduces spatial dimension (reducing the number of parameters), preserving the most important features of the previous layer.
* **Fully Connected Layers:** combines the features generated on the previous layers (2-Dimensions)and transforms into 1-Dimension, a vector with the parameters used to complete the task.
  + Using weights and biases to make a prediction.

Initially used for image classification, CNNs can also perform object detection, image recognition and segmentation. [*Figure 2*](#imgs_annotations_MIT) explains the difference between these functions, using examples from COCO and ImageNet datasets.

A collage of sheep and cat

AI-generated content may be incorrect.

Figure 3. Functions: Classification, Object Detection, And Segmentation [(Tedrake, 2024)](#images_annotations_detect_seg_recog)

Conventional CNN architectures are performed efficiently in classification tasks, where the model assumes only one class to recognise in the image. Object detection tasks search different classes and their positions. [Girshick *et al.* (2014)](#R_CNN_paper) then proposed a solution for locating precisely the classes of an image with Region-Based CNNs. Further improving training speed and object detection accuracy with Fast R-CNN architecture. These methods operate in two steps: (1) region proposals and (2) classification, and then refine the bounding boxes [(Girshick, 2015)](#Fast_R_CNN_paper).

* 1. You Only Look Once: Real-Time Object Detection

Unlike its predecessors, the You Only Look Once (YOLO) approach to object detection uses a single convolutional network analysing the S x S grid rather than regions. YOLO is twice as fast as R-CNN; it quickly identifies objects and then trades off accuracy since the analysis grid fails to understand the full context of an image, misclassifying background with classes sometimes [(Redmon et al., 2016)](#YOLO_origin).

A diagram of a bicycle

AI-generated content may be incorrect.

Figure 4. YOLO Detection Model [(Redmon et al., 2016)](#YOLO_origin)

Classifier-based methods, although highly accurate, require high computational power and are slow. [Redmon et al., 2016](#YOLO_origin) proposed the YOLO method to address these issues related to real-time object detection, using 24 convolutional layers and two fully connected layers, using 1x1 convolutional layers to reduce dimensions, up to the output layer with 7 x 7 x 30 tensor of predictions. The YOLO model, therefore, became the most efficient method for real-time object detection. Updating the architecture over the years to improve performance.[*Chapter 3.4*](#modeling_34) details the architecture used to build the project's proposed model.

* 1. Methodology

Data mining methodologies are steps that guide data mining projects through the phases of data extraction, processing and modelling. Several methodologies, such as Knowledge Discovery in Databases (KDD), form nine steps from step one, learning the application domain, to the final using discovered knowledge [(Plotnikova, Dumas and Milani, 2020)](#crisp_dm2020). This methodology was the foundation for several of the newest data mining tools, such as the SEMMA and Cross Industry Standard Process for Data Minning (CRISP-DM). This project adopts the CRISP-DM methodology, which is ideal for data mining large datasets and comprises six phases.

* **Phase 1. Business Understanding:** Start by defining a clear objective and aligning the project's goals with the business to provide a structured plan to accomplish the objectives.
* **Phase 2. Data Understanding:** Data collection is initiated here and is explored to identify possible quality issues.
* **Phase 3. Data Preparation:** In the third step, the data from the last step is polished (clean, select, integrate) for use in the next step. This step is repeated as many times as necessary to tailor the data.
* **Phase 4.** **Modelling:** This step is experimental, where different techniques (architecture and algorithm) and parameters are employed until the best model is achieved.
* **Phase 5.** **Evaluation:** this step is where the result of the last step evaluates which model accomplishes the business objective.
* **Phase 6. Deployment:** The final step is when the model becomes available for the customer to try and see the results.

A diagram of a diagram of a data processing process

AI-generated content may be incorrect.

Figure 5. CRISP-DM Diagram

These six phases are iterative, where some phases allow return due to the results of the next step, which is why CRISP-DM is the data mining standard methodology. The application of these phases in the project is further implemented in [*Chapter 3*](#chapter_3).

Chapter 3

# Methodology (CRISP-DM)

* 1. Business Understanding

Utilising bicycles as the primary mode of transport has increased in the past few years. Along with the popularisation of electric bicycles and scooters in recent years. This brings risk factors to traditional bikes since these vehicles share the same lane.

Studies have proved the efficiency of the use of helmets. Moreover, statistics regarding cyclist mortality reduced significantly following the implementation of compulsory use regulations, as [Olivier, Boufous and Grzebieta (2019)](#HelmetRegulation2019) discuss.

### Business Objectives

Demonstrate computer vision methods to monitor cycle lanes. Distinguish vehicle types (bike or scooter) and monitor urban safety protective equipment (helmet or high visibility vest). The aim is to develop a model to automatically collect structured data, such as who the cycle lane users are and whether they use protective equipment.

### Business Success Criteria

The model will be successful when it can detect the interested classes: bicycle, helmet, scooter, and vest precisely.

* **Mean Average Precision** **mAP@50** ≥ 60%.

A metric commonly used to evaluate object detection models [(Lin et al., 2014)](#coco_microsoft). This means the model can detect objects with an IoU threshold of 50%. Evaluated in [*Section 3.5.*](#evaluation_35)

* 1. Data Understanding

### Data Collection

The images were collected from two open sources of computer vision datasets. The classes of interest to train the model are listed below in the same order.

1. [**Open Images Dataset V7**](#open_image)

Bicycle: 17631 images.

Helmet: 12614 images.

Open Image is a Google dataset that contains over 1.9 million images with 600 object classes annotated [(Open Images Dataset, 2022).](#open_image)

1. [**Roboflow**](#roboflow)

Scooter:14101 images.

Vests: 17761 images.

Roboflow is a platform used to train and deploy a CV model with over 750,000 datasets available [(Roboflow, 2020)](#roboflow).

The initial dataset contained 62,107 annotated images and approximately 12GB of data, including the labels.

A collage of different images of people

AI-generated content may be incorrect.

Figure 6: Dataset Sample

It is valid to mention that manually annotating the images was impractical due to the project's time limits. Therefore, the crowdsourced annotated (non-expert annotation) labels were checked using LabelImg [(Tzutalin, 2015).](#label_img) These steps allowed a previously deleting the images that were not relevant for the training phase.

A collage of a person riding a bike

AI-generated content may be incorrect.

Figure 7. Pre-Selection of Images with LabelImg

* 1. Data Preparation

This phase was done iteratively; multiple times, the modelling phase revealed problems on the dataset. The first model training revealed some data leakage, where some parts of the data collected were video frames. Present in both train and validation sets justified why scooter and helmet classes outperformed other classes.

On a second training attempt, the model had an average performance. However, the class bicycle still performed poorly, indicating that some data issues were still present. Further analysis found that there were still many images that were irrelevant to the model. For example, where there were only parts of the bicycle (chain, seat, frame).

From cleaning to splitting the dataset was executed manually to prevent any other possible issue caused by having the same image in more than one subset, ruining the training phase.

### Data Splitting

The final dataset contained a total of **40157 images**. And the ratio in which the dataset was split into an 80/10/10 ratio. Most images are intended for the model's training, where the model learns the most important characteristics of the relevant classes. At the same time, the validation set is a smaller subset that helps the model evaluate how it is doing during the training section on unseen images, and the test set serves the purpose of assessing the model [(Géron, 2019)](#hands_on_book). It is important to note that there is no rule to the data split ratio. A typical division ratio was applied.

A diagram of a product

AI-generated content may be incorrect.

Figure 8. Data Splitting Ratio

* **Train:** 32121 images.
* **Validation:** 4015 images.
* **Test:** 4021 images.

### Classes Distribution

Although there is a modest imbalance in the classes scooter and vest representation, initial modelling phases detected that the model learned these classes easier than the helmet because of the size of the object and a bicycle due to empty spaces between the frame and wheels and other parts (irregular geometry).

* **Bicycle:** 22771 instances.
* **Helmet:** 23666 instances.
* **Scooter:** 17701 instances.
* **Vest:** 18398 instances.

A graph of a number of classes

AI-generated content may be incorrect.

Figure 9. Classes Distribution

### Normalisation

The model's learning process is faster when the images are normalised. The pre-trained model used for this project automatically transforms these values using the min-max scale, which maintains the pixel values in a range of 0 and 1 by simply dividing them by 255, the max value of RGB images.

A close-up of numbers

AI-generated content may be incorrect.

Figure 10. YOLO Annotation Format

The YOLO label format starts with the index of the classes the model is being trained, such as 0 (bicycles), 1 (helmet), 2 (scooter), and 3 (vest), a list that goes on the data.yaml file to guide the model when classifying. The other four numbers are the centre of the bounding box and its height and width, which is how the model locates the objects of an image. The model can be trained with different resolutions. These values are usually divided by the image dimension to get a normalised value.

### Data Augmentation

The augmentation techniques helped the model avoid overfitting by generalising the dataset during the training. The pre-trained model used for this project allows online augmentation [(Ultralytics, 2024)](#ultratytics_v8). Offline augmentation is done before the training and generates augmented versions of the original image (Buslaev et al., 2020). Considering the size of the training set, around 30,000 images, it would require considerable space in the memory. Specific examples of the augmentations applied during the training are:

* **Resize:** Resize images before training to standardise at 640x640.
* **Horizontal Flip:** The images are flipped horizontally, never vertically, not to force unrealistic scenarios, as there are no bicycles upside down.
* **Translation:** moves the images left or right up or down. This moves the objects, so there is no fixed location for the objects.
* **HSV:** This technique changes Hue, Saturation, and Value to simulate illumination scenarios.
* **Coarse Dropout:** erases a square randomly to simulate blocking objects.

A collage of several people riding bicycles

AI-generated content may be incorrect.

Figure 11. Augumentation Techniques

* 1. Ethical Considerations

The data collected was used for academic purposes only. To ensure the ethical and legal use of data, the license was verified and confirmed to be under the Creative Commons Attribution 4.0 [(Creative Commons, 2013)](#creative_commons) license, which means the data can be used freely with attribution.

* 1. Modelling

### Transfer Learning

The YOLO architecture was discussed earlier in [*Chapter 2.2.*](#yolo2_3) Fast forward, Ultralytics has released a new version of YOLO nearly every year, changing some aspects of the architecture to improve accuracy-speed trade-off. This project employs YOLOv8, an established version first introduced by [Jocher, Chaurasia and Qiu (2023)](#yolov8). The version adopted is the nano, on detection tasks, achieves a mAP@50-95 ≥ 37.3% on the COCO dataset with 3.2 million parameters [(Ultralytics, 2024)](#ultratytics_v8).

### Fine Tuning

The models were trained under a supervised learning approach with a customised dataset containing four classes: bicycle, helmet, scooter and vest. The dataset is customised to the YOLO format. The first training session was executed on a local machine (RTX 2060, 64GB RAM, 6GB VRAM). Due to time restrictions, the following models were trained on a cloud instance (RTX A5000, 62GB RAM, 20GB VRAM, 24 vCPU). This reduced the training time by approximately 68%.

### Model Architecture

The most important aspects of the YOLOv8 architecture are divided into three parts:

1. **Backbone,** where the primary feature extraction occurs. The blocks Cross-Stage Partial with Fusion (C2f). It reduces the parameters and maintains its feature extraction capacity. It is a mix of dense and transition layers that contribute to lowering computing power [(Wang et al., 2019)](#csp_net).

Additionally, it separates the channels where a part is later concatenated with the other channel coming out of the Bottlenecks blocks. At the end of the backbone, there is a Spatial Pyramid Pooling Fast (SPPF) block, which, instead of a simple pooling layer, concatenates three different kernel sizes max-pooling layers to maintain the general aspects of the image [(He et al., 2015)](#SPPF).

1. **Neck**, a Feature Pyramid Network (FPN) increases the dimension of the feature maps so the FPN can use deep maps to detect small objects [(Lin et al., 2017).](#FPN) Meanwhile, the Path Aggregation Network (PAN) collects features from the top of the pyramid and concatenates important features, such as edges and textures [(Liu et al., 2018)](#PAN).
2. **Head**, anchor-free predictions are based on the feature maps extracted from previous layers. Each detection block considers different sizes of objects respecting the Complete Intersection Over Union Loss (CIoU Loss) to locate the box precisely and Binary Cross-Entropy Loss (BCE Loss) to the define the object's class.

The convolutional layers apply the Sigmoid Linear Unit (SiLU) as an activation function [(Ultralytics, 2024)](#ultratytics_v8). The detection block predicts each bounding box's objects and each class's probability [(Ramachandran, Zoph and Le, 2017)](#activation_function).

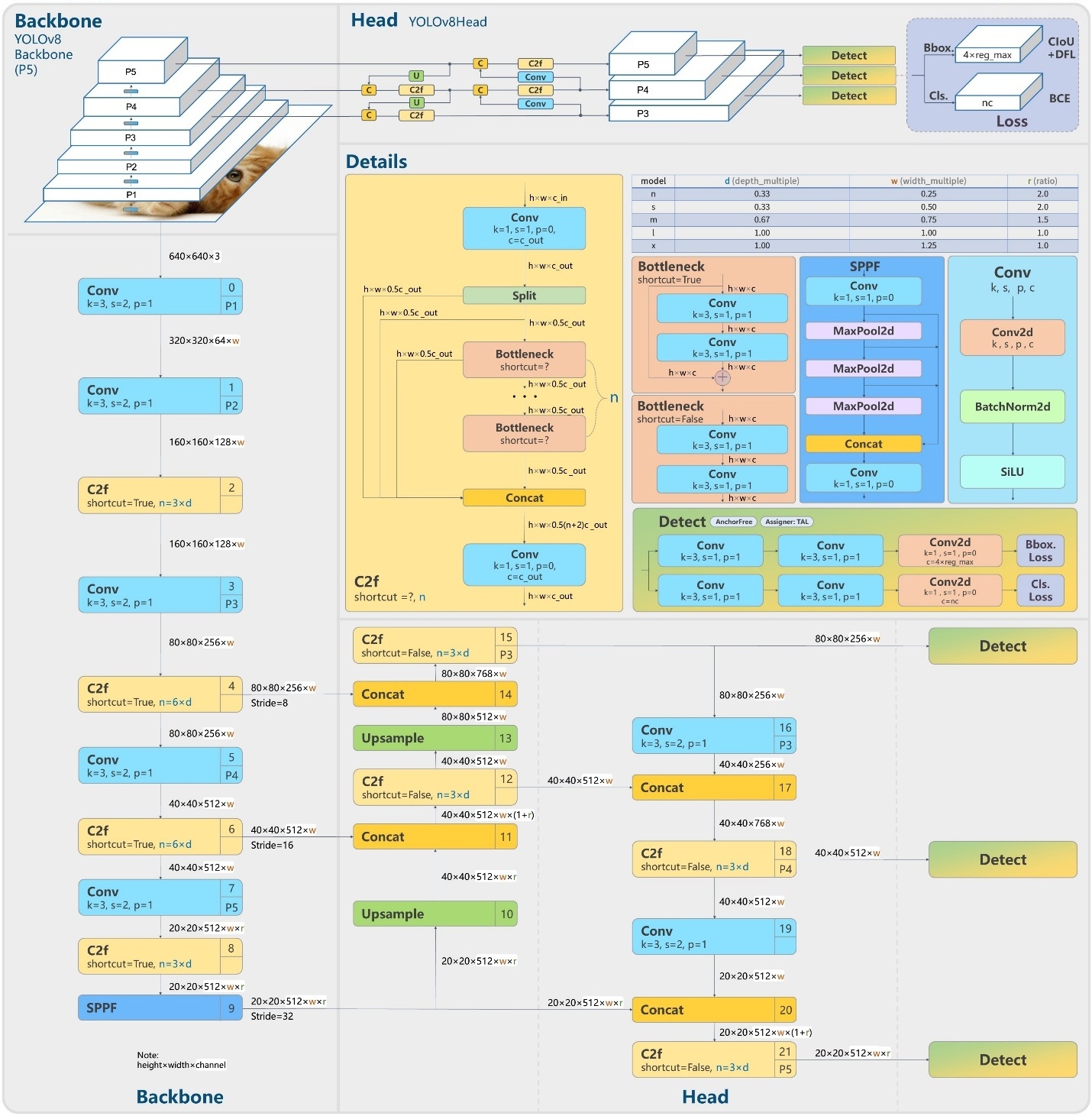


Figure 12. YOLOv8 Architecture [(Yaseen, 2024)](#yolov8_arch)

### Hyperparameters

As previously discussed, several models were trained initially, revealing inadequate data preparation, see [*Section 3.3*](#data_preparation). After these issues were addressed, a few other models were trained to test a selection of hyperparameters, using regularisation techniques to prevent overfitting and help the model generalise.

Early stopping is used to avoid wasting resources and overfitting. In this case, if the mAP or the validation loss had no change for 10 epochs, the model stopped training and saved the best model [(Géron, 2019)](#hands_on_book). Weight decay was kept at 0.0005 to penalise big values and a dropout rate of 0.05 (5%) to avoid error when generalising, so the model does not rely too much on specific units [(Goodfellow, Bengio and Courville, 2016)](#MIT_Deep_Learning).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Optimizer | Epochs | Batch | Learning Rate | Box Loss | Classification Loss | Distribution Focal Loss | mAP@50 |
| Adam | 50  **100** | 32  **64** | 0.01  **0.001** | 7.0  **8.5** | **0.3**  7.5 | **1.5**  2.0 | 81,1% |
| SGD | 50/100  **150**  200 | 64  **128** | 0.1 / **0.01** 0.001  0.0001 | **7.5**  8.5 | 0.5  **0.8**  0.9 | 1.5 | 82,1% |

Table 1. Hyperparameters Tested

The initial training was adjusted considering the pre-trained model hyperparameters, which resulted in a mAP@50 of 62% with the customised dataset. Therefore, changing the learning rate and epochs helped to achieve better results. However, using the Stochastic Gradient Descent (SGD) optimiser required longer convergence, and the model would enter a Plateau.

Further changes were attempted to get the model to learn the classes bicycle and helmet better. For example, increasing the classification weight from 0.5 to 0.8 improved the model's ability to classify these classes correctly [(Géron, 2019)](#hands_on_book). Consequently, it enhanced the overall performance.

The final attempt to adjust the hyperparameters was using Optuna, a hyperparameter optimiser framework proposed by [Akiba et al. (2019)](#Optuna). The suggested hyperparameters implemented the Adaptative Moment Estimation (Adam) optimiser with a small learning rate (0.0006 ). The model converged faster but achieved slightly lower results ( 81,1%), and the issues with those individual classes were not addressed, so the model was abandoned.

The minor differences between the models lead to better performance. The final choice was the model trained with SGD optimiser, a learning rate of 0.01, momentum of 0.936, 150 epochs, 128 batch size, and a classification loss weight of 0.8, which produced a mAP@50 of 82.1%.

* 1. Evaluation

For the evaluation phase, the inference was done using a test subset to assess if the model meets the business criteria (mAP@50 ≥ 60%). However, other metrics are fundamental for identifying the performance of the final multi-class object detection model.

### Intersection over Unit (IoU)

This metric measures the difference between the predicted and the actual bounding box, which is essential to analyse the precision of the model with object location (Ultralytics, 2024). The IoU score ranges from 0 to 1. The perfect score is 1, meaning the model predicted the object's location with 100% precision.

A comparison of a number of objects

AI-generated content may be incorrect.

Figure 13. Intersection over Union (IoU ) Illustration ([Padilla et al. 2021)](#metrics_2)

IoU is essential for determining errors and failures of the objects detected. The IoU threshold was set to 0.7, meaning during the validation, a prediction is only considered a True Positive if it had an IoU ≥ 0.7. Otherwise, it is considered a False Positive, and if the real object is not detected, it is marked as a False Negative ([Padilla et al. 2021)](#metrics_2). The IoU influences all the following metrics.

### Precision and Recall

Precision measures the quality: whether the objects detected were correctly classified. Recall measures the quantity: how many objects of an image were detected. They are calculated based on components of the confusion matrix, which compares the model's predictions with the actual labels [(Géron, 2019)](#hands_on_book):

* **True Positives:** when the model correctly detects an object (IoU ≥ 0.7).
* **False Positives:** when the model detects an object incorrectly (IoU < 0.7).
* **False Negatives:** when the model fails to detect an object present in the image.

Here is how Precision and Recall are calculated for each class:

The individuals and overall values are automatically calculated and displayed at the end of the prediction.

|  |  |  |
| --- | --- | --- |
| Class | Precision | Recall |
| bicycle | 0.621 | 0.654 |
| helmet | 0.66 | 0.688 |
| scooter | 0.942 | 0.95 |
| vest | 0.925 | 0.922 |
| all | 0.787 | 0.804 |

Table 2. Precision and Recall Results

The overall performance of the model is satisfactory. The model recalled 80,4% of the objects present in the dataset. When making a prediction, it was correct 78,7% of the time. However, looking at individual classes, such as bicycle and helmet, the model failed to detect up to 37% of the objects (false negatives) and misclassified up to 24% of the present objects (false positives). This highlights the low recall and precision compared to other classes, such as vest and scooter.

Further analysis using the confusion matrix shows precisely where the model is successful. For example, the True Positives are along the diagonal line, where the model prediction is the actual object. The values around the edges are either False Positives or False Negatives ([Padilla et al., 2021)](#metrics_2). Most errors confuse classes with background, especially helmets and bicycles.

A group of blue and green squares

AI-generated content may be incorrect.

Figure 14. Confusion Matrix

### Mean Average Precision (mAP)

The Average Precision (AP) is calculated using the area under the precision-recall curve considering different confidence thresholds. It is important to show the model balance between precision and recall, whether the model can find most of the objects and detect the objects of this class with few false positives ([Padilla et al. 2021)](#metrics_2).

Therefore, the mean Average Precision is the average AP of all the classes in the dataset to give an overall performance. The mean Average Precision at IoU of 50% is provided by calculating the average precision of only the predictions where IoU ≥ 0.5, then taking the mean of these values to obtain the mAP@50.

Analysing the test subset metrics results with the final model at the metric dictates the model success criteria defined in the business understanding section. The conclusion is that the model was successful considering the target of mAP@50 ≥ 0.6 since the model achieved a **mAP@50 of 0.827** (*82.7%*).

|  |  |  |
| --- | --- | --- |
| Class | mAP@50 | mAP@50–95 |
| Bicycle | 0.649 | 0.411 |
| Helmet | 0.719 | 0.404 |
| Scooter | 0.972 | 0.782 |
| Vest | 0.969 | 0.816 |
| All | **0.827** | **0.603** |

Table 3. Mean Average Precision at IoU 0.5 and 0.5 – 0.95

The overall performance of the final model is satisfactory. However, some areas can improve; for example, the helmet and bicycle classes performed below the other two classes of the dataset. The visual example below highlights these differences. The model detects some classes with more precision than others.

A collage of images of people on a street

AI-generated content may be incorrect.

Figure 15. Labels vs Prediction

* 1. Deployment

The deployment phase is a simulation, as the model is a proof of concept rather than a final model for production. The videos were recorded in two different scenarios, on a bicycle and a scooter and during day and night, to simulate a cycle lane monitoring system. After that, it was tested on the final model using the prediction method.



Figure 16. Cycle lane Monitoring System Simulation

The simulation reflected the model's performance evaluated previously. The easier object to detect was the scooter, followed by the vest when closer to the camera. The bicycles were detected with low precision, and the helmet was detected only in one specific scenario with the lowest confidence.

A collage of images of people riding a bicycle

AI-generated content may be incorrect.

Figure 17. Details of the Simulation

Chapter 4

# Results

The model described in the previous chapter is the product of intuition, a few mathematical insights, and much trial and error. The metrics below are the prediction metrics results of the test subset used to define the model's effectiveness.

A number of numbers on a white background

AI-generated content may be incorrect.

Figure 18. Test Subset Result

The graph below shows the precision-recall curve. This shows how the model performs under different thresholds. Highlights the difference between classes. In comparison, the blue and orange lines (bicycle, helmet) had their curve more to the centre, less precise and low recall. The green and red lines (scooter, vest) have curves closer to the upper right corner, indicating better balance over precision and recall.

A graph of different colored lines

AI-generated content may be incorrect.

Figure 19. Precision-Recall Curve Graph

Although there is room for improvement, the overall precision and recall are acceptable. And the model achieved a **mAP@50 of 0.827**, which surpassed the business criteria.

Chapter 5

# Conclusion

The project aimed to build a computer vision model on cycle lanes that detects bicycles and scooters, high visibility vests, and helmets. This project was intended as a proof concept rather than a final model for production. Its objective was to provide a foundation for robust models integrating computer vision techniques into existing cycle lane monitoring systems.

To succeed, a costumed dataset was collected from open-source datasets, cleaned and customised to the model objective. Then, I trained the model using the YOLOv8 pre-trained weights to transfer learning to speed up the training phase. Multiple combinations of hyperparameters were tested until the best model was selected and evaluated using the test subset. After concluding that the model surpassed the project's initial success criteria (≥ 0.6) with a mAP@50 of 0.827. Following this, the final model was used in a simulated environment to show how it performed.

The implementation phases of this model proved that computer vision can be used in cycle lane monitoring systems to extract relevant data. Computer Vision automates the process and offers support to make decisions about infrastructure investments, campaigns to promote cyclist safety awareness and analysis of its effectiveness, and future introduction of helmet regulations.

Despite the promising advantages computer vision offers, there are challenges. The central issue is data. Training a state-of-the-art CV model requires a large amount of data, so variations of illumination, scale and partial appearance of objects are reduced. The labels need to be standardised, so the models know the bounding box of an object.

For future work, an idea is to collect data from the cycle lanes where the models would be used. These measurements would increase the model's overall performance, especially of the underperformed classes, such as helmets and bicycles.

# References

World Health Organization, 2022. *Cyclist safety: an information resource for decision-makers and practitioners* [online]. Geneva: World Health Organization. Available at: <https://www.who.int/publications/i/item/cyclist-safety-an-information-resource-for-decision-makers-and-practitioners> [Accessed: 16 March 2025]

Thompson, M., Lowry, M. and Abdel-Rahim, A. (2024) *Computer Vision for Traffic Monitoring: Final Project Report.* Pacific Northwest Transportation Consortium (PacTrans), University of Idaho. Available at: https://rosap.ntl.bts.gov/view/dot/79662 (Accessed: 7 March 2025).

Dilek, E. and Dener, M. (2023) *'Computer Vision Applications in Intelligent Transportation Systems: A Survey'*, *Sensors*, 23(6), p. 2938. Available at: <https://doi.org/10.3390/s23062938>

Olivier, J., Boufous, S. and Grzebieta, R. (2019) *'The impact of bicycle helmet legislation on cycling fatalities in Australia',* International Journal of Epidemiology, 48(4), pp. 1197-1203. Available at: <https://doi.org/10.1093/ije/dyz003>

Goodfellow, I., Bengio, Y. and Courville, A. (2016) Deep Learning. Cambridge, MA: MIT Press.

Breum, S.M., Kostic, B. and Szell, M. (2022) *'Computational desire line analysis of cyclists on the Dybbølsbro intersection in Copenhagen'*, Findings, December. Available at: <https://doi.org/10.32866/001c.56683>

Li, Z., Yang, W., Peng, S. and Liu, F. (2020) *'A survey of convolutional neural networks: analysis, applications, and prospects*', arXiv, 2004.02806. Available at: <https://doi.org/10.48550/arXiv.2004.00280>

Sirisha, U., Praveen, S.P., Srinivasu, P.N., Barsocchi, P. and Bhoi, A.K. (2023) *'Statistical analysis of design aspects of various YOLO-based deep learning models for object detection'*, International Journal of Computational Intelligence Systems, 16(1), p.126. Available at: <https://doi.org/10.1007/s44196-023-00302-w>

Tedrake, R. (2024) *Robotic Manipulation: Perception, Planning, and Control – Chapter 9.* Course notes for MIT 6.421. Available at: <http://manipulation.mit.edu> (Accessed: 18 March 2025)

Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S. and Murphy, K. (2017) *'Speed/accuracy trade-offs for modern convolutional object detectors',* Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7310–7311. Available at: <https://doi.org/10.48550/arXiv.1611.10012>

Girshick, R., Donahue, J., Darrell, T. and Malik, J., 2014. *Rich feature hierarchies for accurate object detection and semantic segmentation*. arXiv preprint arXiv:1311.2524. Available at: <https://arxiv.org/abs/1311.2524>

Girshick, R., 2015. *Fast R-CNN*. In: 2015 IEEE International Conference on Computer Vision (ICCV). Santiago, Chile, 7–13 December 2015. IEEE, pp.1440–1448. DOI: 10.1109/ICCV.2015.169

Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., 2016. *You Only Look Once: Unified, real-time object detection*. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA, 27–30 June 2016. IEEE, pp.779–788. Available at: <https://arxiv.org/abs/1506.02640>

Plotnikova, V., Dumas, M. and Milani, F., 2020. *Adaptations of data mining methodologies: a systematic literature review.* PeerJ Computer Science, 6, p.e267. Available at: <https://doi.org/10.7717/peerj-cs.267>

Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. and Zitnick, C.L., 2014. *Microsoft COCO: Common Objects in Context*. In: D. Fleet, T. Pajdla, B. Schiele and T. Tuytelaars, eds. Computer Vision – ECCV 2014. Cham: Springer, pp.740–755. DOI: <https://doi.org/10.1007/978-3-319-10602-1_48>

Open Images Dataset, 2022. *Open Images V7*. [online] Google AI. Available at: <https://storage.googleapis.com/openimages/web/index.html> [Accessed 5 March 2025].

Roboflow, 2020. *Roboflow: Annotate, manage, and train computer vision models*. [online] Available at: <https://universe.roboflow.com/> [Accessed 8 March 2025].

Tzutalin, 2015. *LabelImg: Graphical image annotation tool*. [online] GitHub. Available at: <https://github.com/tzutalin/labelImg> [Accessed 5 March 2025].

Creative Commons, 2013. *Attribution 4.0 International (CC BY 4.0) license*. [online] Available at: <https://creativecommons.org/licenses/by/4.0/> [Accessed 19 March 2025].

Géron, A., 2019. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. 2nd ed. Sebastopol: O'Reilly Media.

Jocher, G., Chaurasia, A. and Qiu, J., 2023. *Ultralytics YOLOv8.* Version 8.0.0. [software] Available at: <https://github.com/ultralytics/ultralytics> [Accessed 10 March 2025].

Ultralytics, 2024. *YOLOv8 Models Documentation*. [online] Available at: <https://docs.ultralytics.com/models/yolov8/> [Accessed 20 March 2025].

Yaseen, M. (2024) *What is YOLOv8: An in-depth exploration of the internal features of the next-generation object detector*. arXiv preprint arXiv:2408.15857v1. Available at: <https://arxiv.org/abs/2408.15857> (Accessed: 18 March 2025).

Wang, C.Y., Liao, H.Y.M., Yeh, I.H., Wu, Y.H., Chen, P.Y. and Hsieh, J.W., 2019. *CSPNet: A new backbone that can enhance learning capability of CNN*. arXiv preprint arXiv:1911.11929. Available at: <https://arxiv.org/abs/1911.11929> [Accessed 18 March 2025].

He, K., Zhang, X., Ren, S. and Sun, J., 2015. *Spatial pyramid pooling in deep convolutional networks for visual recognition*. arXiv preprint arXiv:1406.4729. Available at: https://arxiv.org/abs/1406.4729 [Accessed 18 March 2025].

Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B. and Belongie, S., 2017. *Feature Pyramid Networks for Object Detection*. arXiv preprint arXiv:1612.03144. Available at: <https://arxiv.org/abs/1612.03144> [Accessed 18 March 2025].

Liu, S., Qi, L., Qin, H., Shi, J. and Jia, J., 2018. *Path Aggregation Network for Instance Segmentation*. arXiv preprint arXiv:1803.01534. Available at: <https://arxiv.org/abs/1803.01534> [Accessed 18 March 2025].

Ultralytics, 2025. *Performance Metrics Deep Dive.* Available at: https://docs.ultralytics.com/guides/yolo-performance-metrics/ [Accessed 12 May 2025].

Ramachandran, P., Zoph, B. and Le, Q.V., 2017. *Searching for activation functions.* arXiv preprint arXiv:1710.05941. Available at: <https://arxiv.org/abs/1710.05941>

Akiba, T., Sano, S., Yanase, T., Ohta, T. and Koyama, M., 2019. *Optuna: A next-generation hyperparameter optimization framework.* arXiv preprint, arXiv:1907.10902. Available at: <https://doi.org/10.48550/arXiv.1907.10902> [Accessed 9 May 2025].

Padilla, R., Passos, W. L., Dias, T. L. B., Netto, S. L., & da Silva, E. A. B. (2021). *A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit.* Electronics, 10(3), 279. Available at: <https://doi.org/10.3390/electronics10030279>

Appendix A

Roles and Responsibilities

This project was an individual effort. I was responsible for the whole project, from researching the topics to analysing the literature on the subject and documenting the process, including deciding which methodology to use and implementing it. I was also responsible for following the phases to develop, evaluate, and deploy the simulation (which I recorded myself in different vehicles and at times of the day) of the final computer vision model, proving that the technology could enhance current cycle lane monitoring systems.

Personal Reflection

Working on this project gave me a deep knowledge of Computer Vision technologies, the steps to develop a model for object detection, and the phases we must go through to create a model, understand the model's quality and know the limits related to data quality. The data collection and preparation phases were the most challenging as I used a customised dataset to develop my model. However, the project's objective interested me, which kept my hopes that I would achieve a good result. The project highlighted the importance of data for projects like mine. I have already left my acknowledgement at the beginning of the report.

I would like to leave my feedback on the Capstone Project. I mostly enjoyed it. However, it would greatly help future students if at least two supervisors orientate them weekly. I benefited from Professor Muhammed's feedback when I could speak to him due to the number of students. Regarding the time constraints, the fact that there are a few concomitant CAs did not help either. Overall, it is a challenging project, but it is doable, and you finish having a good knowledge of the chosen subject and a sense of accomplishment.

Appendix B

Links

This project's final dataset and other links are public for transparency. It can be accessed at:

GitHub Repository: [CA2 90 Joelma Rodrigues 2023246](https://github.com/CCT-College-Dublin/ca2-90-JoelmaRodrigues2023246)

Dataset

* **Kaggle Datasets:** [ciclovia-yolo.zip](https://www.kaggle.com/datasets/joelmarodrigues/ciclovia)
* **Google Drive:** [ciclovia-yolo.zip dataset](https://drive.google.com/file/d/1b0QnPMZSy6Adt3LHLIevtRys6PA3rKhC/view?usp=sharing)

Videos

* **Deployment Simulation:** [Cycle Lane Detection – Video Demo (CCT Final Project)](https://www.youtube.com/watch?v=vJMxqcAvS1o)