**A Study into The Convolutional Neural Network Object Detection Performance Between an Emulator Dataset and A Real-Life Dataset for Billboards Detection**

Han Wang

Faculty of Engineering, Environment and Computing, Coventry University

300COM: Individual Project (Informatics)

Supervisor: David Croft

Assignment Due Date: 30/04/2021

**300COM / 303COM Declaration of originality**

***I Declare that This project is all my own work and has not been copied in part or in whole from any other source except where duly acknowledged. As such, all use of previously published work (from books, journals, magazines, internet etc.) has been acknowledged by citation within the main report to an item in the References or Bibliography lists. I also agree that an electronic copy of this project may be stored and used for the purposes of plagiarism prevention and detection.***

**Statement of copyright**

I acknowledge that the copyright of this project report, and any product developed as part of the project, belong to Coventry University. Support, including funding, is available to commercialise products and services developed by staff and students. Any revenue that is generated is split with the inventor/s of the product or service. For further information, please see www.coventry.ac.uk/ipr or contact ipr@coventry.ac.uk.

**Statement of ethical engagement**

I declare that a proposal for this project has been submitted to the Coventry University ethics monitoring website (https://ethics.coventry.ac.uk/) and that the application number is listed below (Note: Projects without an ethical application number will be rejected for marking)

Signed: Han Wang Date: 2021/4/28

Please complete all fields.

| | |
|---|---|
| First Name: | Han |
| Last Name: | Wang |
| Student ID number | 9987188 |
| Ethics Application Number | P117469 |
| 1st Supervisor Name | David Croft |
| 2nd Supervisor Name | Tariq Aslam |

This form must be completed, scanned and included with your project submission to Turnitin. Failure to append these declarations may result in your project being rejected for marking.

# Certificate of Ethical Approval

| | |
|---|---|
| Applicant: | Han Wang |
| Project Title: | A study into the convolutional neural network Object detection performance and OCR Performance between an emulator dataset and a real-life dataset base on billboards detection |

This is to certify that the above named applicant has completed the Coventry University Ethical Approval process and their project has been confirmed and approved as Low Risk

| | |
|---|---|
| Date of approval: | 22 Feb 2021 |
| Project Reference Number: | P117469 |

## Abstract

This paper focuses on comparing the object detection performance difference between a simulator dataset and the real-life dataset to confirm whether the simulator dataset can replace the real-life dataset for billboard detection.

The experiments involve the objective of detecting advertising billboards through the use of a Convolutional Neural Network (CNN). To improve the detection performance of billboards as well as to save time, the network is retrained using transfer learning from the pre-trained neural network architecture (YOLOv4). Besides, the author has also proposed an Optical Character Recognition application based on object detection to extracting text from billboards, which can be applied to other different domains such as extracting road signs and notice boards.

The results showed that the model trained on the real-life dataset had an average precision of 92.16%, while the model trained on the simulator dataset had an average precision of 50.88%. By calculating the Two-Proportions Z-Test on F1-score, the obtained p-value is less than 0.00001. If the p-value is less than 0.05, which means there is a statistically significant difference between the simulated dataset and the real-life dataset. We can conclude that the simulator-based generated datasets cannot fully replace the role of real datasets in model training because the models trained on the simulated dataset performed worse than those trained on the real-life

dataset in object detection. However, combining both of them may be a good future research direction to address the shortage of real-life datasets.

**Acknowledgements**

First, thanks to David Croft, the author's supervisor, for his help during this project, second, to the statisticians at the Center for Mathematics and Statistics for their statistical assistance, and finally to the author family and friends for all the support and help to complete this project at this difficult time.

**Keywords**

Object detection, YOLOv4, Billboard detection, Robotics simulator, Autonomous vehicle, Optical Character Recognition, Darknet.

# Table of Contents

# 1. Introduction

The role of datasets in CNN training has received increased attention across several disciplines in recent years. Image detection performance is influenced by the network structure itself and the size of the dataset. In particular, the size of the dataset has a significant impact on the detector performance. For the same CNN-based network, the experimental results of average accuracy show that the larger the training set, the higher the test accuracy. (C. Luo et al., 2018)

For most people, it is difficult to collect large datasets in person because of privacy and copyright protection laws. With the global pandemic, there are significant restrictions on people travelling. (Chinazzi et al., 2020) , As a result, it has caused great difficulties for the collection of some real-world data sets.

A simulator-based dataset may play a significant role in addressing the issue of the insufficient number of datasets. The purpose of this paper is to find out whether the simulator dataset can replace a real-life dataset when the dataset is insufficient.

The object to be identified in this project is the billboards. Billboards are ubiquitous in our daily life as a modern mode of commercial advertising. Statistics show 68% of U.K. drivers admit to regularly making purchase decisions when interacting with billboards in their cars (Paul Inman, 2017). If we can accurately identify the location of billboards in images and extract their content, we can more effectively help advertisers analyse customer buying desires and more effectively place and maintain billboards.

Besides, an OCR application on the top of object detection will be proposed. This technique efficiently extracts the text content of billboards for future business analysis or personalised recommendations, which can also be applied to the recognition of text on road signs to improve Advanced Driver Automation Systems (ADAS) systems

Finally, suppose we can use it in Advanced Driver Automation Systems(ADAS), as self-driving cars can automatically collect datasets for us, which will be a huge saving in time and money spent on collecting datasets. In that case, the same object detection technique could also be used to identify pedestrians and road signs.

### 1.1 Project Aim

This project aims to compare the object detection performance between the simulator datasets generated by a robot simulator and the real-life datasets and extend their application to Optical Character Recognition (OCR).

### 1.2 Project Objectives

- Create different billboard samples in the simulated world.

• Collecting billboard images via self-driving cars in the simulator world.

• Collecting real-life billboard images by google open image datasets.

• Training real-life models and simulator models through Yolov4 transfer learning.

• Determine if the simulator dataset can replace the real dataset for billboards by comparing the two models' object detection performance differences.

• Extract the text content from billboards by optical character recognition.

### 1.3 Background and Context

Billboard detection was first introduced in soccer matches in the 1980s to correctly detect billboard areas in soccer games to replace the advertisers that appear in them effectively. However, there are relatively few historical studies in the area of CNN is used in the billboard detection field, and besides, access to datasets on billboards is very sparse.

### 1.4 Motivation

The motivation for this research comes from the fact that people are not able to travel freely to collect datasets due to the epidemic, and there is a lack of corresponding datasets on the internet. Therefore, this project provides a method to collect datasets from the simulator world in case of insufficient datasets to explore whether simulator datasets can replace real datasets by comparing their object detection performance.

In addition, the object of recognition in this study is billboards because billboards and other objects on the road, such as license plates, road signs, etc., have some similarity in colour and shape. Once we can successfully recognise billboards, we can apply them to other research areas as well.

### 1.5 Structure

The project will be structured as follows:

- Literature review – explore the current state of the art in billboard detection and recognition, as well as the gaps in related research areas.
- Research methodology – The steps of how to use self-driving cars to collect simulator datasets in the simulator world to train convolutional neural network models to recognise billboards are outlined. Besides, OCR will be applied for billboard text extraction after object detection.
- Analysis of results – analyse and compare performance differences between emulator datasets and real-world datasets by using Z-Test to show significant differences between them
- Project management – explains common agile development methods and project management tools used in the project.
- Conclusions – the overall results of the project are reflected and analysed, as well as some discussion of future research

## 1.6 Research Question

"Can simulator-based datasets replace real-life datasets in convolutional neural networks for billboard detection? "

# 2. Literature Review

## 2.1 Introduction

This chapter aims to introduce the reader to the state-of-the-art in understanding billboard advertising and detection. It then examines the application of convolutional neural networks in the field of object detection, which includes some of the common CNN frameworks such as Yolo and AlexNet as it relates to billboard detection.

It will then followed by an introduction to object detection performance metrics, which will help us understand object detectors' performance.

The simulator dataset in this project will be generated by a robotics simulator, so introducing the robotics simulator and some of the commonly used software will be presented. This will be followed by introducing the history of Optical Character Recognition it commonly used software, Tesseract. It will then explore how the project should build on previous research and the gaps in the relevant body of scientific knowledge.

## 2.2 Billboard Advertising

Billboard advertising is an Out-Of-House (OOH) advertising. It remains one of the most profitable and effective forms of advertising, even though it is one of the earliest advertising forms.

Billboards can capture customers' attention and engage them in a meaningful way. More specifically, a staggering 98% of British encounter at least one or more billboards every week, demonstrating the potential of such channels in the U.K. (Paul Inman, 2017)

On top of this, 71% of consumers say they regularly check and notice the information displayed on roadside billboards. While up to 68% of U.K. drivers admit to regularly making purchase decisions when interacting with billboards in their cars. (Visionsoutdoor, 2017)

Billboard advertising also offers more than one viable option and is up to 80% cheaper than selling the product on T.V. or other digital platforms. (Paul Inman, 2017)

In some ways, traditional billboards are seen as the antithesis of technology and innovation. However, this is not the case, as billboards and other OOH channels are increasingly dependent on data analytics and smart technology integration. This topic was explored in more detail by E.E. Mobile, who revealed that they could track exactly which users of specific sites and applications had been exposed to specific OOH ads. The study also found that awareness of unprompted ads doubled as a result, and the percentage of online searches increased by about 150%. (Bronwen Morgan, 2014)

Therefore, we propose a system in this paper that combines Convolutional Neural Network and OCR to detect billboards and extract the information to improve billboard marketing efficiency and improve online search rates after users see the billboard.

### 2.3 Billboard Detection

Much of the current literature on billboard detection pays particular attention to Sports T.V. They use a range of computer vision algorithms to detect the billboard on the image.

G. Cai et al. (2003) use Fast Hough transform and text geometry features to extract advertisement from Sports T.V. image. The basic idea is to analyse the variation of the white dots on the lines. The upper border of the Sports T.V. billboard conveys much more white pixels than the other lines. The Fast Hough transform will be used to summarises the white dots on the line segment. The result shows a roughly 90% accuracy in identifying billboards on Sports T.V.

The Fast Hough transform is a feature extraction technique used in image processing and feature detection, such as in Computer Vision. It finds the lines' parameters through a voting mechanism (the extreme value of the accumulator matrix). (Li et al., 1986)

$$\mathrm{Quan}(\mathrm{R}, \theta) = \sum_{x,y} BW(x, y) \qquad \mathrm{x}\cos(\theta) + y \sin\theta = \mathrm{R}$$

Figure 1. Fast Hough transform formula. (Li et al., 1986).

However, G. Cai et al.'s research does not consider Billboard colour and shape variations if the edge colour on the billboard is irregular, e.g., black and white, or a combination of different colours. The results of the research will vary considerably from expectations.

Medioni et al. (1999) also use the Hough transform to track billboards via video, intending to identify billboards' position for a future replacement to adjust audience advertising diversity. They use interest point operator, colour-based point filter, point matcher, and precise lock-in using Sum of Squared Differences (SSD):

$$\mathrm{SSD}(m, n) = \sum i \sum j [\, g(i, j) - t(i - m, j - n)]$$

Figure 2. Sum of squared differences formula. (Li et al., 1986)

The predictor uses Measure of Belief to detect and substitute billboard in a broadcast video, allowing the system to make replacement decisions based on complete sequences, thus avoiding mid-sequence on-screen billboard changes. One of the limitations of this work is that it does not include broadly applicable; once the objectives are changed, all work will no longer be applicable.

Together, these studies provide important insights into the detection of billboards. However, there have been no large-scale studies investigating the prevalence of billboards on roads and in urban environments to date. This is because, in previous work, occlusion and environmental complexity have been difficult aspects of billboard detection.

Unless the environment is uncomplicated, assuming an ordinary background and that objects are not obscured, edge and colour detection techniques can only be used for sports T.V. billboard detection because they have a uniform shape and a simple background environment.

In addition, previous works have relied on edges when detecting the shape of billboards. When one of the edges is covered by an obstacle, the detection will not detect the billboard. Furthermore, the colour detection method is only effective when the billboard has a specific colour and contrasts with the background colour. To overcome this problem, supervised machine learning methods such as CNN could be used to detect billboards in more complex environments, such as roads with many obstacles such as trees, pedestrians, vehicles, cables, utility poles, etc.

### 2.4 Convolutional Neural Network (CNN)

In recent years, convolutional neural networks (CNNs) have been applied to research in computer vision fields such as detection, classification, and recognition. one of the powers of CNNs is their ability to learn the features of objects.

Researchers have found that image classification accuracy can be improved using pre-trained neural network models with large learning capacities, such as YOLO, AlexNet, and GoogleNet. (Abdullah & M. S. Hasan, 2017)

In addition, deep convolutional neural networks (DCNN) have been shown to solve image classification problems for hard visual recognition tasks (Krizhevsky et al., 2017). Rahmat et al. (2019) use AlexNet's DCNN architecture with 60 million parameters, 650.000 neurons consisting of five convolutional layers, and three pooling layers to recognise different types of billboards.

| Layer | Size | Kernel | Neuron |
|---|---|---|---|
| Input: (Resized Image) | 224 x 224 x 3 x 1 | - | 150528 |
| Layer 1: Convolution + Max Pool | 55 x 55 x 48 x 2 | 96 | 290400 |
| Layer 2: Convolution + Max Pool | 27 x 27 x 128 x 2 | 256 | 186624 |
| Layer 3: Convolution | 13 x 13 x 192 x 2 | 384 | 64896 |
| Layer 4: Convolution | 13 x 13 x 192 x 2 | 384 | 64896 |
| Layer 5: Convolution + Max Pool | 13 x 13 x 128 x 2 | 256 | 43264 |
| Layer 6: Fully-Connected | 2048 x 2 | - | 4096 |
| Layer 7: Fully-Connected | 2048 x 2 | - | 4096 |
| Layer 8: Softmax Output | 1000 x 1 | - | 1000 |

Figure 3. AlexNet's DCNN architecture. (Rahmat et al. 2019)

The results showed that the system achieved a training accuracy of 92.7% for billboard detection.

W. Lan et al. (2018) obtained the YOLO-R network model based on the network structure of YOLOv2. It achieved good results in improving the real-time frame rate of pedestrian detection. It is experimentally shown that the algorithm improves the accuracy of pedestrian detection to 25 frames/s, which basically meets the requirement of real-time performance.

Mysore Jayakumar et al. (2019) have built a traffic violation system that automatically looks for a traffic violation of not wearing a helmet while riding a motorcycle and automatically extracting the vehicles' license plate number by using YOLO to detect the traffic violation and OCR to extract the plate number.

### 2.5 Object Detection Performance

There are several methods to determine the performance of an object detector designed to improve the detector's performance. These include mean accuracy (mAP), Intersection over Union (IoU), Recall, and F1-score.

All of these methods focus on the ground truth position or the actual position of the desired object. Precision and Recall are calculated using True Positives(TP), False Positives(FP), and False Negatives(FN) (Khandelwal, 2020).

Applied to this project, these metric represent:

- True Positive(TP): the detector can recognise the billboard when the image has a billboard.
- False Positive(FP): the detector can recognise the billboard, but there is no billboard.
- False Negative(FN): the detector says there is no billboard, but there is a billboard.
- True Negative(TN): the detector says there is no billboard and no billboard.

T.N. indicates that we did not detect any object in the image. This indicator is not involved in any calculation for object detection. Therefore, we ignore the TN.

IoU is an evaluation metric used in object detection challenges to measuring the accuracy of an object detector on a given dataset. (Adrian Rosebrock, 2016)

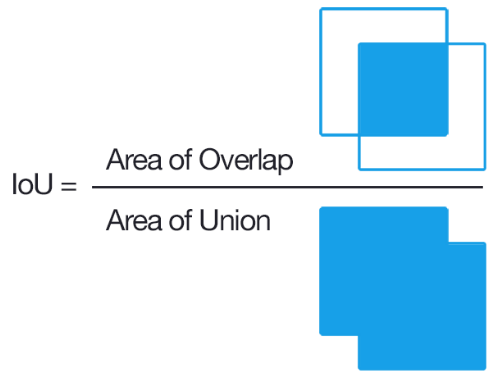



Figure 4. Intersection over Union (IoU) (Adrian Rosebrock, 2016)

A more rigorous approach could be to determine if the object detection is valid or not by setting the IoU object detector's accuracy on a particular dataset.

For example:

- If IoU $\geq$0.5, the detector will classify it as TP.
- If IoU <0.5, the detector will classify it as FP.
- When ground truth is present in the image and the model fails to detect it, classify it as FN.

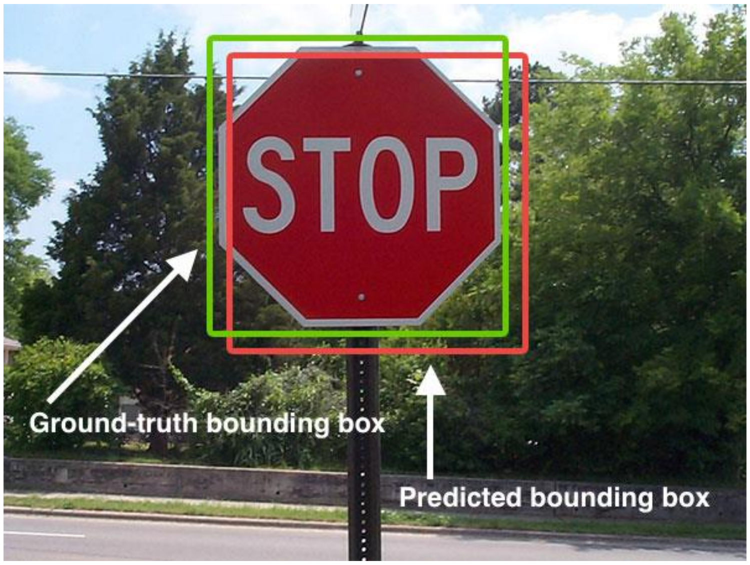



Figure 5. Ground-truth bounding box and Predicated bounding box are used to calculate the Intersection over Union (IoU). (Khandelwal, 2020)

### 2.6 Robotics Simulator

Webots is a free open source 3D robot simulator which widely used in education, research, and industry. The software is based on a combination of modern GUI (Qt), physics engine (ODE branch), and OpenGL 3.3 rendering engine. It also supports multi-platforms as it can run on Windows, Linux, and macOS.

In addition, Webots contains a large number of built-in modifiable robots such as two-wheeled tabletop robots, bipeds, industrial arms, modular robots, cars, drones, tracked robots, aerospace vehicles, and more. It also contains some sensors and real-world models such as road signs and billboards.

We can simulate the operation of self-driving cars collecting data sets in the virtual simulator world through its camera sensor and controller code. The controller code supports multiple languages such as C, C++, Python, or MATLAB.

We used Webots for this project for three reasons. Firstly, it is easy to use for beginners as it supports multiple programming languages without the need to learn other languages thus reducing the cost of learning. Secondly, it has a built-in billboard model rather than having to recreate it, reducing time costs. Thirdly, we can easily get help from the supervisor when we encounter some difficult problems.

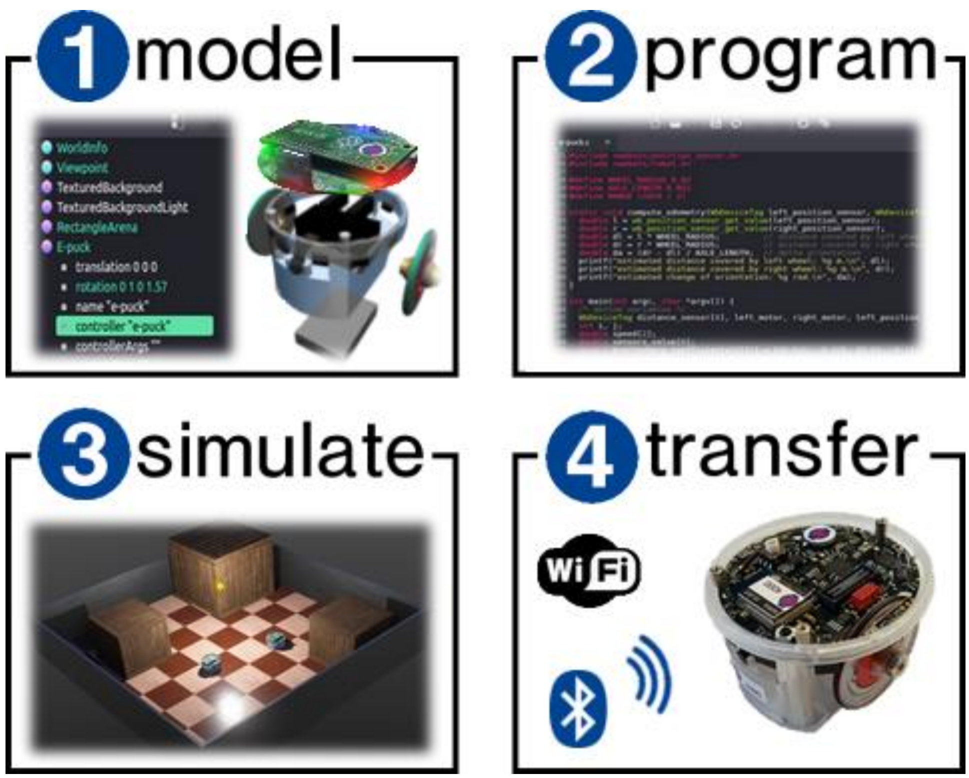



Figure 6. Four main functions of Webots. (Cyberbotics Ltd.,2020)

## 2.7 Optical Character Recognition (OCR)

The application of OCR to the recognition of billboard content might be very useful. Simply put, people interested in the billboard content can automatically search the relevant website or product through their smartphone.

People can also quickly locate a store's location by using some store billboard information, which could be a new direction for assisted location in the field of autopilot. Besides, OCR applications can be extended to other areas, such as recognising road signs with text, which will extend the usefulness of intelligent systems for self-driving cars.

Optical character recognition (OCR) methods were first used to convert printed fonts into editable text. OCR is a pragmatic method in different applications such as digitalised maps, recognition of vehicle license plates (Patel et al., 2012), text readers for visually impaired people, understanding handwritten office forms (J. Memon et al., 2020), bank checks, etc.

Patel et al. (2012) introduce the Optical Character Recognition (OCR) method, History of Open Source OCR tool Tesseract, its architecture, and experiment result of Tesseract's experiment different kinds of images. The results show that Tesseract has a strong performance in recognition of license plate information

Tesseract started as a PhD research project at H.P. Labs (M. Bhargava et al., 2016) for additional software to support H.P. scanners. The popularity of Tesseract was fuelled by the fact that commercial OCR engines were still in their infancy and performed poorly in all areas except print quality. Tesseract's main strength may have been its multiple function options available. Its main weakness was probably using polygon approximations as input to the classifier rather than the original contours. (R. Smith, 2007)

## 2.8 Gaps in Knowledge

There is a relatively small body of literature that is concerned with CNN-based detection in the field of billboard detection. In particular, this paper also presents a method for datasets collected from self-driving cars in a simulated world. There are relatively few historical studies in the area of CNN model training with a simulated dataset as well. Besides, the application of OCR build on object detection can be followed by an effective extraction of the text in it, which has not been considered by previous work. These studies are not only applicable to billboards but can also be used for other objects such as road signs, road markers, banners, etc.

# 3. Research Methodology

## 3.1 Introduction

The chapter will introduce the software used to generate the dataset and the methods used to clean the dataset, then it will explore why we use Darknet and YOLO and how to set the YOLO model training parameters to maximise the results, and finally, it will present the optical character recognition process and some experimental results of text extraction.

This paper will use Convolutional Neural Network (CNN) based on YOLOv4 because it is faster to do retaining than a completely new CNN. To reach this, we need to collect training data by using a robotic simulator (Webots), because this project has limited time, and it is easier to learn and get support from the supervisor.

## 3.2 Research Philosophy and Strategy

There are five main research philosophies: positivism, postmodernism, pragmatism, critical realism, and interpretivism. This project will use the positivist research philosophy, which assumes that facts can be measured statistically and confirmed by the scientific method, and the data set for this project contains a large sample of data sets and uses the statistical method Z-Test to demonstrate significance between two datasets. (Crossan, 2003).

### 3.3 Dataset Generation

The simulator-based dataset will be generated in Webots (Cybertronics n.d), which will involve obtaining Billboard images in the simulated worlds and placing them on the simulated world's billboard model. The image will contain some commercial advertisements as well as normal store advertisements. Also, in order to restore the reality of the authenticity, these pictures will be manually smudged, faded colour, physical damage, etc.

In the Webots simulator, the self-driving car can drive on the roads in the simulated world. Once the car camera sensor detects a billboard, it will save the image and annotation file, thus generating a dataset for later model training and billboard detection. To ensure that the simulator environment is close to the real environment, we also simulate different environmental conditions, such as sunny days, night, snow and different scenarios such as countryside, city, highway etc.

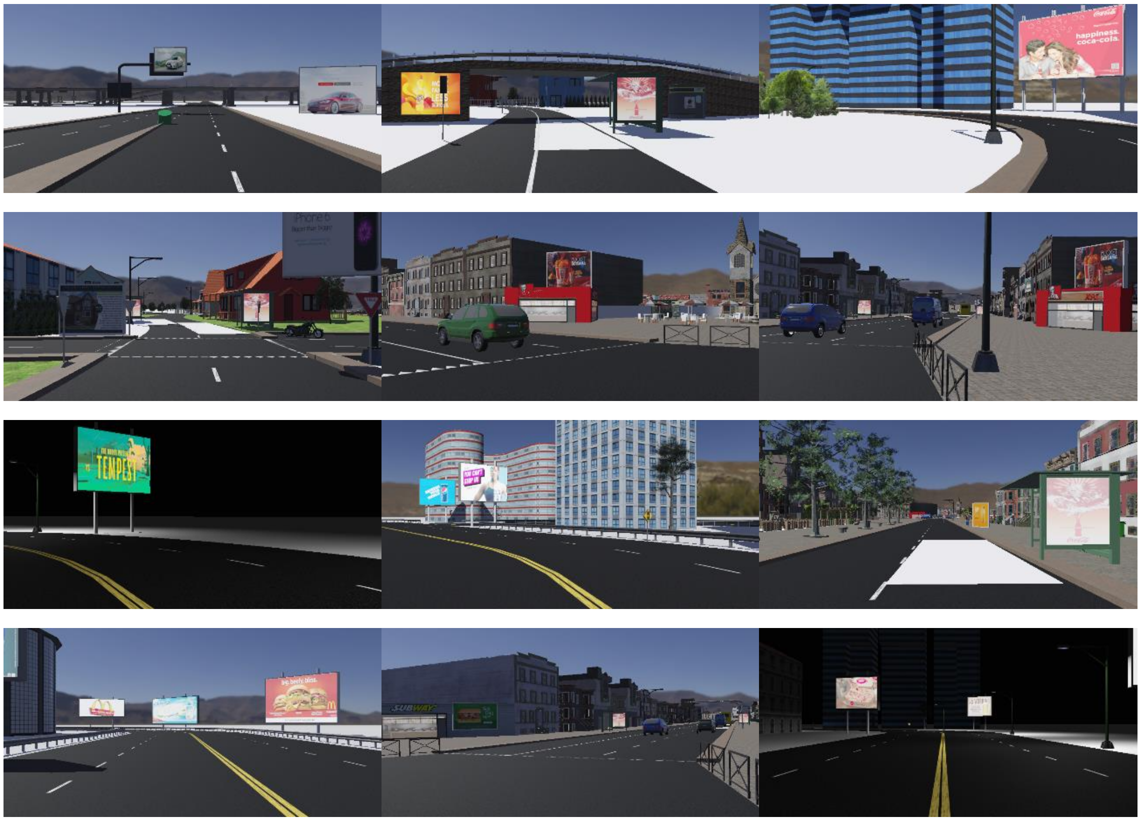



Figure 7. Simulator worlds in different scenarios. (Cyberbotics Ltd.,2020)

The real-life dataset will be pulled from Google Open Images, and so far, this is its sixth version containing over 9 million images. We will pull images of billboards from it as it could auto-generate their labels within minutes. We can use the official download.py[1] script to pull the dataset we want or use a third-party tool to download.

[1] https://raw.githubusercontent.com/openimages/dataset/master/downloader.py

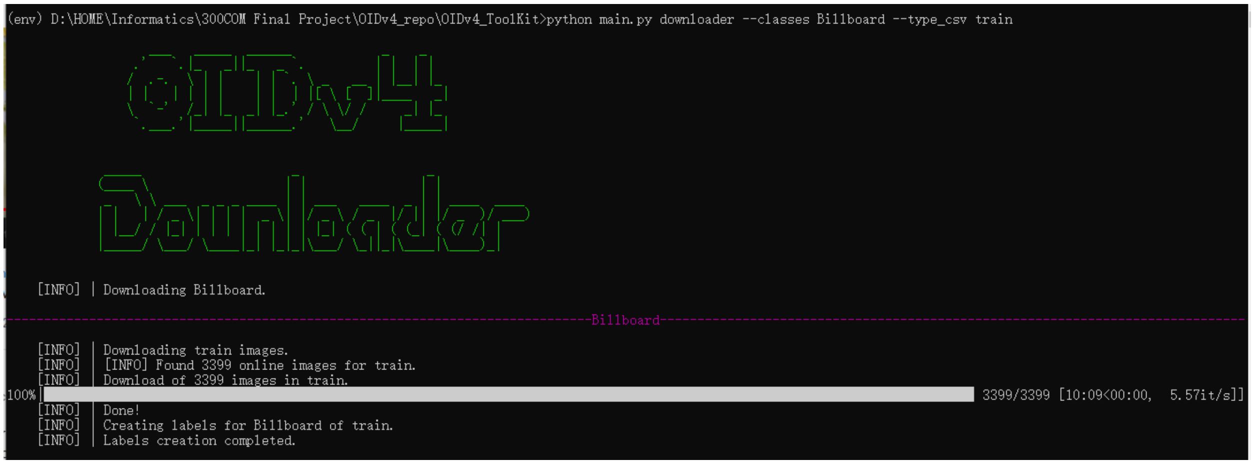



Figure 8. Download the dataset on billboards from Google Open Image

### *3.3.1 Controller Code*

The vehicle will follow a specified route with the help of a controller[2]. This section provides a flowchart of the controller code, as well as an explanation of its function.

The car will use the inertial unit and targets points to inform itself of its position in the simulated world and the direction to go. It is worth noting that the "supervisor" function needs to be set to "true". This will ensure that the car can use additional functions that are not available to normal robots.

As the car travels through the simulated world, the camera will automatically capture the target with the billboard and generate a png image format. An annotation text file contains information about the billboards' position in the image; with the image and annotation file, we can create a

[2] See Appendix 4

dataset to allow the Darknet to use train the dataset and distinguish which objects are billboards.

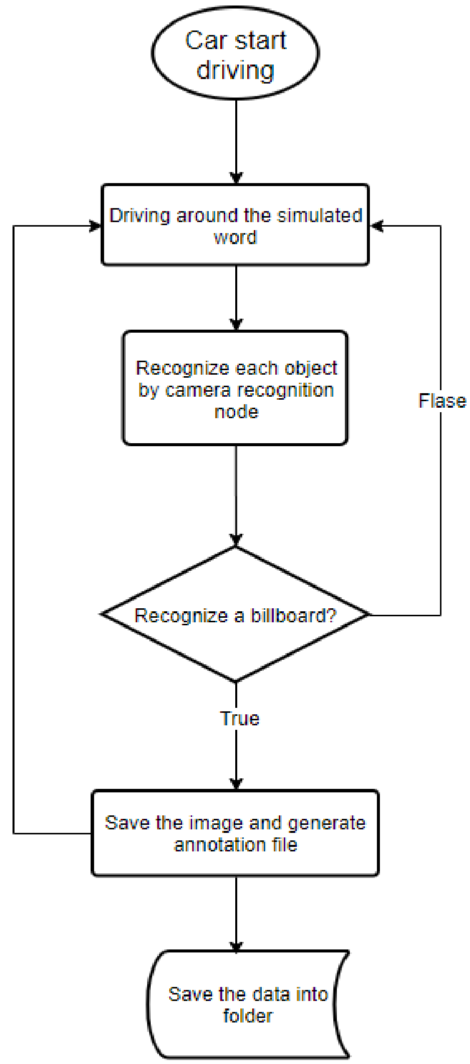



Figure 9. Controller code flowchart.

### 3.4 Data Cleaning

During the generation of the dataset, an empty annotation file occurs when the recognition node of Webots does not recognise the billboard in the camera feed. It would be detrimental to the training of the model if such data were still in the dataset. Therefore, it is necessary to write a python script to remove the invalid annotations and their corresponding images.

```python
import os
path =r'D:\HOME\Informatics\controllers\test00\images'
filenames = os.listdir(path)

#Iterate through all files in a folder
for filename in filenames:
    #Check if the file size is empty
    if os.stat(os.path.join(path,filename)).st_size == 0:
        #Slicing the file name and changing the extension to png
        os.remove(os.path.join(path,filename[:-3]+'png'))
        #Delete empty annotation text files
        os.remove(os.path.join(path,filename))
        print(f"Empty annotation: {filename} and it's image has now being removed")
```

Figure 10. Code for clean the dataset.

### 3.5 Experiment Constructs

The algorithm chosen for training the CNN will be YOLOv4 under the Darknet framework. Darknet is a cross-platform open-source neural network framework based on C and CUDA that supports both CPU and GPU. (Joseph Redmon, 2016)

We use Darknet because it is easy to operate and it supports CPU training, which will facilitate training in situations where computer hardware is inadequate. Secondly, Darknet has native support for YOLO, which will avoid some unnecessary problems that can be aggregated on model training. Finally, we can also get technical support from the supervisor of this framework.


Figure 11. Darknet (Joseph Redmon, 2016)

The model used for this project would be the YOLOv4 (You Only Look Once V4) (Bochkovskiy et al., 2020). We use it for the following reasons: First, it is a pre-trained model that already can recognise more than 80 classes from the COCO dataset (COCO n.d) which means we don't need to spend more time training our model from scratch. Second, YOLOv4 is fast and accurate than Other models, In mAP measured at 64.9%(see figure 16). Besides, there is a significant performance improvement over its predecessor, YOLOv3.

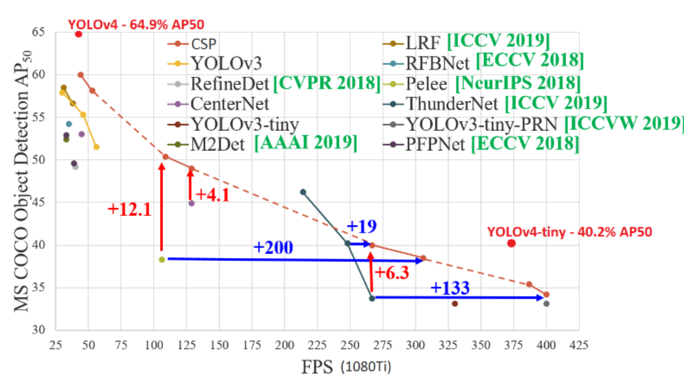



Figure 16. YOLOv4 performance comparison (Bochkovskiy et al., 2020)

### 3.6 Development

The YOLOv4 model will be trained on the two datasets we have generated by using transfer learning. Furthermore, as the algorithm is trained on one labelled dataset, it will be using supervised learning. By taking into account the time cost factor, this project's training will be done on the Google Colab platform, which has a free GPU.

Google Colab provides cloud virtual machines with powerful GPUs for students and researchers to conduct data-intensive and scientific experiments. There were three reasons for deciding to use Colab. The first reason is to simplify the installation process of Darknet. Secondly, it provides free GPU resources allowing us to train for more than twelve hours at a time, and finally, it can be embedded with Google Drive to save the trained weight files to ensure that the files are not lost.

Before training the model, we will make some considerations on the training parameters through the guidance of the YOLOv4 authors to ensure that the training remains relatively fair for both datasets to ensure that experimental results are not influenced by factors other than the dataset. Both the simulator-based dataset and the real-life dataset will be retrained in YOLOv4, which is based on the Darknet framework. We need to experiment with different training parameters to create a valid model, and these parameters include the number of classes, batches, steps, filters and maximum batches.

Classes refer to the number of different objects that the CNN was trained to recognise. In our case, this number was 1, as only billboards were recognised in this project. The spanning step was set to 1. Furthermore, the recommended formula by the darknet authors indicates that the value of the filter should be set to the number of classes plus 5 times 3, which in our case equals 18. Max_batches equals the number of classes multiplied by 2000, but it is worth noting that it cannot be less than 6000. For example, if we train 1, 2, or 3 classes, it will be 6000. However, a detector for five classes would have max_batches=10000. The step's value should be equal to 80% of max_batches, which in this project is 4800. The Filters' value is equal to classes plus five multiplied by 3, so it should be 18 in our case. The above values have been set according to the YOLOv4 authors to provide the best training results. The configuration file will be the same for both datasets to ensure that experimental results are not influenced by factors other than the dataset.

```
!./darknet  detector  train  data/obj.data  cfg/yolov4-obj.cfg  /mydrive/yolov4/backup/yolov4-obj_last.weights  -dont_show
```

```
!./darknet  detector  train  data/obj.data  cfg/yolov4-obj.cfg  /mydrive/yolov4/backup/yolov4-obj_last.weights  -dont_show
 157 conv   1024       3 x 3/ 1     19 x  19 x 512 ->   19 x  19 x1024 3.407 BF
 158 conv    512       1 x 1/ 1     19 x  19 x1024 ->   19 x  19 x 512 0.379 BF
 159 conv   1024       3 x 3/ 1     19 x  19 x 512 ->   19 x  19 x1024 3.407 BF
 160 conv     18       1 x 1/ 1     19 x  19 x1024 ->   19 x  19 x  18 0.013 BF
 161 yolo
[yolo] params: iou loss: ciou (4), iou_norm: 0.07, obj_norm: 1.00, cls_norm: 1.00, delta_norm: 1.00, scale_x_y: 1.05
nms_kind: greedynms (1), beta = 0.600000
Total BFLOPS 127.232
avg_outputs = 1046213
 Allocate additional workspace_size = 118.88 MB
Loading weights from /mydrive/yolov4/backup/yolov4-obj_last.weights...
 seen 64, trained: 192 K-images (3 Kilo-batches_64)
Done! Loaded 162 layers from weights-file
Learning Rate: 0.001, Momentum: 0.949, Decay: 0.0005
 Detection layer: 139 - type = 28
 Detection layer: 150 - type = 28
 Detection layer: 161 - type = 28
Resizing, random_coef = 1.40

 896 x 896
 Create 6 permanent cpu-threads
 try to allocate additional workspace_size = 258.14 MB
 CUDA allocate done!
Loaded: 0.000075 seconds
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 139 Avg (IOU: 0.702465), count: 2, class_loss = 0.505120, iou_loss = 14.480536, total_loss = 14.985656
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 150 Avg (IOU: 0.832056), count: 13, class_loss = 1.410845, iou_loss = 82.674332, total_loss = 84.085175
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 161 Avg (IOU: 0.850230), count: 5, class_loss = 0.506006, iou_loss = 4.957251, total_loss = 5.463257
 total_bbox = 20, rewritten_bbox = 10.000000 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 139 Avg (IOU: 0.827330), count: 1, class_loss = 0.003557, iou_loss = 8.101332, total_loss = 8.104889
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 150 Avg (IOU: 0.891409), count: 17, class_loss = 0.392353, iou_loss = 51.264008, total_loss = 51.656361
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 161 Avg (IOU: 0.918158), count: 13, class_loss = 0.157916, iou_loss = 10.521458, total_loss = 10.679374
 total_bbox = 51, rewritten_bbox = 3.921569 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 139 Avg (IOU: 0.807149), count: 3, class_loss = 0.648881, iou_loss = 30.178438, total_loss = 30.827320
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 150 Avg (IOU: 0.887272), count: 18, class_loss = 2.371045, iou_loss = 57.504612, total_loss = 59.875656
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 161 Avg (IOU: 0.905537), count: 16, class_loss = 0.241070, iou_loss = 20.518126, total_loss = 20.759195
 total_bbox = 88, rewritten_bbox = 10.227273 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 139 Avg (IOU: 0.000000), count: 1, class_loss = 0.000254, iou_loss = 0.000000, total_loss = 0.000254
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 150 Avg (IOU: 0.845314), count: 5, class_loss = 0.579095, iou_loss = 11.229162, total_loss = 11.808257
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 161 Avg (IOU: 0.892652), count: 12, class_loss = 1.026397, iou_loss = 10.558197, total_loss = 11.584595
```

Figure 12. Colab console.

Because Colab has a twelve-hour usage limit, all weight files will be automatically saved to Google Drive as a backup before the resource is closed. The parameter configuration files and the weight files have been uploaded to the repo for the supervisor to review.

### 3.5 OCR Text Extraction

After successfully locating the billboard in the image with YOLOv4, we can extract the billboard's text by locating the billboard's position. In order to properly extract the text from the image, we need to have thoroughly preprocessed the billboard.

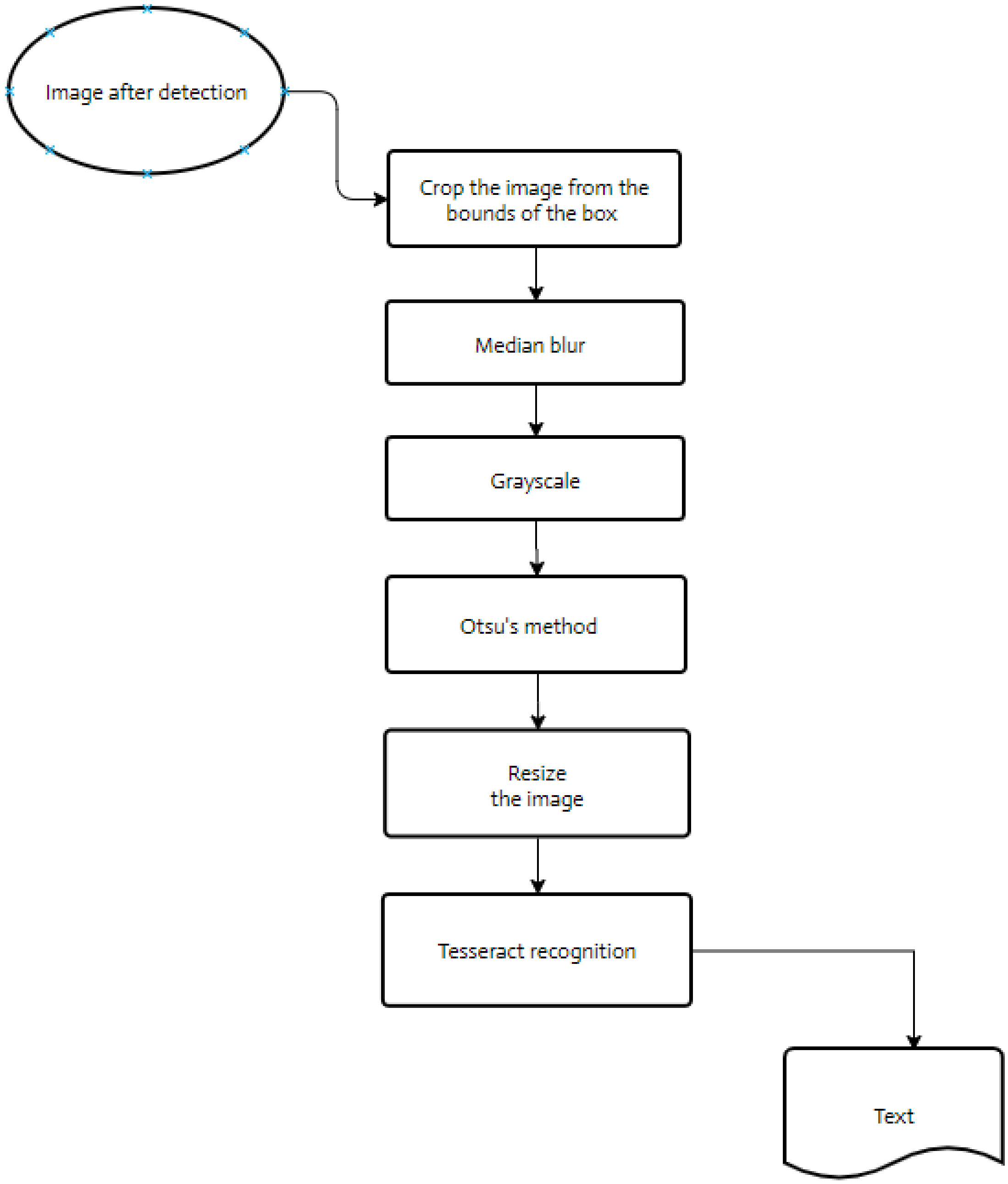



Figure 13. OCR process flowchart.

The first step is to crop out the main part of the billboard based on the detected image's detection frame. This step is to remove the redundant parts of the image to ensure that the OCR engine is more aware of the object.

In the next step, we adjusted the overall colour of the image to grayscale and applied a median blur effect to smooth out the entire image. After this, the image was converted to white text with a black background. Then we apply an algorithm to help find the outline of the text image, called Otsu's method.

In the next step, we resize the image to twice its original size, as the Tesseract works better at certain text sizes. Finally, we run the Tesseract and convert the image text to a string and output them at the terminal.

For the images captured in the emulator, the text cannot be extracted, probably because the billboard images have been distorted in the emulator. So we do not consider extracting the text of the billboard in the emulator world. In the below, we present some real-life billboard text extraction examples:

We can see in Figure 14 that the OCR application has successfully extracted the text from the billboard.

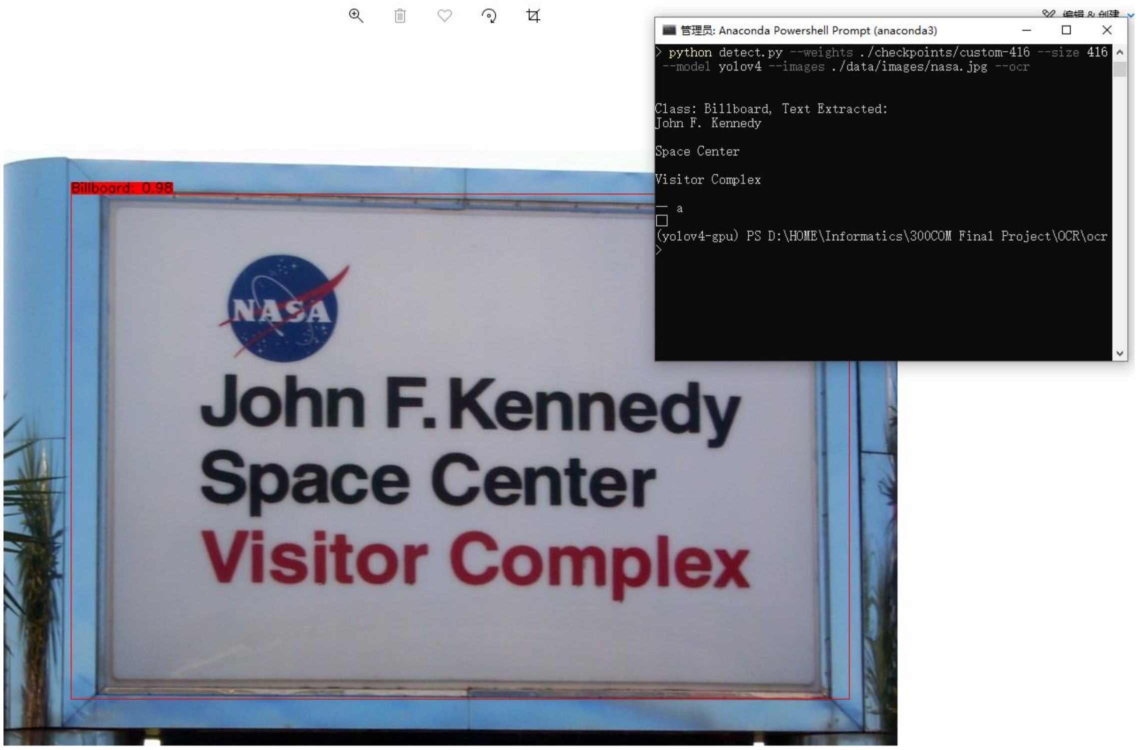



Figure 14. Successfully extracted the text out.

We can also extract information about road signs similar in shape to billboards, so the detector identifies them as billboards, proving that the same technology can also be used in road signs to improve autonomous driving systems.


Figure 15. Road sign OCR Application.

In Figure 16. OCR systems can also work for multiple billboard recognition, but for some artistic fonts and skewed images, the OCR function of extracting text does not achieve an ideal state.

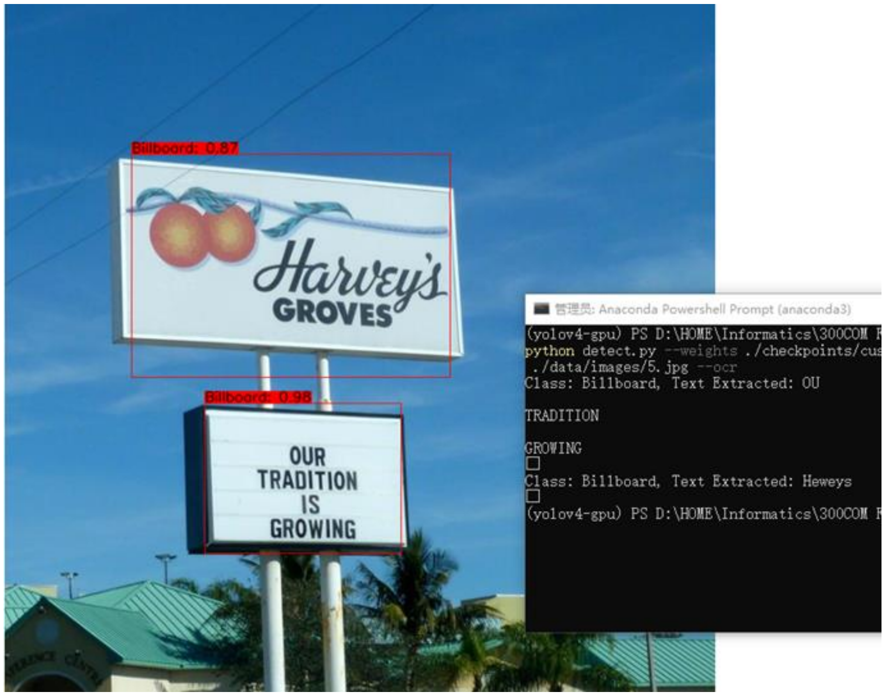



Figure 16. Multiple billboards.

# 4. Results and Analysis

This chapter provides an overview of how to calculate mAP, F1, IoU, and Precision-Recall values and evaluates the results generated from real-life datasets and simulator-generated datasets Two-Proportions Z-Test.

## 4.1 Testing Dataset

The same testing dataset will test the two trained weights to determine how they behave in object detection. To make the testing fair, the testing dataset contained 1000 images. The number of real-life billboards and simulated world billboards is 500, respectively.

Besides, the test dataset also contains images containing billboards and images where no billboards appear. The testing dataset images contain the respective scenes, including urban, rural, highway, and village.

Darknet does provide a command: darknet detector map to correctly calculate the mAP, F1, IoU, Precision-Recall of the weight. The performance metrics will be output on the terminal and are attached in Appendix 6.

```
!./darknet detector map data/obj.data cfg/yolov4-obj.cfg /mydrive/yolov4/backup/yolov4.weights
```

Figure 17. Detector map command.

### 4.2 Precision-Recall Curve (PR-Curve)

Recall indicates how many objects are detected by the detector, and precision indicates how many actual objects are in the bounding boxes we detect. Both of them are the basic metrics for object detection.

Precision-recall curves are used to demonstrate the trade-off between precision and Recall in unbalanced datasets.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive + False Positive}}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive + False Negative}}$$

Figure 18. Formulas for calculating Precision and Recall (Khandelwal, 2020)

### 4.3 Mean Average Precision (mAP)

The mAP is the average value of all classes in the dataset. In this experiment, since there is only one identified class, the value of mAP is equal to the average precision (A.P.), where the average precision means the average of 11 points on the Preston-Recall curve on a single class.

## 4.4 F1-Score

If we only consider the precision or Recall of the model, it does not fully reflect the performance of the detector, and that is the place where the F1 score needs to be introduced to find a balance between these two metrics. An F1-Score is designed to be a balanced measurement of Precision and Recall.

$$F1 - Score = 2 * \frac{Recall * Precision}{Recall + Precision}$$

Figure 19. Formulas for calculating F1-Score (Khandelwal, 2020)

## 4.5 Results

| Dataset | TP | FP | FN | Precision | Recall | F1-Score | Average IoU | AP |
|---|---|---|---|---|---|---|---|---|
| Real life Dataset | 2205 | 581 | 220 | 0.79 | 0.91 | 0.85 | 64.28% | 92.16% |
| Simulated Dataset | 1036 | 86 | 1389 | 0.92 | 0.43 | 0.58 | 83.28% | 50.88% |

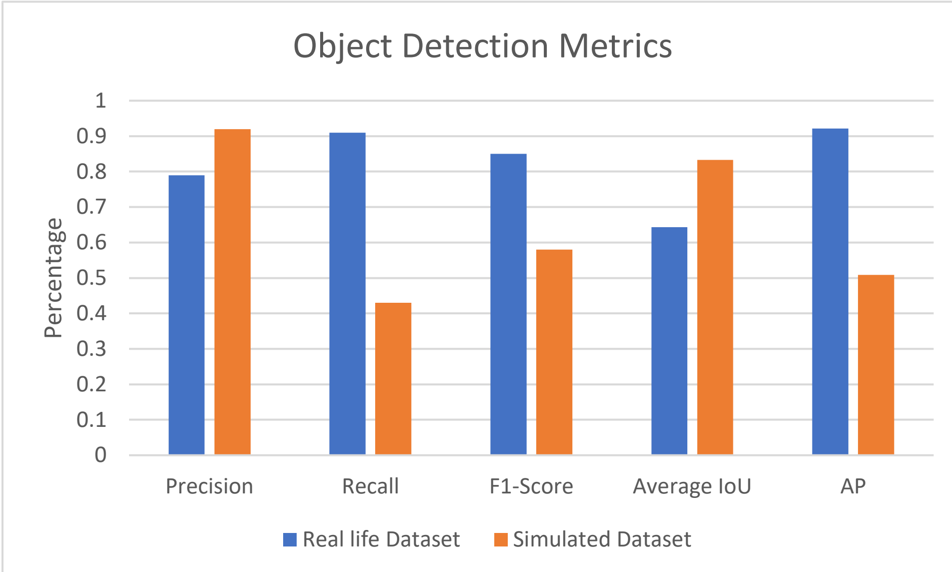



Figure 20. Comparison of object detection performance metrics between two datasets.

The results from the graphs show that the dataset generated from the simulated world lags far behind the model trained on the real-world dataset in Recall, F1-Score; the differences between them were 0.48 and 0.27, respectively. Furthermore, Compared to the real-world dataset, the simulated dataset is 41.28% less on Average Precision. However, it slightly exceeds Precision and Average IoU.

### 4.6 Two-Proportions Z-Test Analysis

The two-proportions z-test is used to compare two observed proportions as it tests the mean of a distribution. For each significance level in the confidence interval, the two-proportions Z-test has a single critical value (for example, 1.96 for 5% two-tailed). We can use it to determine whether two groups differ significantly on some single (categorical) characteristic. We want to see a significant difference in the F1-score because F1-score is the most comprehensive metric used to measure object detection performance.

The null hypothesis for the two-proportions is that the two proportions are the same. The alternate hypothesis is that the two proportions are not the same.

In our case, p1 is the F1-Score of Real-life Dataset, which is 0.85, p2 is the average score of Simulated Dataset, which is 0.58. The p is the overall proportion which is the T.P. of the two data sets added together and divided by n1+n2, which equals 0.74. The n1 and n2 are the sample size of each dataset, which is the denominator of the F1-score. It is added by Recall and Precision. The actual figure is 2*TP+FP+FN. The total sample size of the Real-life dataset and Simulated dataset are 5211 and 3547, respectively.

$$\frac{(\overline{p}_1 - \overline{p}_2) - 0}{\sqrt{\overline{p}(1 - \overline{p})\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}}$$

Figure 21. Two-Propotions Z-Test formulation.

By calculation, the value of z is 28.301, which is not in the 95% critical value accepted range: [-1.9600: 1.9600]. The value of p is < 0.00001. The result is significant at p <0 .05, which means there is a significant difference between the real-life dataset and the simulated dataset on the F1-score.

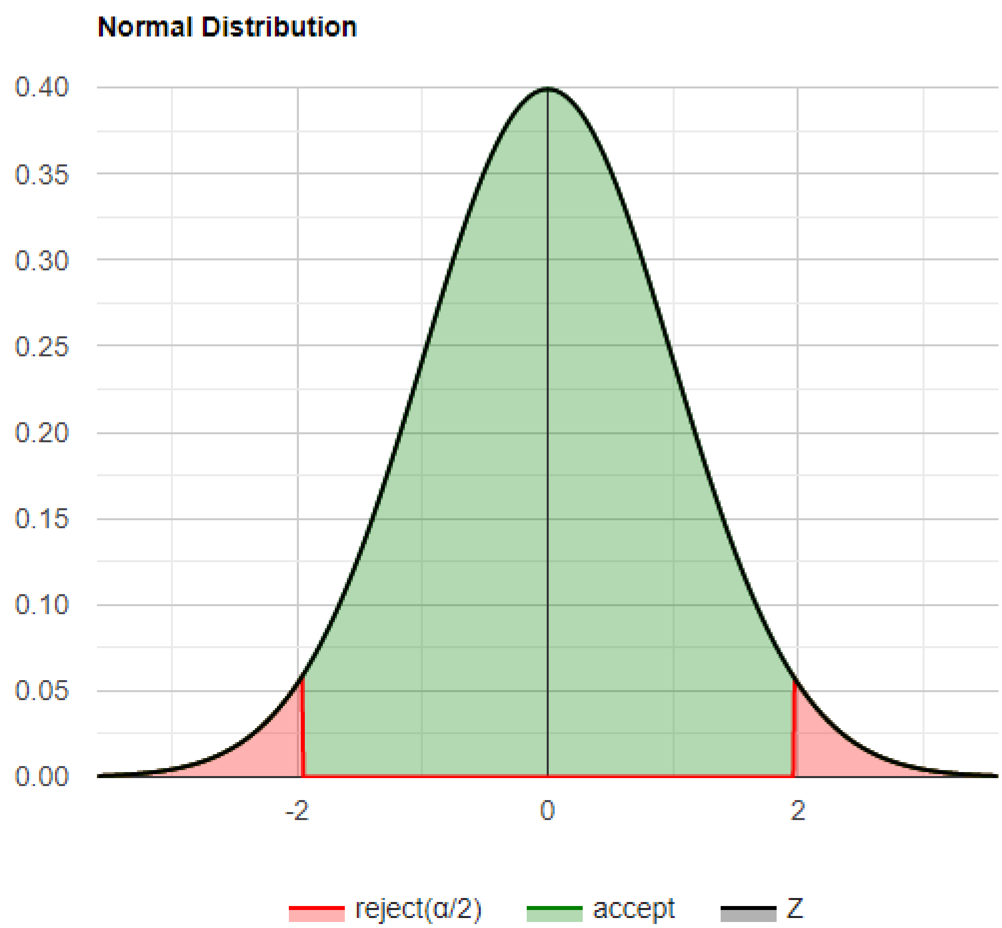



Figure 22. Normal distribution of Z-Test.

# 5. Project Management

## 5.1 Agile Development

Agile is a set of values and principles that specifically focus on agility, which is the ability to continually adapt and constantly improve the way we work. We can use several methodologies to follow the Agile principles and values, such as Scrum and Extreme Programming.

Both frameworks are concerned with producing a high-quality product with the fastest possible delivery. No general best framework is suitable for all scenarios. They each have their own strengths, weaknesses and use cases. In this project, I will use a combination of both.

### *5.1.1 Scrum*

Rather than a project with a remote deadline, we work on a project in which we are continuously delivering a functional product. We do not use final assessments, but we do receive constant feedback from the supervisors.

In a world of rapid change, Scrum is a way to work flexibly as it is used in I.T. projects regularly. In Scrum, there are three roles: the product owner, scrum master, and members. As this is a personal project, the author will act three roles at once. The supervisor will be seen as a customer and will provide ongoing follow-up and advice to improve the project.

The whole project will be broken down into smaller tasks and represented on the Kanban board, with each Kanban card moving from the left column to the right as time goes on. The Kanban board tasks will be divided into different cycles to be completed, which are called Sprints.

300COM Sprint Tracking Form

| Sprint | Task ID | Task | Status |
|---|---|---|---|
| Sprint 1 | 1 | Obtaining images of commercials from google image search | Complete |
| | 2 | Becoming acquainted with the Webots simulated environment | Complete |
| | 3 | Placing the images on billboard models in the virtual environment | Complete |
| Sprint 2 | 4 | Complete the ethics form | Complete |
| | 5 | Modify the proposal according to the feedback | Complete |
| Sprint 3 | 6 | Write python scripts for data collection | Complete |
| | 7 | Debug the controler code to make it work | Complete |
| | 8 | Collect images by driving the virtual car on roads in the simulated world | Complete |
| | 9 | Collect real life dataset from Google open images | |
| Sprint 4 | 10 | Training two datasets by YOLO under the darknet framework | Complete |
| | 11 | Extract text from the billboards by Tesseract OCR | Complete |
| Sprint 5 | 12 | Writing the abstract and introduction sections of the dissertation | Complete |
| | 13 | Writing the literature review and research methodologies sections | Complete |
| Sprint 6 | 14 | Obtain project results and drawing diagrams | Complete |
| | 15 | Writing the results analysis and project management sections of the project | Complete |
| | 16 | Writing project management part | Complete |
| Sprint 7 | 17 | Writing conclusions section | Complete |
| | 18 | Complete appendices and citations | Complete |
| | 19 | Revise the paper based on comments | Complete |

Figure 23. Sprint Tracking From.

### *5.1.2 Extreme Programming (XP)*

Extreme Programming XP is a lightweight, efficient and versatile approach to software development. It takes its name from taking elements of traditional software engineering in practices to the extreme. XP is based on a particular planning methodology, on-site customers and continuous testing.

Extreme programming uses some traditional principles to their fullest extent with some practice. As this is a computer thesis, the programming part is essential. We will focus on Simple design, constant testing, ongoing iteration, refactoring, coding standard, and small releases.

### 5.2 Version Control

This project uses Git to monitor code and help manage project progress. Repo is set to be a private mode in order to organise access without authorisation. Each time a new development is made on the project, a section of the code is submitted to the Coventry GitHub Enterprise. The supervisor will be set up as a collaborator and given permission to access. Repo address is displayed in Appendix 3.

### 5.4 Kanban Board

Kanban means visual signal in Japanese. This is a work management system that aims to help visualise work, limit work in progress and optimise work efficiency.

Kanban helps to visualise the work to understand it better. The purpose of Kanban is to categories all the stages of work through the workflow, from what we have not started to things that have been completed.

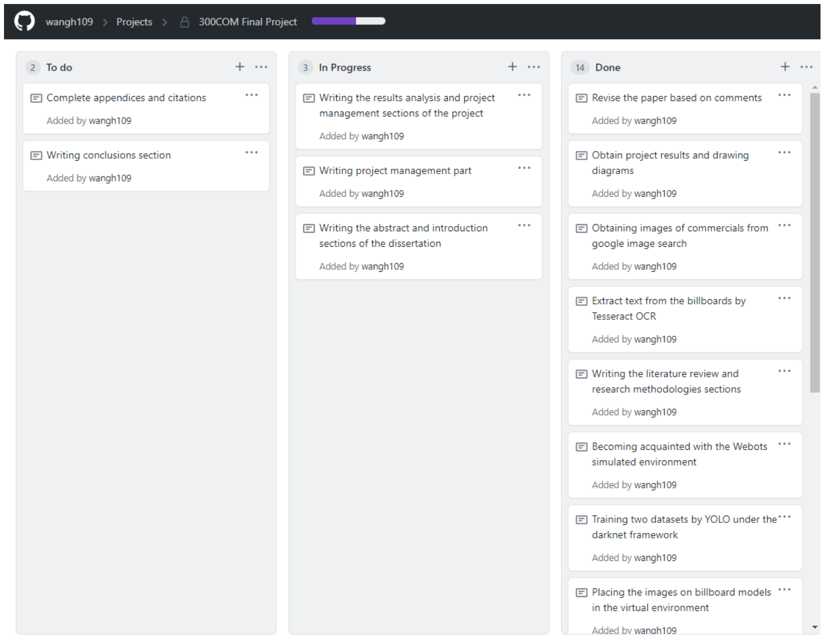



Figure 24. Kanban board for 300COM.

## 5.5 Gantt Chart

Gantt charts are a popular type of bar chart that shows the relationships inherent in the progress of projects, schedules, and other time-related systems over time. We use Gantt Chart to manage project time progress.

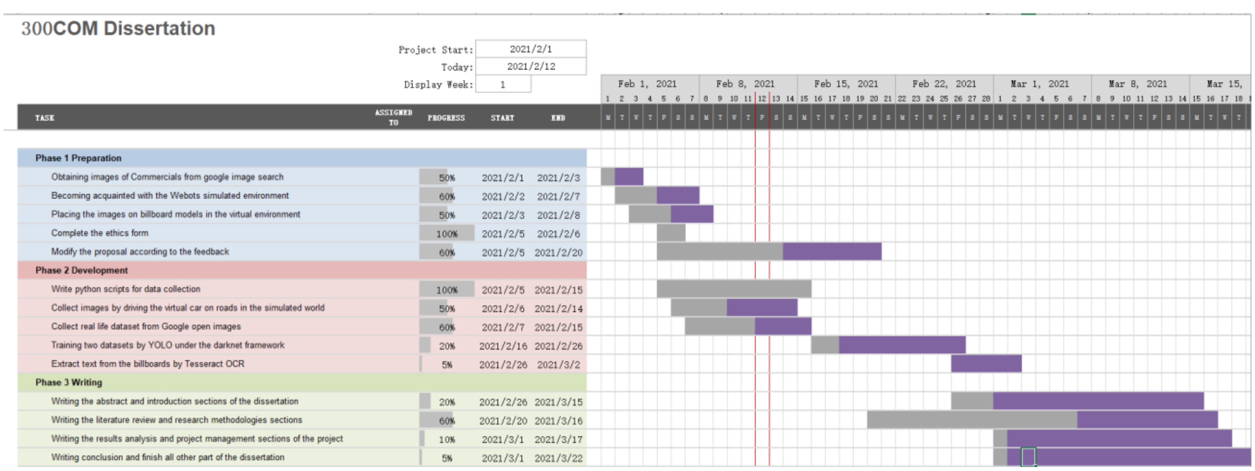



Figure 25. Gantt Chart for manage the project time.

## 5.3 Social, Legal, Ethical, and Professional Considerations

### *5.3.1 Social Considerations*

Creating a billboard recognition system and the ability to extract billboard text will make it possible to communicate advertising more efficiently. In the field of autonomous driving, the ability of cars to recognise billboards in front of them and recommend personalised advertising recommendations relative to them in the car will likely be a new direction for in-car advertising in the future. For the first time, some billboards can assist in locating shops, restaurants, or motels. Besides, one positive consequence of studying simulator-generated data sets instead of real ones is a reduction in $CO_2$ emissions, as when collecting real-world data sets, one inevitably has to travel from one place to another to complete the data collection.

### *5.3.2 Legal Considerations*

This project aims to identify billboards and extract their text content. Possible legal implications include advertising law, commercial law, and road safety law. If the billboard holder does not allow the collection of information from the billboard, there may be legal implications. Also, the collection of billboard information in self-driving cars may incorrectly identify billboards as other objects and cause injuries and accidents to pedestrians.

### *5.3.3 Ethical Considerations*

The collection of real-world billboard datasets will involve the participation of vehicles and pedestrians, which may affect portrait rights and property protected by copyright law, such as license plate numbers. The two main ethical issues here are pedestrians' facial features and the license plate numbers of vehicles. Generating datasets in the simulator world effectively avoids such risks, and Google has screened the data collected from the Google Open image database in this project to mitigate this ethical risk. Due to the creation of a world pandemic and government embargo restrictions, this project was not able to collect real datasets in person.

### *5.3.4 Professional Considerations*

By comparing the performance difference between the dataset generated by the simulator and the real-world dataset, we can provide a new way of thinking about training models to CNNs. In the field of CNN expertise, this can be an effective solution to the situation where the dataset is not large enough.

The paper also uses several project management methods such as Agile, Scrum, XP, Git, Kanban board, and Gantt Chart. These methods are used to continuously improve projects' progress and constantly change the criteria as the project is updated to meet the requirement.

## 5.5 Project Supervision Meeting

| Project Supervision Meeting | | |
|---|---|---|
| Date | Conference content | Reaction |
| 21/1/2021 | The main task this week was to complete the ethical clearance form. This is a low risk project as the author's project does not involve too many ethical issues | Complete the ethics review form and submit it to the supervisor and module leader for review by the deadline. |
| 28/1/2021 | The project proposal was completed this week and submitted to the supervisor for review, who raised issues such as the lack of headings between chapters, jumps in some paragraphs without logical support and the fact that project time management could be better presented through Gantt charts. | Revise the Proposal according to the comments, adding subheadings to distinguish the various sections of the literature review. Added Gantt charts to make the project time management more professional. Trimmed some unnecessary paragraphs |
| 4/2/2021 | No big problems this week, supervisor shared a markdown document about the structure of the dissertation description as well as shared how to introduce critical think in the paper, a good website is the phrase bank at the University of Manchester. | Read the provided Mardown document carefully and apply it to your own articles. Go to the Manchester phrase bank website for examples of sentences and phrases to add more critial to the paper |
| 11/2/2021 | The problem this week is that the controller code does not control the driving of the car. Supervior has shared his code for reference as well as helping me with camera size issues | Went to the Webots website to find the relevant documentation and found that the problem was that the latest version of webot had changed the way the API was used and it worked fine after updating the method |
| 18/2/2021 | Still on the technical side, when collecting photos in the emulator world, the billboard was found to be too dark and the supervisor informed that it could be made brighter by turning on the backlight button. When the backlight was turned on, the photos were again too bright and caused distortion. | The supervisor informs that it is possible to go into the webots prototype file and change the intensity of light emission in the billboard prototype file. Changing the intensity to 3 is a suitable value, and the billboard then most closely resembles a real-world billboard |
| 25/2/2021 | The main question for this week's meeting was how to add rigour to the experimental results of this article, and the supervisor suggested that help could be sought from the school's Maths and Statics Centre which is sigma Maths and Stats Support and reminded to book early as there would probably be no available seats when the end of term was approaching | Go to the website of the Centre for Mathematics and Statistics to reserve an available space. Fortunately, they offer drop in sessions and I had direct access to one of the consultants and helped me to understand and define the Object detection measure |
| 4/3/2021 | Reported on the latest progress and asked the supervisor if there was a presentation? | Continue to write the results and analysis section. and ongoing tracking of issue results to the supervisor |
| 18/3/2021 | Schedule a one-on-one presentation with supervisor | Prepare slides to show the current progress of the project |
| 25/3/2021 | An one-to-one presentations with supervisor to explain the current progress of the project and the problems encountered | Get valuble Feedback as well as changes will be made based on the issues raised and reflections will be specifically recorded in the paper |
| 1/4/2021 | Lacking a rigorous statistical method to prove the difference between the two data sets, the supervisor recommends going to the Mathematical Statistics Center for help | I went to a total of 3 times at the Math and Statistics Center and were initially given the option of using a t-test but were unable to derive a standard variance for our data and therefore could not use the t-test.On the last visit, the statistician informed me that I could use the Z-test. this solved the problem |
| 8/4/2021 | The code in the controller and OCR sections is replaced by pseudo-code or flowcharts, and I wasn't sure how to make them so I asked the supervisor for advice. | Through the tutor's explanation, I finally decided to use flowcharts because the source code is written in python which is already very close to natural language. |

Figure 26. Supervision meeting log.

## 5.6 Supervision Meeting Reflection

Choosing a direction for my final year project was difficult, and I had many other ideas before deciding to test the billboards, such as a self-driving car with automatic lane finding using a Raspberry Pi or building an envelope management system for student flats. It was a fun process to train the model using CNN networks, and the key thing was that I could get help from my supervisor. The author found the agile approach very useful during the implementation of the project. Breaking up the many tasks into individual tasks and integrating them into sprints made the project seem achievable, avoided frustrating situations, and, more crucially, allowed me to get ahead of schedule and avoid the deadline rush.

My strengths are in programming, which I am well aware of, so I am very fast at programming projects. However, I still lack academic writing, so I must complete all tasks as soon as possible and submit them to the supervisor for comments and improvements, enabling me to present a better project success. During the project, I learned different software, such as Webots and Darknet, and Tesseract. These were challenging because they were in different areas, but they gave me the confidence to do the whole project's job as each sprint progressed.

### 5.7 Response to Supervision

1. IoU is mentioned in the paper but not introduced.

    This point is addressed by reintroducing the introduction of the IoU

2. Word count exceeds the limit.

    This point is addressed by removing the ROS part in the literature review and other content.

3. There is no need to repeat the formula that already in the figures.

   This point is addressed by deleting the formula in the text.

4. There is no GitHub link in the paper.

   This point is addressed by adding a footnote to the repo to markup links

5. There are various ways of writing about YOLOv4 in the text and being consistent in referring to it. Is it YOLO4 or Yolov4, or YoloV4?

   This point is addressed by change all of them into one format -OLOv4

6. The text does not reflect the reason why the billboard should be identified

   This point is addressed by introducing a new section in the literature review about billboard advertising.

7. There is no direct answer to the question: Can simulator-based datasets replace real-life datasets in convolutional neural networks for billboard detection?

   This point is addressed by answer the questions in the summary of the results section directly.

8. Need some more examples of OCR application.

   This point is addressed by adding more examples of OCR and Introduce some shortcomings

9. The experiment lacks data statistics to demonstrate the difference between the two datasets.

   This point is addressed by adding a two-proportions z-test to get a p-value to demonstrate if there is a significant difference between the two datasets.

10. Controller code and OCR code seem too lengthy and difficult to understand.

This point is addressed by using flowcharts instead of code snippets to help understand.

11. The paper has some formatting issues, should following APA7th.

    This point is addressed by visiting the academic writing centre for help and modify the formatting issues on this paper.

# 6. Conclusions

## 6.1 Summary of Results

- A simulator is used to generate datasets for training convolutional neural network models to create authentic object detector in the event that there are no publicly available datasets.
- The results illustrate that the dataset created based on the emulator is still a little short of the real dataset regarding object detection performance and is not a complete replacement for the real dataset.
- Some of the text on the billboard was extracted by OCR.

Through this paper, we can answer the questions: "Can simulator-based datasets replace real-life datasets in convolutional neural networks base on billboard detection?"

The answer to our research question is No. The simulated dataset cannot fully replace the real-life dataset because the p-value on F1-Score is < .00001, when the p-value is less than .05, it means there is a significant difference between the two datasets. By comparing the figure of their F1-scores (0.85>0.58) and mAP (92.16%>52.88%) of the two datasets, it indicates that the simulated dataset is worse than the real-life dataset on object detection performance.

## 6.2 Relevance to Real World

This project's findings are significant and create an object detector that can be used in a number of varied areas, such as the recognition of road signs and the recognition of text on road signs containing text. One possible use is the integration of the detector into the External Human-Machine Interface (eHMI) (Moore et al., 2019) of a car and its application to Advanced Assisted Driving Systems (ADAS), examples of which are already used by some advanced car manufacturers such as Tesla.

### 6.3 Study Contributions

This project has successfully demonstrated that datasets generated based on simulator worlds cannot fully replace real-world datasets when the number of real datasets is insufficient. However, simulator-based datasets have the advantage of being easily accessible, especially when many subjects and scenarios cannot be replicated in real life. Furthermore, this knowledge can be used to improve object detector in other domains, not just billboard recognition and text extraction.

### 6.4 Degree of Reflection

The project's progress has been of great interest to the author, as have been the many challenges faced. As the project progressed, the authors had to change the project's objectives to accommodate the progress. For example, the project was initially intended to compare the difference in text extraction performance on OCR between the two datasets. However, as the OCR extraction results were not satisfactory, the entire OCR section was presented as a target to detect subsequent applications. Another issue was the very slow training progress due to the lack of computing power on the author's laptop, which was well overcome by borrowing the GPU from Google Colab. Overall, the project went well, particularly in terms of exceeding the expected timeline, which gave the author a large part of the time to refine and improve the text to meet the required standards.

The next step in the research could be to categories the billboards, for example, food, merchandise, restaurants, etc. This would facilitate the personalisation and placement of passenger advertising in self-driving cars. A potential improvement would be to increase both dataset's size, which would lead to more precision and recall values, resulting in a better-performing object detector. Another potential improvement would be to add more different billboard models to the simulator world, such as differences in size, dimensions, and shapes to enable better results.

Besides, we can introduce more statistical methods to measure and compare the performance difference between two datasets in many ways, such as t-test.

The OCR in this article is not ideal for extracting text as it does not extract all the text effectively. An improvement would be to add more methods to preprocess images for OCR to make the text easier to recognise and extract in the future.

## References


A. Harris, & J. M. Conrad. (2011). Survey of popular robotics simulators, frameworks, and toolkits. Paper presented at the - *2011 Proceedings of IEEE Southeastcon,* 243-249. https://doi.org/10.1109/SECON.2011.5752942

Abdullah, & M. S. Hasan. (2017). An application of pre-trained CNN for image classification. Paper presented at the - *2017 20th International Conference of Computer and Information Technology (ICCIT),* 1-6. https://doi.org/10.1109/ICCITECHN.2017.8281779

Rosebrock, A. (2016). Intersection over Union (IoU) for object detection. *Online] http://www. pyimagesearch. com/2016/11/07/intersection-overunion-iou-for-objectdetection*.

Aldershoff, F., & Gevers, T. (2003). Visual tracking and localisation of billboards in streamed soccer matches. Paper presented at the*, 5307.* https://doi.org/10.1117/12.526871 https://doi.org/10.1117/12.526871

Ammar, A., Koubaa, A., Ahmed, M., & Saad, A. (2019). Aerial images processing for car detection using convolutional neural networks: Comparison between faster R-CNN and YoloV3. https://arxiv.org/abs/1910.07234v2

B. Q. Vuong, & H. N. Do. (2014). Design and implementation of multilanguage name card reader on android platform. Paper presented at the - *2014 International Conference on Advanced Technologies for Communications (ATC 2014),* 528-531. https://doi.org/10.1109/ATC.2014.7043445

Bochkovskiy, A., Wang, C., & Liao, H. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv Preprint arXiv:2004.10934,*

Bo-Yao Lin, & C. Chen. (2015). Two parallel deep convolutional neural networks for pedestrian detection. Paper presented at the - *2015 International Conference on Image and Vision Computing New Zealand (IVCNZ),* 1-6. https://doi.org/10.1109/IVCNZ.2015.7761510

Bronwen Morgan. (2014). *Mobile data boosts OOH advertising by 200% | news.* http://www.research-live.com/article/news/mobile-data-boosts-ooh-advertising-by-200/id/4011796

C. Luo, X. Li, L. Wang, J. He, D. Li, & J. Zhou. (2018). How does the data set affect CNN-based image classification performance? Paper presented at the - *2018 5th International Conference on Systems and Informatics (ICSAI),* 361-366. https://doi.org/10.1109/ICSAI.2018.8599448

Chinazzi, M., Davis, J. T., Ajelli, M., Gioannini, C., Litvinova, M., Merler, S., Pastore y Piontti, A., Mu, K., Rossi, L., Sun, K., Viboud, C., Xiong, X., Yu, H., Halloran, M. E., Longini, I. M., & Vespignani, A. (2020). The effect of travel restrictions on the spread of the 2019 novel coronavirus (COVID-19) outbreak. *Science, 368*(6489), 395-400. https://doi.org/10.1126/science.aba9757

Crossan, F. (2003). Research philosophy: Towards an understanding. *Nurse Researcher (through 2013), 11*(1), 46.

Cyberbotics Ltd. (2020). *Webots user guide.* https://cyberbotics.com/doc/guide/index

Du, J. (2018). Understanding of object detection based on CNN family and YOLO. *Journal of Physics: Conference Series, 1004,* 012029. https://doi.org/10.1088/1742-6596/1004/1/012029

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision, 88*(2), 303-338.

G. Cai, L. Chen, & Junchang Li. (2003). Billboard advertising detection in sport T.V. Paper presented at the - *Seventh International Symposium on Signal Processing and its Applications, 2003. Proceedings. , 1* 537-540 vol.1. https://doi.org/10.1109/ISSPA.2003.1224759

G. Liu, C. Wang, & Y. Hu. (2018). RPN with the attention-based multi-scale method and the adaptive non-maximum suppression for billboard detection. Paper presented at the - *2018 IEEE 4th International Conference on Computer and Communications (ICCC),* 1541-1545. https://doi.org/10.1109/CompComm.2018.8780907

H. Jiang, & E. Learned-Miller. (2017). Face detection with the faster R-CNN. Paper presented at the - *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (F.G. 2017),* 650-657. https://doi.org/10.1109/FG.2017.82

Ichimura, N. (2006). Recognising multiple billboard advertisements in videos. Paper presented at the *Advances in Image and Video,* 463-473.

Ordelman, J. H. S. (2010). *Automatic Billboard Detection* (Doctoral dissertation, Thesis. Twent: University of Twente).

J. Memon, M. Sami, R. A. Khan, & M. Uddin. (2020). *Handwritten optical character recognition (OCR): A comprehensive systematic literature review (SLR).* https://doi.org/10.1109/ACCESS.2020.3012542

Kerzner, H. (2017). *Project management: A systems approach to planning, scheduling, and controlling*. John Wiley & Sons.

Khandelwal, R. (2020). *Evaluating performance of an object detection model.* https://towardsdatascience.com/evaluating-performance-of-an-object-detection-model-137a349c517b

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Commun.ACM, 60*(6), 84-90. https://doi.org/10.1145/3065386

Li, H., Lavin, M. A., & Le Master, R. J. (1986). Fast hough transform: A hierarchical approach. *Computer Vision, Graphics, and Image Processing*, *36*(2), 139-161. https://doi.org/10.1016/0734-189X(86)90073-3

M. Bhargava, P. Dhote, A. Srivastava, & A. Kumar. (2016). Speech enabled integrated AR-based multimodal language translation. Paper presented at the - *2016 Conference on Advances in Signal Processing (CASP),* 226-230. https://doi.org/10.1109/CASP.2016.7746170

Ranadive, A., Rizvi, S., & Daswani, N. M. (2013). *U.S. Patent No. 8,516,590*. Washington, DC: U.S. Patent and Trademark Office.

Medioni, G., Guy, G., Rom, H., & François, A. (1999). Real-time billboard substitution in a video stream. Paper presented at the *Multimedia,* 71-84.

Michel, O. (2004). Cyberbotics ltd. webots™: Professional mobile robot simulation. *International Journal of Advanced Robotic Systems, 1*(1), 5. https://doi.org/10.5772/5618

Moore, D., Currano, R., Strack, G. E., & Sirkin, D. (2019). The case for implicit external human-machine interfaces for autonomous vehicles. Paper presented at the *Proceedings of the 11th International Conference on Automotive User Interfaces and Interactive Vehicular Applications,* 295-307.

Mysore Jayakumar, P., B, T., v, V., Madivalappa, M., & Rudregowda, S. (2019). Detection of non-helmet riders and extraction of license plate number using yolo v2 and OCR method.*9*, 5167-5172. https://doi.org/10.35940/ijitee.B6527.129219

Patel, C., Patel, A., & Patel, D. (2012). Optical character recognition by open source OCR tool tesseract: A case study. *International Journal of Computer Applications, 55*, 50-56. https://doi.org/10.5120/8794-2784

Paul Inman. (2017). 8 OOH advertising stats you need to know. https://75media.co.uk/blog/outdoor-advertising-stats/

R. Smith. (2007). An overview of the tesseract OCR engine. Paper presented at the - *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), , 2* 629-633. https://doi.org/10.1109/ICDAR.2007.4376991

Rahmat, R., Dennis, D., Sitompul, O., Purnamawati, S., & Budiarto, R. (2019). Advertisement billboard detection and geotagging system with inductive transfer learning in deep convolutional neural network. *TELKOMNIKA (Telecommunication Computing Electronics and Control), 17*, 2659. https://doi.org/10.12928/telkomnika.v17i5.11276

Rawat, W., & Wang, Z. (2017). Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computation, 29*(9), 2352-2449. https://doi.org/10.1162/neco_a_00990

Rice, S. V., Jenkins, F. R., & Nartker, T. A. (1995). *The fourth annual test of OCR accuracy* (Vol. 3). Technical Report 95.

Visionsoutdoor. (2017 ). Billboard marketing statistics. http://www.visionsoutdoor.net/billboard-marketing-statistics/

W. Bieniecki, S. Grabowski, & W. Rozenberg. (2007). Image preprocessing for improving OCR accuracy. Paper presented at the - *2007 International Conference on Perspective Technologies and Methods in MEMS Design*, 75-80. https://doi.org/10.1109/MEMSTECH.2007.4283429

W. Lan, J. Dang, Y. Wang, & S. Wang. (2018). Pedestrian detection based on YOLO network model. paper presented at the - *2018 IEEE International Conference on Mechatronics and Automation (ICMA),* 1547-1551. https://doi.org/10.1109/ICMA.2018.8484698

Watve, A. K., & Sural, S. (2006). Detection of on-field advertisement billboards from soccer telecasts. https://doi.org/10.1049/cp:20060494

Yoon, H., Bae, Y. J., & Yang, Y. (2002). A soccer image sequence mosaicking and analysis method using line and advertisement board detection. *ETRI Journal, 24*(6), 443-454. https://doi.org/https://doi.org/10.4218/etrij.02.0102.0005

Zhang, F., Xu, X., & Qiao, Y. (2015). Deep classification of vehicle makers and models: The effectiveness of pre-training and data enhancement. Paper presented at the 231-236. https://doi.org/10.1109/ROBIO.2015.7418772

# Appendices

## Appendix 1-Project Proposal

### 1. Research Question, Problem Statement, or Topic for Investigation

Can simulator-based datasets replace real-life datasets in convolutional neural network for billboard detection?

"A study into the convolutional neural network object detection performance between an emulator dataset and a real-life dataset for billboards detection."

### 2. Initial/Mini Literature Review

With the global pandemic epidemic, there are significant restrictions on people travelling. (Chinazzi et al., 2020) As a result, it has caused great difficulties for the collection of some real-world data sets. For most people, it is difficult to obtain large datasets.

The role of datasets in CNN training has received increased attention across several disciplines in recent years. Image detection performance is influenced by the network structure itself and the size of the dataset. In particular, the size of the dataset has a significant impact on the detector performance. For the same CNN-based network, the experimental results of average accuracy show that the larger the training set, the higher the test accuracy. (C. Luo et al., 2018)

A simulator-based dataset may play a significant role in addressing the issue of the insufficient number of datasets. The purpose of this paper is to find out whether the simulator dataset can replace the real-world dataset when the dataset is insufficient?

**Billboard Detection**

Billboards as a mode of commercial media are very common in urban and highway scenes. However, billboards and their content are still managed in a primitive way. Therefore, we need a more efficient way to identify billboards for future maintenance as well as replacement. Much of the current literature on billboard detection pays particular attention to Sports T.V. They use a range of computer vision algorithms to detect the billboard on the image.

G. Cai et al. (2003) use Fast Hough transform, and text geometry features to extract advertisement from Sports T.V. image. The basic idea is to analyse the variation of the white dots on the lines. The upper border of the Sports T.V. billboard conveys much more white pixels than the other lines. The Fast Hough transform will be used to summarises the white dots on the line segment. The result shows a roughly 90% accuracy in identifying billboards on Sports T.V.

The Fast Hough transform is a feature extraction technique used in image processing and feature detection, such as in Computer Vision. It finds the lines' parameters through a voting mechanism (the extreme value of the accumulator matrix). (Li et al., 1986)

However, G. Cai et al.'s research does not consider Billboard colour and shape variations if the edge colour on the billboard is irregular, e.g., black and white, or a combination of different colours. The results of the research will vary considerably from expectations.

Another study that also used the Hough transform is by Watve. Instead of processing detection in the image, He detects advertisement billboard in soccer video using hue slicing and Hough Transform. The result shows that the approach achieves 90% accuracy in detecting advertisement billboards. (Watve & Sural, 2006).

Together, these studies provide important insights into the detection of billboards, however. There have been no large-scale studies to investigate the prevalence of billboards on roads and in urban environments. This is because, in previous work, occlusion and environmental complexity have been difficult aspects of billboard detection.

Unless the environment is uncomplicated, assuming an ordinary background and that objects are not obscured, edge and colour detection techniques can only be used for sports T.V. billboard detection because they have a uniform shape and a simple background environment.

Besides, previous works have relied on edges when detecting the shape of billboards. When one of the edges is covered by an obstacle, the detection will not detect the billboard. Furthermore, the colour detection method is only effective when the billboard has a specific colour and contrasts with the background colour. To overcome this problem, supervised machine learning methods such as CNN are needed to detect billboards in more complex environments, such as roads with many obstacles such as trees, pedestrians, vehicles, cables, utility poles, etc.

**Convolutional Neural Network**

In recent years, convolutional neural networks (CNNs) have been applied to research in computer vision fields such as detection, classification, and recognition as one of the power of CNNs is their ability to learn the features of objects.

Researchers have found that image classification accuracy can be improved using pre-trained neural network models with large learning capacities, such as YOLO, AlexNet, and GoogleNet. (Abdullah & M. S. Hasan, 2017)

In addition, deep convolutional neural networks (DCNN) have been shown to solve image classification problems for hard visual recognition tasks (Krizhevsky et al., 2017). Rahmat et al. (2019) use AlexNet's DCNN architecture with 60 million parameters, 650.000 neurons consisting of five convolutional layers, and three pooling layers to recognise different types of billboards. The results showed that the system achieved a training accuracy of 92.7% for billboard detection.

W. Lan et al. (2018) obtained the YOLO-R network model based on the network structure of YOLOv2. It achieved good results in improving the real-time frame rate of pedestrian detection.

It is experimentally shown that the algorithm improves the accuracy of pedestrian detection to 25 frames/s, which basically meets the requirement of real-time performance.

Mysore Jayakumar et al. (2019) have built a traffic violation system that automatically looks for a traffic violation of not wearing a helmet while riding a motorcycle and automatically extracting the vehicles' license plate number by using YOLO to detect the traffic violation and OCR to extract the plate number.

**Optical Character Recognition**

The application of OCR to the recognition of billboard content might be very useful. Simply put, people interested in the billboard content can automatically search the relevant website or product through their smartphone.

People can also quickly locate the location of a store by using some store billboard information. This will be a new direction for assisted location in the field of autopilot. Also, OCR applications can be extended to other areas, such as recognising road signs with text, which will extend the usefulness of intelligent systems for self-driving cars.

Optical character recognition (OCR) methods were first used to convert printed fonts into editable text. OCR is a pragmatic method in different applications such as digitalised maps, recognition of vehicle license plates (Patel et al., 2012), text readers for visually impaired people, understanding handwritten office forms (J. Memon et al., 2020), bank checks, etc.

**Webots**

The simulator dataset for this project will be collected from the robotics simulator-Webots. Webots is a free open source 3D robot simulator which widely used in education, research and industry. The software based on a combination of modern GUI (Qt), physics engine (ODE branch) and OpenGL 3.3 rendering engine. It also supports multi-platform as it can run on Windows, Linux and macOS.

In addition, Webots contains a large number of built-in modifiable robots such as two-wheeled tabletop robots, bipeds, industrial arms, modular robots, cars, drones, tracked robots, aerospace vehicles, and more. It also contains some sensors and real-world models such as road signs, and billboards.

We can simulate the operation of self-driving cars collecting data sets in the virtual simulator world through its camera sensor and controller code. The controller code supports multiple languages such as C, C++, python or MATLAB.

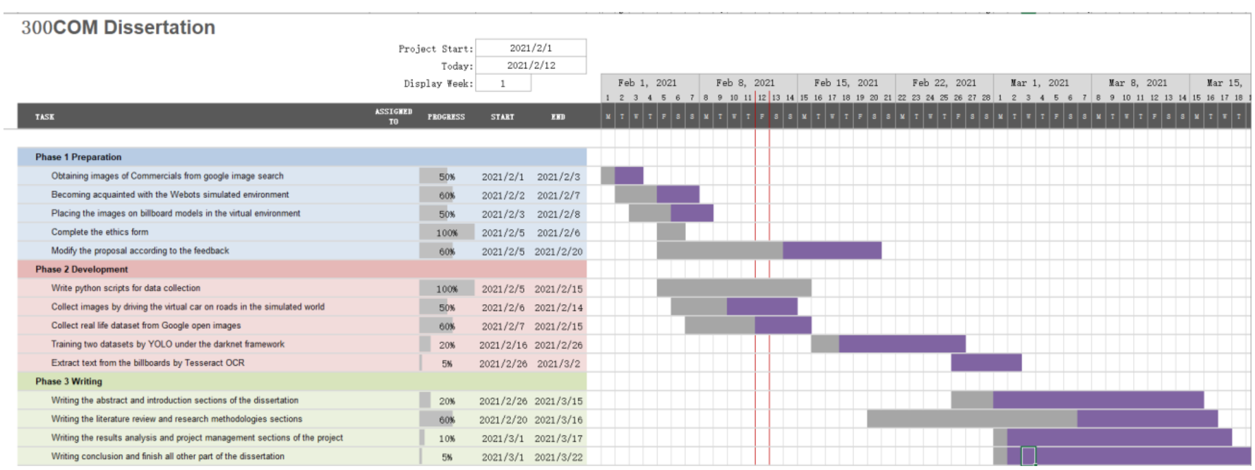
## 3. Client, Audience, or Motivation:

The motivation for this research comes from the fact that people are not able to travel freely to collect datasets due to the epidemic and there is a lack of corresponding datasets on the internet. Therefore, this project provides a method to collect datasets from the simulator world in case of insufficient datasets to explore whether simulator datasets can replace real datasets by compare their object detection performance.

In addition, the object of recognition in this study is billboards, because billboards and other objects on the road, such as license plates, road signs, etc., have some similarity in color and shape, once we can successfully recognise billboards, we can apply them to other research areas as well.

Besides, OCR can be applied after the object detection to extract the billboard's text, which will help the machine understand the billboard's content to make a segmentation within the billboard. Similarly, this research can be applied to other areas, such as detecting road signs with text and extracting the text from them to help autonomous driving intelligence systems analyse road conditions.

## 4. Primary Research Plan



## 5. Intended Project Outcome

This project aims to compare the object detection performance between the simulator datasets generated by a robot simulator and the real-life datasets and extend their application to Optical Character Recognition (OCR).

The question "Can simulator-based datasets replace real-life datasets in convolutional neural network base on billboard detection?" will be answered by comparing the difference in detection performance metrics between the real-life dataset and the simulator-generated dataset for the object.

The object detectors created in this project can be used in many different areas, such as recognising road signs and the recognition of text on road signs containing text. One possible use is the integration of the detector into the External Human-Machine Interface (eHMI) (Moore et al., 2019) of a car and its application to Advanced Assisted Driving Systems (ADAS), examples of which are already used by some advanced car manufacturers such as Tesla.

## Appendix 2-Records of supervisor meetings

| Project Supervision Meeting | | |
|---|---|---|
| Date | Conference content | Reaction |
| 21/1/2021 | The main task this week was to complete the ethical clearance form. This is a low risk project as the author's project does not involve too many ethical issues | Complete the ethics review form and submit it to the supervisor and module leader for review by the deadline. |
| 28/1/2021 | The project proposal was completed this week and submitted to the supervisor for review, who raised issues such as the lack of headings between chapters, jumps in some paragraphs without logical support and the fact that project time management could be better presented through Gantt charts. | Revise the Proposal according to the comments, adding subheadings to distinguish the various sections of the literature review. Added Gantt charts to make the project time management more professional. Trimmed some unnecessary paragraphs |
| 4/2/2021 | No big problems this week, supervisor shared a markdown document about the structure of the dissertation description as well as shared how to introduce critical think in the paper, a good website is the phrase bank at the University of Manchester. | Read the provided Mardown document carefully and apply it to your own articles. Go to the Manchester phrase bank website for examples of sentences and phrases to add more critial to the paper |
| 11/2/2021 | The problem this week is that the controller code does not control the driving of the car. Supervior has shared his code for reference as well as helping me with camera size issues | Went to the Webots website to find the relevant documentation and found that the problem was that the latest version of webot had changed the way the API was used and it worked fine after updating the method |
| 18/2/2021 | Still on the technical side, when collecting photos in the emulator world, the billboard was found to be too dark and the supervisor informed that it could be made brighter by turning on the backlight button. When the backlight was turned on, the photos were again too bright and caused distortion. | The supervisor informs that it is possible to go into the webots prototype file and change the intensity of light emission in the billboard prototype file. Changing the intensity to 3 is a suitable value, and the billboard then most closely resembles a real-world billboard |
| 25/2/2021 | The main question for this week's meeting was how to add rigour to the experimental results of this article, and the supervisor suggested that help could be sought from the school's Maths and Statics Centre which is sigma Maths and Stats Support and reminded to book early as there would probably be no available seats when the end of term was approaching | Go to the website of the Centre for Mathematics and Statistics to reserve an available space. Fortunately, they offer drop in sessions and I had direct access to one of the consultants and helped me to understand and define the Object detection measure |
| 4/3/2021 | Reported on the latest progress and asked the supervisor if there was a presentation? | Continue to write the results and analysis section. and ongoing tracking of issue results to the supervisor |
| 18/3/2021 | Schedule a one-on-one presentation with supervisor | Prepare slides to show the current progress of the project |
| 25/3/2021 | An one-to-one presentations with supervisor to explain the current progress of the project and the problems encountered | Get valuble Feedback as well as changes will be made based on the issues raised and reflections will be specifically recorded in the paper |
| 1/4/2021 | Lacking a rigorous statistical method to prove the difference between the two data sets, the supervisor recommends going to the Mathematical Statistics Center for help | I went to a total of 3 times at the Math and Statistics Center and were initially given the option of using a t-test but were unable to derive a standard variance for our data and therefore could not use the t-test.On the last visit, the statistician informed me that I could use the Z-test. this solved the problem |
| 8/4/2021 | The code in the controller and OCR sections is replaced by pseudo-code or flowcharts, and I wasn't sure how to make them so I asked the supervisor for advice. | Through the tutor's explanation, I finally decided to use flowcharts because the source code is written in python which is already very close to natural language. |

**Appendix 3-GitHub Repo**

https://github.coventry.ac.uk/wangh109/300COM-Final-Project

**Appendix 4-Controller Code**

```
import sys, os
import math
import itertools
import json
import time
from vehicle import Driver


sensorMax = 1000

driver = Driver()


for i in range(driver.getNumberOfDevices()):
    device = driver.getDeviceByIndex(i)
    print(i,device.getName(),type(device))


refresh = 50

printCounter=0

camera_enabled=False

#camera
camera_name= "camera"
camera_finder = driver.getDevice(camera_name)
camera_finder.enable(refresh)

if camera_finder.hasRecognition():
    camera_finder.recognitionEnable(refresh)
    camera_enabled= True

#ineritial
inertial_name = "inertial unit"
inertial = driver.getDevice(inertial_name)
inertial.enable(refresh)

targets = ( ( -60,  0,  88), \
```

```
        )
targetId = 0

folder = "images"
annotations_folder="annotations"
maxImages = 50

def local_point( point, origin, heading ):
    relative = [ p-o for p, o in zip(point,origin) ]
    return relative[0]*math.cos(-heading) + relative[2]*math.sin(-heading), \
           relative[1], \
           relative[2]*math.cos(-heading) - relative[0]*math.sin(-heading)


def save_annotated_image( camera_finder, position, heading ):
    filename = str(int(time.time()*1000))

    camera_finder.saveImage(os.path.join(folder,filename+".png"),100)

    positions = ( ( *object.get_position_on_image(), *object.get_size_on_image() )

                for object in camera_finder.getRecognitionObjects()
                if "advertising board" in object.get_model().decode())

    filtered = ( (x, y, w, h) for x, y, w, h in positions
                if y<=510 and x>10 and x <camera_finder.getWidth()-10)

    for object in camera_finder.getRecognitionObjects():
        print(object.get_model().decode())



    width = camera_finder.getWidth()
    height = camera_finder.getHeight()
    scaled = ( (x/width, y/height, w/width, h/height) for x, y, w, h in filtered )


    with open(os.path.join("D:\\HOME\\Informatics\\controllers\\test00",folder,filename+".txt" ),
"w" ) as f:
        for object in scaled:
```

```
        print( "0 {} {} {} {}".format( *object ), file=f )

while driver.step() != -1:

    # get current position
    position = driver.getSelf().getPosition()
    heading = inertial.getRollPitchYaw()[2]
    if heading == float('nan'): continue

    # get next target
    target = targets[targetId% len(targets)]
    localTarget = local_point( target, position, heading )

    print( "target> {:.3f} {:.3f} {:.3f}".format(*target) )

    print( "local > {:.3f} {:.3f} {:.3f}".format(*localTarget) )

    localHeading = math.atan2( localTarget[0], localTarget[2] )

    while localHeading > math.pi: localHeading -= math.pi*2
    while localHeading < -math.pi: localHeading += math.pi*2

    print( "head  > {:.3f}".format(math.degrees(localHeading)) )

    localHeading = math.atan2(localTarget[0],localTarget[2])

    distance = math.sqrt( abs(localTarget[0]**2) + abs(localTarget[2]**2) )
    if distance <1:
        targetId +=1

    print( "dist  > {:.3f}".format(distance) )


    driver.setSteeringAngle( -localHeading )
    driver.setCruisingSpeed( 6 )

    save_annotated_image(camera_finder,position, heading )

sys.exit()
```

**Appendix 5-List of Figures**

## Appendix 5- Performance Metrics of Two Datasets

### The real-life dataset performance results.

```
[yolo] params: iou loss: ciou (4), iou_norm: 0.07, obj_norm: 1.00, cls_norm: 1.00, delta_norm: 1.00, scale_x_y: 1.10
nms_kind: greedynms (1), beta = 0.600000
 151 route  147                                    ->   38 x  38 x 256
 152 conv    512       3 x 3/ 2     38 x  38 x 256 ->   19 x  19 x 512 0.852 BF
 153 route  152 116                                ->   19 x  19 x1024
 154 conv    512       1 x 1/ 1     19 x  19 x1024 ->   19 x  19 x 512 0.379 BF
 155 conv   1024       3 x 3/ 1     19 x  19 x 512 ->   19 x  19 x1024 3.407 BF
 156 conv    512       1 x 1/ 1     19 x  19 x1024 ->   19 x  19 x 512 0.379 BF
 157 conv   1024       3 x 3/ 1     19 x  19 x 512 ->   19 x  19 x1024 3.407 BF
 158 conv    512       1 x 1/ 1     19 x  19 x1024 ->   19 x  19 x 512 0.379 BF
 159 conv   1024       3 x 3/ 1     19 x  19 x 512 ->   19 x  19 x1024 3.407 BF
 160 conv     18       1 x 1/ 1     19 x  19 x1024 ->   19 x  19 x  18 0.013 BF
 161 yolo
[yolo] params: iou loss: ciou (4), iou_norm: 0.07, obj_norm: 1.00, cls_norm: 1.00, delta_norm: 1.00, scale_x_y: 1.05
nms_kind: greedynms (1), beta = 0.600000
Total BFLOPS 127.232
avg_outputs = 1046213
 Allocate additional workspace_size = 6.65 MB
Loading weights from /mydrive/yolov44/backup/yolov4-obj_3000.weights...
 seen 64, trained: 192 K-images (3 Kilo-batches_64)
Done! Loaded 162 layers from weights-file

 calculation mAP (mean average precision)...
 Detection layer: 139 - type = 28
 Detection layer: 150 - type = 28
 Detection layer: 161 - type = 28
1000
 detections_count = 9751, unique_truth_count = 2425
class_id = 0, name = Billboard, ap = 92.16%       (TP = 2205, FP = 581)

 for conf_thresh = 0.25, precision = 0.79, recall = 0.91, F1-score = 0.85
 for conf_thresh = 0.25, TP = 2205, FP = 581, FN = 220, average IoU = 64.28 %

 IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
 mean average precision (mAP@0.50) = 0.921602, or 92.16 %
Total Detection Time: 121 Seconds

Set -points flag:
 `-points 101` for MS COCO
 `-points 11` for PascalVOC 2007 (uncomment `difficult` in voc.data)
 `-points 0` (AUC) for ImageNet, PascalVOC 2010-2012, your custom dataset
```

**The simulated dataset performance results.**

```
 148 conv    512       3 x 3/ 1     38 x  38 x 256 ->   38 x  38 x 512 3.407 BF
 149 conv     18       1 x 1/ 1     38 x  38 x 512 ->   38 x  38 x  18 0.027 BF
 150 yolo
[yolo] params: iou loss: ciou (4), iou_norm: 0.07, obj_norm: 1.00, cls_norm: 1.00, delta_norm: 1.00, scale_x_y: 1.10
nms_kind: greedynms (1), beta = 0.600000
 151 route  147                                    ->   38 x  38 x 256
 152 conv    512       3 x 3/ 2     38 x  38 x 256 ->   19 x  19 x 512 0.852 BF
 153 route  152 116                                ->   19 x  19 x1024
 154 conv    512       1 x 1/ 1     19 x  19 x1024 ->   19 x  19 x 512 0.379 BF
 155 conv   1024       3 x 3/ 1     19 x  19 x 512 ->   19 x  19 x1024 3.407 BF
 156 conv    512       1 x 1/ 1     19 x  19 x1024 ->   19 x  19 x 512 0.379 BF
 157 conv   1024       3 x 3/ 1     19 x  19 x 512 ->   19 x  19 x1024 3.407 BF
 158 conv    512       1 x 1/ 1     19 x  19 x1024 ->   19 x  19 x 512 0.379 BF
 159 conv   1024       3 x 3/ 1     19 x  19 x 512 ->   19 x  19 x1024 3.407 BF
 160 conv     18       1 x 1/ 1     19 x  19 x1024 ->   19 x  19 x  18 0.013 BF
 161 yolo
[yolo] params: iou loss: ciou (4), iou_norm: 0.07, obj_norm: 1.00, cls_norm: 1.00, delta_norm: 1.00, scale_x_y: 1.05
nms_kind: greedynms (1), beta = 0.600000
Total BFLOPS 127.232
avg_outputs = 1046213
 Allocate additional workspace_size = 52.43 MB
Loading weights from /mydrive/yolov4/backup/yolov4-obj_3000.weights...
 seen 64, trained: 192 K-images (3 Kilo-batches_64)
Done! Loaded 162 layers from weights-file

 calculation mAP (mean average precision)...
 Detection layer: 139 - type = 28
 Detection layer: 150 - type = 28
 Detection layer: 161 - type = 28
1000
 detections_count = 2413, unique_truth_count = 2425
class_id = 0, name = Billboard, ap = 50.88%      (TP = 1036, FP = 86)

 for conf_thresh = 0.25, precision = 0.92, recall = 0.43, F1-score = 0.58
 for conf_thresh = 0.25, TP = 1036, FP = 86, FN = 1389, average IoU = 83.28 %

 IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
 mean average precision (mAP@0.50) = 0.508848, or 50.88 %
Total Detection Time: 34 Seconds

Set -points flag:
 `-points 101` for MS COCO
 `-points 11` for PascalVOC 2007 (uncomment `difficult` in voc.data)
 `-points 0` (AUC) for ImageNet, PascalVOC 2010-2012, your custom dataset
```