DOCTORAL THESIS

SAFE TRAINING OF TRAFFIC ASSISTANTS FOR DETECTION OF DANGEROUS ACCIDENTS

A Thesis submitted to Brunel University
in accordance with the requirements
for award of the degree of Doctor of Philosophy
in
Department of Electronic and Electronic Engineering

Yifan Li

March 31, 2023

Abstract

As the automotive industry continues to develop, the car's function is no longer limited to a simple means of transportation. Instead, it is more of a technological product that combines safe mobility, entertainment and safe driving technology. Furthermore, ensuring the safety of passengers is a critical element in the development of cars. Thus, safety-based autonomous driving and assisted driving systems are essential in developing cars and future development strategies.

However, the limitations of the current radar systems used in automobiles are widespread. A single radar detection makes it difficult to carry out accurate object detection and identification and can only provide vague conceptual feedback. The radar feedback needs to be more accurate, especially when the distance is too close or too far. Any object that reflects radar waves is being used as a hazard warning.

Due to the continuous development of information technology and artificial intelligence technology, integrating artificial intelligence into traditional industries to achieve automation and intellectual development is the main direction of current technological development and industry progress. For example, the application of AI technology to the automotive industry enables comprehensive and immediate environmental awareness, comprehensive and accurate planning and decision-making, and precise and efficient vehicle control to ensure the safety of passengers.

In this thesis we propose the use of the YOLO algorithm in Virtual Worlds to safely train the car's recognition to detect dangerous traffic accident situations in different environments without damage to property and danger to human well-being through real-time video detection by obtaining more accurate information about obstacles or hazards. The YOLO series of Artificial Intelligent (AI) detection algorithms are used to detect objects through video or pictures. Unlike radar detection, YOLO can accurately analyse obstacles. Assisted driving and autonomous driving will be an essential part of the future of transportation, but training them for object detection and recognition of dangerous traffic situations, which is a key aspect of its operation, is difficult because of the damage to property and human well-being. Therefore performing this training in virtual world is essential.

First, we designed and built a virtual 3D city platform using the Unity 3D engine, recreated as much realistic road information as possible in the 3D city. Then we used the YOLOV5 algorithm for detection of objects to obtain accurate virtual identification information successfully. After training, YOLOV5 can detect all vehicles and obstacles on the virtual road. From there, it can alert the driver of dangerous traffic situations accordingly instead of alerting for all objects to avoid unnecessary danger warnings.

On the other hand, environmental perception is essential to safe driving. Nevertheless, current research has seen various technologies applied to environmental perception, such as Microsoft's AIRSIM autonomous driving simulator, LIDAR technology and millimetre wave radar technology, which are currently heavily used. However, technology is constantly evolving, and LIDAR and millimetre wave radar are now at the forefront of environmental awareness. Accurate one-stage algorithms and databases are an important direction for the future. This is because such algorithms not only indicate the presence of an object in front of them but also identify exactly what type of object the output is(people, pets, ground obstacles, etc.). We have built on the Yolo algorithm and applied it to assisted driving with a focus on safely training the AI to detect the driver's blind spot, by analysing the environment out of the driver's view and giving timely feedback.

This thesis explores in depth the application of how to safely train YOLO for assisted driving, building a 3D virtual city and testing it in different stages in a virtual environment. The usefulness of the YOLO algorithm for driving car safety is verified. Through the continuous training of the YOLO algorithm, an extensive database can give the driver more results in terms of environmental perception. As a result, the occurrence of traffic accidents due to insufficient environmental perception for training YOLO can be increased by constructing virtual accidents without damage to property and human well-being.

Declaration of Authorship

I, Yifan Li, declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work.

Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: .. Yifan Li DATE: ...18/01/2024.......

(Signature of student)

Contents

List of Figures

Chapter 1 Introduction

London’s road network faces many challenges over the next 20 years: a rising population expected to exceed 10 million by 2031, the need for significant investment in road infrastructure, rising aspirations for high quality public space, conflicts between competing road users and the imperative to improve road safety. Now is the time to ensure that the right systems and procedures are in place to maximise the effectiveness of the road network in light of London’s growing population.(Katherine 2014)

The most basic requirement of the road transport industry is safety and security, but there remain a number of challenges we must overcome. Every year, between 1.25 and 1.5 million people die on the roads around the world. (Matthias Maedg 2019)One of the crash types involved in these fatalities are blind spot crashes between trucks and bicyclists. (ReinierJansen, Silvia Varotto 2022).

A driver’s field of view is an essential requirement for decreasing traffic crashes and increasing safety (Mohammed beida 2022). The development of the automobile has always been inseparable from traffic safety, and with the increasing speed of cars, traffic safety is also an issue that the government should address. LKA (Lane-Keeping Assist), FCW (Forward-Collision Warning), and other similar safety systems are also used on most cars based on this and other safety measures.

1.1 Background

There are many causes of traffic accidents, of which speeding and drivers' blind spots are two main factors. Speeding is mostly caused by drivers driving aggressively, while blind spots are unavoidable when driving a car. Blind spots can be identified and detected by AI algorithms to reduce the number of accidents caused by blind spots.

According to statistics, approximately 1,000,000 accidents occur worldwide each year as a result of vehicles on the rear side being in the driver's blind spot and thus the vehicle in front changing lanes. With these facts in mind, a system that detects neighbouring vehicles entering the rear blind spot and warns the driver of a lane change would prevent and reduce the number of accidents that could occur (Chen 2011). There are several areas around the car that the driver cannot see, these areas are called Blind spot (Verhaevert 2017). The area to the left and right of the rear door of a vehicle is the most likely blind spot for drivers when driving a vehicle (Zhang Rong 2009).

The blind spots are even worse for large trucks; according to the OTS database, about 76% of accidents are caused by blind spots. he reason is the trucks have much more prolonged and taller than ordinary cars. As a result, the driver's blind spot can be even worse. (Russell, D. 2009)

1.2 Flat-Screen Display (HUD) Technology

In the movie "Mission Impossible 4", a scene recently appeared: Tom Cruise drove the BMW i8 concept car to the headquarters (Figure 1.1). The scene of using the touch screen on the windshield is the future screen interaction technology. A new direction of development. This technology has also been realized in real world. We call it an augmented reality windshield. "Augmented reality" technology, also known as AR, is a technology that integrates virtual information generated by a computer into the actual environment that users want to experience. Its purpose is to supplement the real-world information.

Figure 1.1 Mission impossible

Today's automotive technology is not as sophisticated as in the movies, but similar technology is already available - flat-screen displays (HUD). More and more cars have been equipped with flat-screen display(HUD) technology, which displays current information about the car in the form of text or graphs projected onto the car's front windscreen. This enables the driver to understand information about the vehicle, such as driving speed, even when looking straight ahead (Tonnis, Sandor, Klinker, Lange, and Bubb, 2005).

Figure 1.2 illustrates the use of flat-screen display (HUD) technology in today's cars, displaying basic information such as the car's current speed, the time of day, and the remaining battery level of the tram. This technology allows the driver to focus on the road ahead by reducing the number of times the driver has to look down to see information about the vehicle while driving.

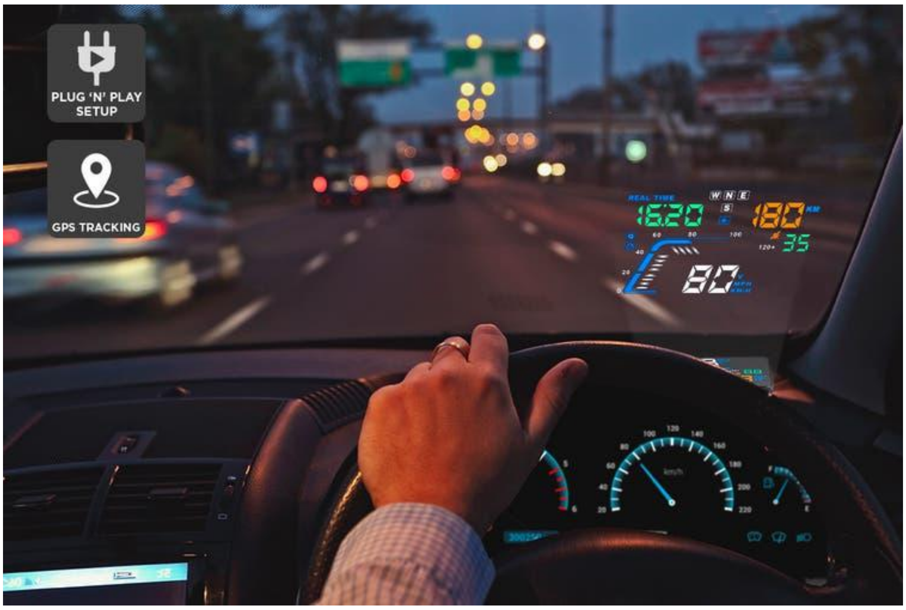


Figure 1.2 Flat-screen display(HUD) From google image

1.3 Virtual environment building

This virtual 3D city environment based on Unity includes buildings, roads, traffic lights, busy intersections and moving vehicles. Simulation scenarios allow for a multi-faceted analysis of the accident.(Jingwen,2021) The virtual environment provides an extensive database of samples and allows for the simulation of various accidents.(Leudet,Christophe,2018) Figure 1.3 virtual 3D city preview and Figure 1.4 virtual 3D city detail In this environment, we have recreated a large number of traffic accidents, simulating traffic and accident-prone scenarios.

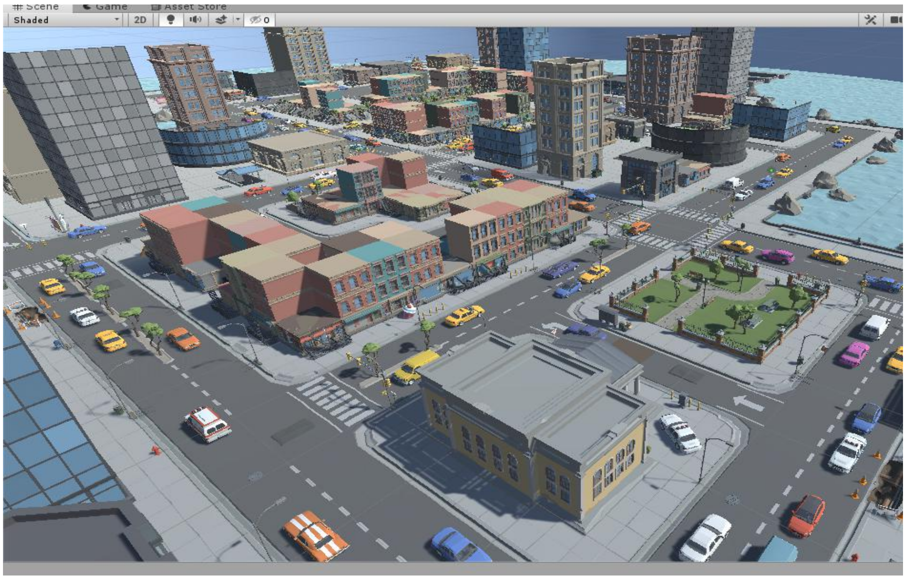


Figure 1.3 Virtual 3D city preview

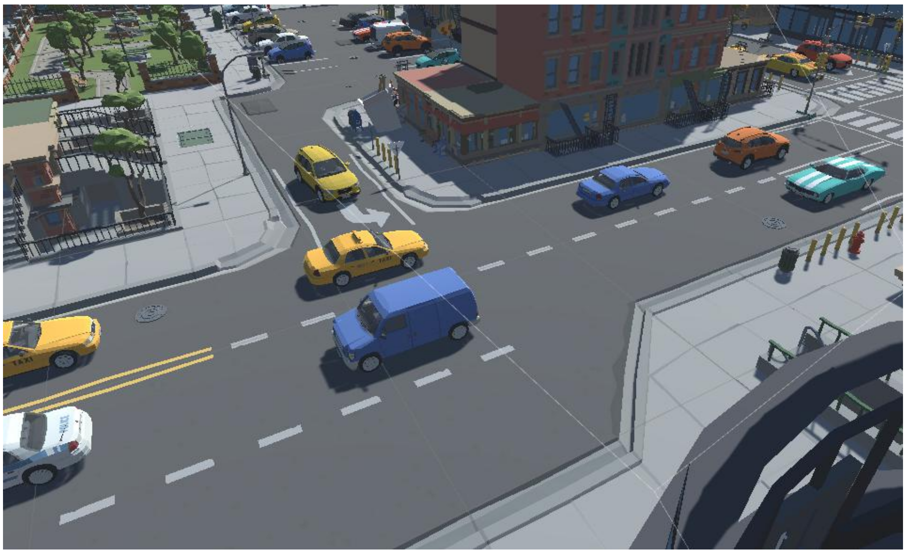


Figure 1.4 Virtual 3D city detail

1.4 The PhD Aim and Objective

The main research objective of this study is to safely train a Yolo AI to identify objects in the blind zone of the car driver without damage to property or injury to persons using object detection algorithms in a virtual scene. In order to improve the driver's ability to perceive the environment outside the car and reduce the probability of traffic and accidents due to blind zones.

The main objectives of this research are:

1. Analysis and statistics on the leading causes of current traffic to find the main factors that trigger traffic accidents.

2. Introduction and analysis of measures to prevent/avoid motor vehicle traffic accidents and literature review.

3. Set up an experimental virtual city scenario to simulate traffic accidents and collect data.

4. Data analysis for image detection and test.

5. Algorithm analysis and application of driver blind spot video detection.

6. Database creation and study of AI training algorithms.

1.5 Research Contributions

This research project aims to reduce the number of traffic accidents caused by blind spots in cars. The YOLO detection algorithm is used to safely train a Yolo AI system for blind zone detection and recognition without damage to property or injury to persons. Detection algorithms enhances the driver's perception of the environment outside the car. In addition

1. The project design and builds a sizeable 3D model city, which can work for diverse

traffic safety tests, including complex traffic networks, different types of road information and moving vehicles.

2. Collection of information on traffic accidents. Statistics of traffic accident samples from recent years are analysed and summarised. Finding the leading causes of traffic accidents and planning reasonable solutions are presented.

3. We summarise the leading causes of traffic accidents and reproducing these accidents in a virtual city without damage to property or injury to persons. Proposing new solutions — Identification of the driver's blind spot using object detection algorithms to enhance the driver's perception of the car's surroundings.

4. Conduct an in-depth study of the YOLO algorithm and analyse the implementation process of the YOLO algorithm for target detection. Compare and summarise the advantages and disadvantages of different versions of the YOLO algorithm

5. Train the YOLO detection algorithm and then use the YOLO algorithm to detect the scene in the virtual environment and successfully get the detection results of all the objects in the scene. This verifies the feasibility of the YOLO algorithm in automotive blind spot monitoring.

1.6 The Research Scope and Thesis Structure

1.6.1 Research Scope

There are many causes of traffic accidents, and this thesis focuses on traffic accidents caused by driver blind spots.

Yolo is divided into various versions of recognition. This thesis focuses on the algorithm recognition of the Yolo V5 version and the environmental perception analysis of the driver's blind spot.

1. This thesis does not consider factors caused by the driver's factors (drunk driving or aggressive driving) or visibility.

2. Only the driver's blind spot around the car is analysed and discussed, with a small range of objects identified and analysed and propose solutions.

1.6.2 Thesis Structure

This research project aims to reduce the number of traffic accidents. With the development of modern transportation, fast travelling humans need to give more consideration to safety issues. This thesis focuses on the leading causes of traffic accidents, recreating them in virtual scenarios and using Ai recognition algorithms to address the identification of potential hazards.

The thesis is divided into several chapters, each focusing on a critical topic. Finally, all chapters are combined and summarised in the conclusion. The second chapter contains extensive literature reading and citations on the causes of traffic accidents, known solutions, and an analysis of innovative solutions.

Chapter 3 focuses on constructing a 3D virtual scenario that will be used for data collection and simulation experiments, and illustrates the differences and benefits of testing in a virtual scenario versus testing in the field.

Chapter 4 contains data analysis and data collection on the driver's blind spot, recording scenarios that are prone to traffic accidents on real roads. The scenarios are then recreated in a virtual city and video recorded.

In Chapter 5, we compare several currently popular detection algorithms and finally identify YOLO as the algorithm for our project requirements. The discussion of YOLO is also developed, and the composition of the YOLO algorithm is explained.

Chapter 6 is an extension of the YOLO algorithm and the presentation of the final detection results. This chapter describes configuring the YOLO environment, analysing common errors and providing solutions.

The final chapter is Chapter 7, which concludes our project and this article with a summary of our future work.

Chapter 2 Literature Review

2.1 Introduction

In this chapter literature review, there are four parts. The first part 2.2 is the analysis of traffic accidents and statistics on the leading contributing causes of traffic accidents. The final results are in line with the case study. In section 2.3, we focus on most leading causes of traffic accidents, driver's perception of the environment. Section 2.4 is an extension of section 2.3, which explains in detail the causes of blind spots and discusses the dangers of blind spots and the effects of blind spots on drivers. Firstly, in section 2.4, the current solutions to blind spots in cars are presented, and their practical effects are reviewed. Various solutions are described in detail, including additional mirrors and warnings for large vehicles and the use of radar in cars (millimetre wave radar and LIDAR). A review of science fiction films summarises human depictions of the future of the car and tries to find ways to implement them today. An attempt is made to find new solutions in the context of AI algorithm recognition and car blindness. 2.5 section received inspiration from the depiction of automotive technology in a future movie, combined with today's rapidly evolving artificial intelligence algorithms, to use the CNN family of algorithms for the detection of automotive blind spots. 2.6 Section reviews the literature on detection algorithms, describing how they differ from traditional solutions. Convolutional neural networks are introduced in advance of the in-depth study of recognition algorithms that follows in chapter 4. The section 2.7 is summary of Literature review chapter.

Figure 2.1 Driving on motorway from Google

2.2 Analyze Traffic Accidents

Even with the advances in Vehicle technology, rising standards of Vehicle safety and the increasing attention paid by the government to traffic safety, the incidence of road traffic accidents is increasing yearly (Ryder, Gahr, Egolf, Dahlinger, Wortmann, 2017). Unexpectedly, traffic accidents occur not because drivers need to learn the rules but more often due to inattention, rash decisions and overconfidence (Srinivasa, Roy, Jagdish, and Minocha, 2004). Crashes caused by lane changes are partly due to the driver's blind spot at the rear of the car (Isaksson-Hellman, & Lindman, 2018).

The increasing popularity of cars leads to more traffic congestion and frequent traffic accidents. However, the traffic safety problem will be more severe, and we have had to revisit the transport issues to find more convenient solutions.

Figure 2.2 Traffic accident from google

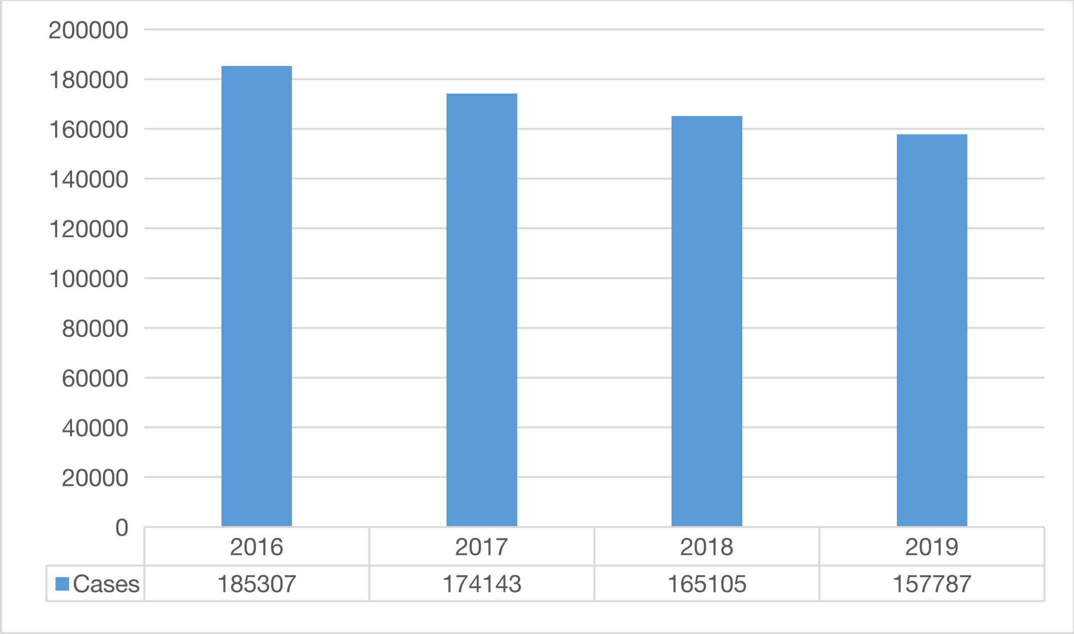
2.2.1 Background of Traffic Accidents

Terrorist attacks concern most people, but the number of people injured or killed in traffic accidents yearly is far higher than the number of casualties by terrorist attacks. More than one million people are killed in road accidents each year, and tens of millions suffer injuries of varying degrees of severity as a result of road accidents (Tanaboriboon, and Satiennam,, 2005).

Thousands of people are currently injured or killed due to traffic problems each year, and they may be walking, riding, driving and crossing the road alive. They could be working-class people busy going to work, children playing in the street, or loved ones making the long journey home. In the event of a road accident, they leave behind a broken family and community. Every year, millions of people spend long weeks or even months in hospital because of road accidents (Singh, Sahni, Bilquees, Khan, and Haq, 2016). Road traffic accidents are one of the leading causes of death among young people and the eighth leading cause of death worldwide, with around 1.24 million people dying in road accidents yearly. Most of these deaths are preventable (Serrano2022).

There were 23,139 reported collisions in London in 2021, resulting in 75 people being killed, 3,505 being seriously injured and 23,092 being slightly injured. This report provides a summary of personal injury road traffic collisions and casualties, reported to and by the police, in Greater London in 2021 (transport London 2021). The early injuries sustained in road traffic accidents cause significant financial losses to individuals, families and the nation as a whole. These losses come from the cost of treatment, as well as the lost productivity of those who die or are disabled as a result of their injuries, and family members who need to take time off work or attend school to care for the injured. Road traffic accidents cost most countries up to 3% of their GDP (Global status report 2018). In comparison to the overall caseload, 15-19-year-olds will have a greater risk of dying in a road traffic accident, almost twice as much as the general population average, 82.5 deaths per million (Gov Transport, 2022). A significant challenge for urban traffic safety and the transport network is the increasing number of traffic accidents in recent years, which are related to the population's health and add an economic burden to society and the government (Kumar, Toshniwal,and Parida,2017).

Figure 2.3 This is number of car drivers involved in reported road accidents in Great Britain from 2016 to 2019 (Google scholar 2020).



	2016	2017	2018	2019
■ Cases	185307	174143	165105	157787

Figure 2.3 Drivers involved in reported road accidents

From table 2.2.1 shows that the number of traffic accidents involving drivers has decreased yearly but has yet to improve significantly. In 2019, more than 150,000 traffic accidents were directly related to drivers. Based on the above data, the negative impact of traffic accidents is much higher than we expected.

China's traffic fatality rate of 22 per 100,000 people is slightly higher than the global average of 18, ranking it 44th out of 193 countries studied. According to a recent report by the University of Michigan Transportation Research Institute, the regions with the highest traffic fatality rates are Africa and Latin America. Namibia has the highest traffic fatality rate in the world, with 45 deaths per 100,000 people (Mich-statistics 2014).

According to Figure 2.4, The Causes of Traffic of Fatalities (Rac,2021), vehicular occupants account for 47% of all traffic fatalities, nearly half of the total. On the other hand, pedestrians and motorcyclist account for 25% and 21% of accidents. Pedal Cyclists for the least at around 7%. Since vehicle occupant fatailities are more numerous than others, we need to pay more attention to vehicle traffic safety.

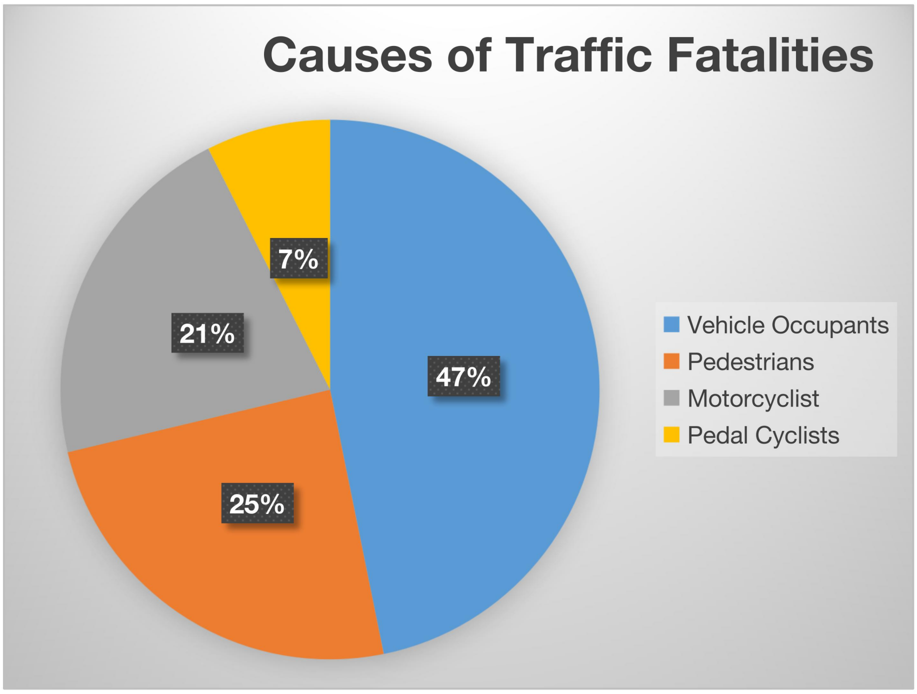


Figure 2.4 Cases of traffic fatalities (Rac Office 2021)

Our awareness of traffic accidents and safety is insignificant compared to the harm caused by traffic accidents. With such a considerable amount of data, we are forced to re-examine the issue of traffic safety today, and improving it is one of the critical directions for future social development.

2.2.2 Different Reason with Traffic Accidents

Ten of the most common causes of vehicle accidents include: [1] Distracted driving. [2]

Drunk and drugged driving. [3] Severe weather. [4] Reckless driving and road rage. [5] Speeding. [6] Running red and yellow lights. [7] Running stop signs. [8] Improper turns. [9] Road hazards. [10] Driver fatigue (Levin. G 2022).

Over a third of all road accidents in the UK are caused by failing to look properly. It is the most common cause of all UK road accidents, yet it should be the easiest to prevent. However, there are many reasons why it can happen.

If you have ever driven on a UK motorway, you will be familiar with the phrase ‘Tiredness kills’. This is because tiredness can have a dramatic impact on a

Driver’s awareness and reactions. Crashes caused by driver fatigue are about 50% more likely to result in death or severe injury as they tend to be high-speed impacts because a driver who has fallen asleep cannot brake or swerve to avoid or reduce the impact. Alternatively, failing to look properly before other forms of distraction whilst driving could cause maneuvering, perhaps by a fellow passenger, a mobile device or the radio. Some accidents are simply caused by driver complacency, familiarity with the route, or even laziness. Driving requires the utmost concentration. (Jonathan.2020)

The National Highway Traffic Safety Administration reports that approximately 60,000 crashes occur each year due to a driver's lack of concentration due to overexertion or special causes of drowsiness or fatigue (Kaplan, Guvensan, Yavuz, & Karalurt, 2015).

Heavy and oversized vehicles are much more likely to be involved in accidents than ordinary civilian vehicles travelling on the road. (Krishnan, Sheel, Viswanadh, Shetty, and Seema, 2018)

In terrible weather conditions, drivers usually adjust their driving behaviour, such as slowing down, avoiding overtaking, increasing the distance to the vehicle in front, turning on unique lights, sounding the horn, etc.. However, these safe driving behaviours are not worth mentioning compared to lousy weather. Severe weather tends to cause traffic accidents, the main reason which is low visibility (Hammad, Ashraf, Abbas, Bakhat, Qaisrani, Mubeen, Fahad,and Awais,2019). Distracted drivers, or drivers who trust their judgement too much, are the leading cause of most road accidents. The majority of accidents caused by low concentration of drivers are less severe vehicle occupants cuts or minor injuries, and almost all drivers have been involved in such minor accidents (Pecherková and Nagy, 2017).

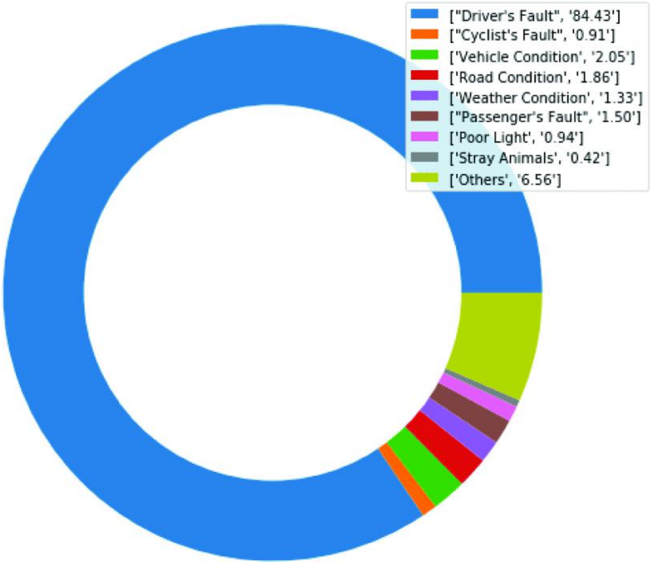


Figure 2.5 Percentage of leading causes of traffic accidents

From Figure 2.5 The careless, inattentive and reckless behaviour of drivers causes more than 80% of traffic and accidents. (Krishnan, Sheel, Viswanadh, Shetty and Seema, 2018).

Poor driver judgement or bad decision-making was the leading cause of most teenage crashes. Over half of all crashes are caused by driver’s driving blind spot due to inaccurate perception of the environment or lack of information about the environment in the driver's blind spot (Curry, Hafetz, Kallan, Winston and Durbin, 2011). The most common cause of car accidents in Great Britain is the driver (or motorcycle rider) failing to look properly — this factor contributes to 37.8% of car accidents. (Yurday 2022). Insufficient careful observation, misjudgment of road surface information and misunderstanding of unknown environments are the leading causes of road user (driver) crashes, of which this category accounts for 72% of overall crashes. Minimal differences in the road surface and environmental information can lead to serious traffic accidents (Thomas, Morris, Talbot and Fagerlind, 2013).

According to the data and conclusions reviewed in the literature, the leading cause of most road traffic safety accidents is the improper operation of individual drivers. The main reason for this is a lack of concentration, comprehensive observation of the car's surroundings and incorrect subjective judgement of the dangers, ultimately leading to traffic accidents. Therefore, comprehensive environmental perception plays a vital role in safe driving.

2.3 Driver Blind Spot

In the previous section, we have analysed and summarised traffic accidents based on literature reading and research. This subsection focuses on the Blind Spot, which affects safe driving.

2.3.1 Explain Driver's Blind Spot

The Vehicle blind spot is an area where the driver's view of the road is blocked by the structure of the vehicle and is not visible from the driver's point of view. Blind spots are notoriously difficult for most drivers to improve. On average, drivers make lane changes every 2.76 miles during their daily commute, which increases significantly during traffic jams and peak commuting times. In addition, there are approximately 630,000 crashes annually in the US due to drivers being unable to see the vehicle behind them changing lanes in their blind spots (Hester, Lavalliere, Laurendeau, Simoneau, and Teasdale, 2011). A blind spot is the area of the road that can't be seen by looking forward through your windscreen, or by using your rear-view and side-view mirrors. Blind spots can be large enough in size to easily block another car, motorbike, cyclist or pedestrian from your view. (AA driving school 2021). Blind spots arise from two factors. First, rear-view and side mirrors reflect a limited field of view defined by their size, shape, and curvature. Anything outside that field of view is, in effect, invisible to the driver looking in the mirror. Second, a vehicle’s components (the vehicle frame, roof, hood, trunk, or trailer) and contents (passengers, seats, and cargo) can block a driver’s view, blinding the driver to whatever is behind those components or contents.

It is worth noting that blind spots in vehicles can affect the safety of drivers and passengers on the road, with drivers unable to detect information about their environment within the blind spot, including vehicles, pedestrians, cyclists or road obstacles (Hughey 2021). The presence of pedestrians or cyclists in the driver's blind spot is one of the leading causes of crashes, and it is unpredictable for most drivers. (Saito, Sugaya, Inoue, Raksincharoensak, and Inoue, 2021).

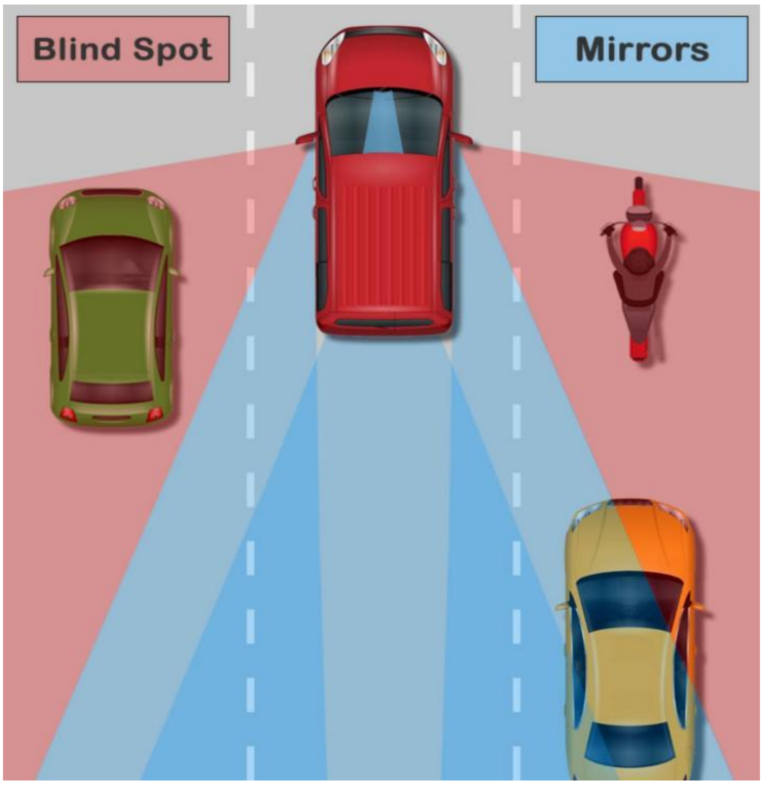


Figure 2.6 Normal Vehicle Driver Blind Spot

The Figure 2.6 Shows the blind spot range of an average vehicle (rear side). The blue area is where the driver can see through the vehicle's rearview mirror, while the red area is where the driver cannot see.

In addition, there are blind spots not only at the rear of the vehicle but also in the front due to the vehicle's structure. The A-pillar obscures the driver's side view in front of him. In Figure 2.7 the Vehicle's A-pillar shows the blind spot in front of the driver.

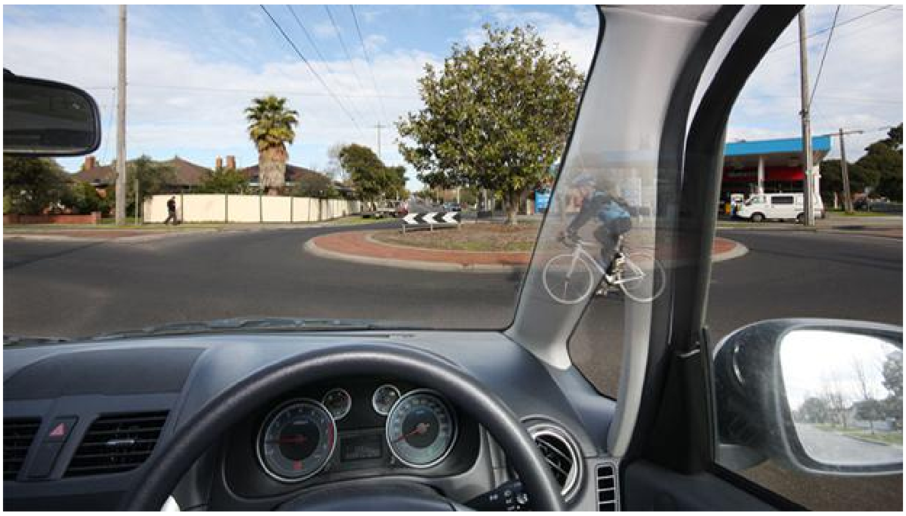


Figure 2.7 Vehicle's A-pillar Blind Spot (from google image)

The pillar between the left and right front doors and the windscreen is called the A-pillar and is the main structure that holds the windshield in place and absorbs impact energy when the car is mounted. However, such a structure will inevitably block the driver's view. Under normal circumstances, the driver will have a blind spot of approximately 8 degrees, the size of which increases with distance. As the speed of the vehicle increases, the driver's view becomes smaller, and the A-pillar becomes more obscured, thus seriously compromising traffic safety (Sui, Chen, 2022).

Figure 2.8 explains the extent of the blind spot blocked by the A-pillar, which, as can be seen from the picture, increases the distance from the driver. The yellow-coloured area is the blind spot that the driver cannot see.

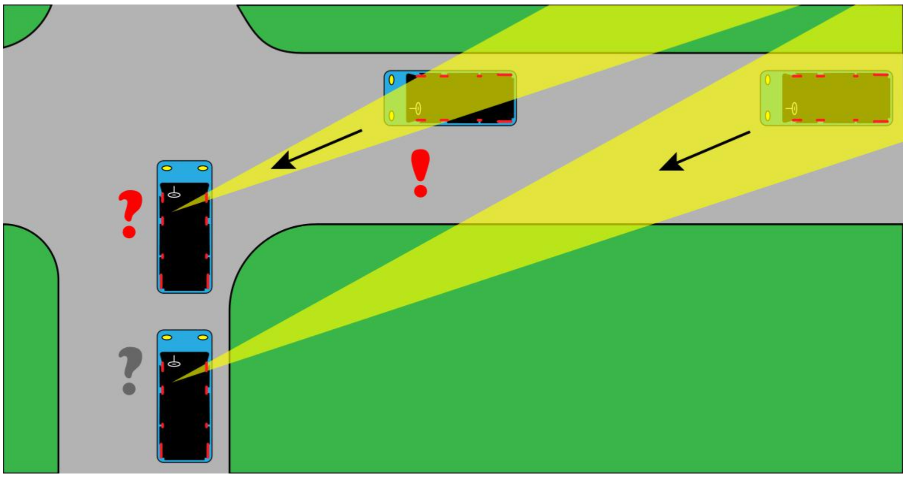
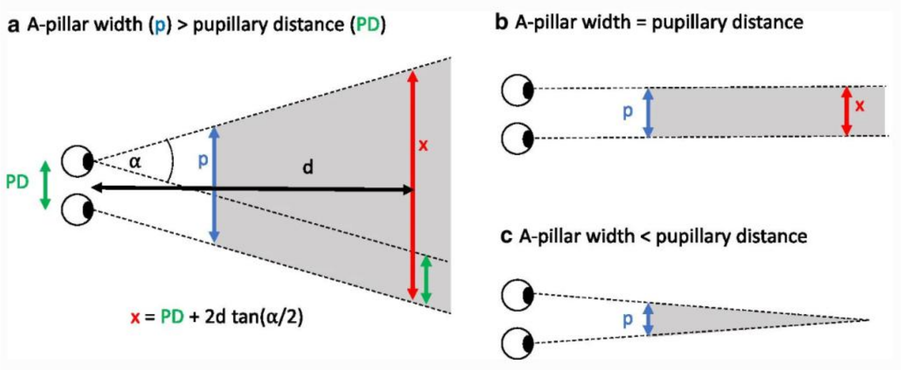


Figure 2.8 A-pillar blind spot area

The A-pillar is the same as the driver's pupil distance and remains constant when the driver's pupil distance is greater than the width of the A-pillar. Conversely, when the width of the A-pillar is less than the pupil distance, the driver's blind spot range decreases (Ekroll, Svalebjørg, Pirrone, Böhm, Jentschke, van Lier, Wagemans,2021). The width of the A-pillar determines the size range of the driver's blind spot. In Figure 2.9 gray area is blind spot, P is the A-pillar PD is driver's pupil distance, X represents the width of the blind spot area, which is a variable that changes with d (distance)


a A-pillar width (p) > pupillary distance (PD)

b A-pillar width = pupillary distance

c A-pillar width < pupillary distance

Figure 2.9 A-pillar blind spot explain (Ekroll, Svalebjørg, Pirrone, Böhm, Jentschke, van Lier, Wagemans, and Høye, 2021)

For large vehicles, the blind spot range is far greater than for ordinary cars. The driver's blind spot for large vehicles, especially trucks, is much larger than for regular use, and much of the road surface around the truck is unobservable to the driver (Harmon Parker 2020).

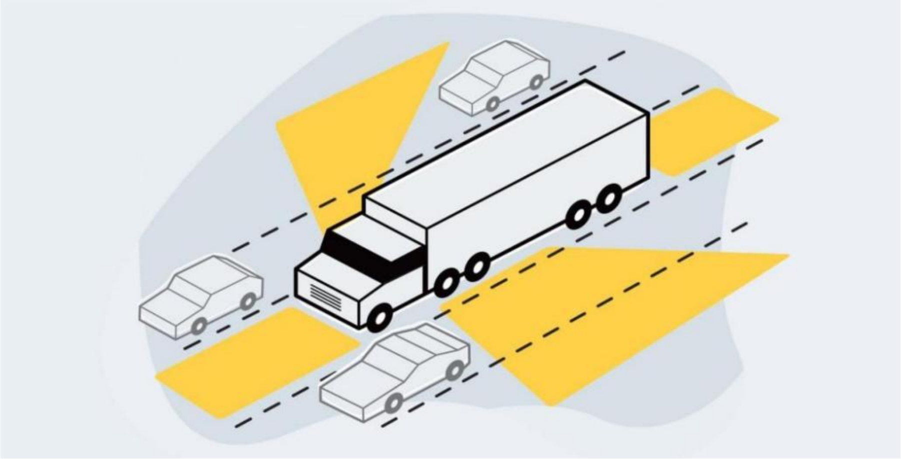


Figure 2.10 Truck blind spot

Figure 2.10 shows the extent of the truck's blind spot (in yellow), which includes the following points.

1. Directly in front of the truck’s cab for about 20 feet.
2. Directly behind the truck’s trailer for about 30 feet.
3. Along each side of the truck extending backward diagonally.
4. Immediately below and behind the driver’s window.

Semi-trucks have larger blind spots than other vehicles. Drivers of passenger vehicles or smaller vehicles can more easily turn their heads to see out their side windows, they have rear-view mirrors, and they are more likely to use their mirrors to check their blind spots (Bohn & Fletcher 2022). Compared with regular cars, a truck has a more prominent blind spot for the driver, which also increases the chances of an accident. The truck has a broader body knot and a longer body.

2.3.2 Blind Spots in Road Traffic

The blind spot are also on the left side of the car.

Motorbikes are in the driver's blind spot on the left side, so if the driver makes a direct left turn, the steering angle is too small, and the motorbike will be directly involved in an accident (Figure 2.11) Suppose A represents an oversized lorry. Motorbike B changed to a small vehicle, which is difficult for the driver to notice without the help of blind spot radar or additional blind spot reflectors.

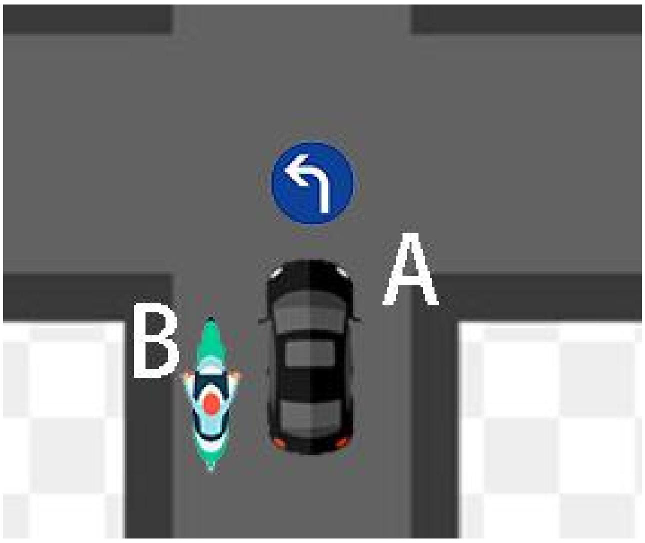


Figure 2.11 Blind Spot on Left Side

Overtaking in Singe Lanes

On a two-way road, if C wants to overtake, but B is in the way, the driver will not be able to see A. If the driver does not look ahead, he could quickly get into an accident with A when overtaking. This situation is very likely to occur in places with many bends. If B is a large lorry, C will not be able to see A (Figure 2.12).

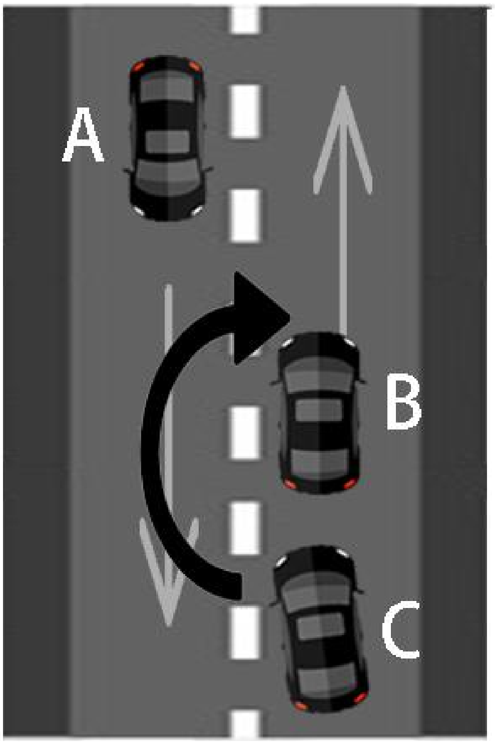


Figure 2.12 Overtaking in single lanes

2.4 Solutions for Blind Spots

The driver's blind spot causes a large proportion of traffic accidents, where two vehicles suddenly appear from each other's realization nearby and the driver is unable to stop the moving vehicle in a short time, or both parties do not have enough time and space to avoid it, thus causing a traffic accident (Wang, Jin, and Wu, 2022,).

There are many solutions to the blind spot problem from the government and car manufacturers, which we have reviewed a large amount of literature to summarise and analyse later. These include the simple and convenient use of physical lenses and the application of electronic radar.

2.4.1 Additional Rear View Mirror

Special reflectors with large angles are one of the most common and straightforward blind spot solutions in people's lives regarding traditional driver blind spot solutions. There are two types of reflectors, Figure 2.13, which has mounted on vehicles, and sizeable blind spot reflectors, Figure 2.14, which are retrofitted on sharp turning roads with blind spots or in indoor car parks

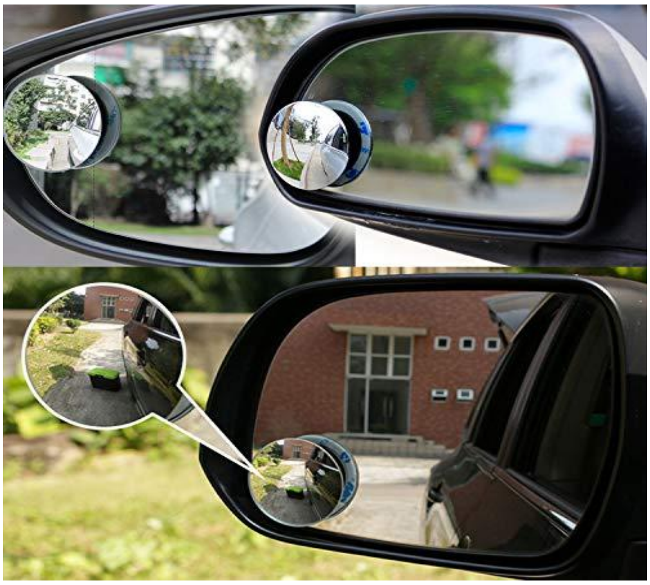


Figure 2.13 Vehicles Blind Spot Mirror

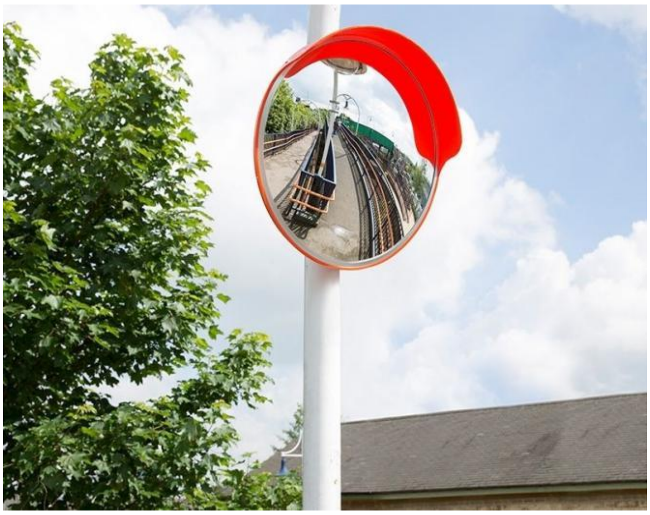


Figure 2.14 Streets Blind Spot Mirror

The Blind spot Assist Mirror is a simple wide-angle mirror used to widen the rear-view mirror's field of vision, mainly for the driver's blind spot in the road and the vehicle behind.

Mirrors are quite useful while driving a car. A driver cannot see the blind spot. If a vehicle or pedestrian comes into this blind spot, the chances of a dangerous accident increase. With these special mirrors attached to your side-view mirrors, you can see the blind spot clearly to avoid any mishap (Ghosh, 2022). Blind spot mirrors aim to cover the parts your standard mirrors miss, giving you a far greater view of what’s going on behind your car. If a car does not come with radar or driver blind spot assist, adding a blind spot assist mirror is an excellent option to reduce the risk of dangerous traffic situations due to driver blind spots (Hester, 2021). Although additional wide-angle mirrors can give drivers a more excellent range of vision in their blind spots, small objects are difficult to spot due to the small size of the mirrors. Too many mirrors can also distract the driver's attention.

2.4.2 Automotive RADAR and LIDAR Applications

Automotive radar systems are one of the essential measures to ensure safety on the road. The millimetre wave radar detects obstacles around the vehicle, and LiDAR works to detect the distance and relative speed of obstacles. In addition, radar systems are an essential component of Advanced Driver Assistance Systems (ADAS) (Patole, Torlak, Wang, and Ali, 2017).

RADAR

Improving the driver's awareness of the road environment and the vehicles in front and behind him is one of the principles of traffic accident reduction. Radar is a safety technology system currently implemented and used in their cars (Grimes, and Jones,1974). Radar technology is the application of radio waves. The presence is determined by the emission and reception of radio waves, and advanced radars can detect the size and speed of objects. Most automotive radars use millimetre wave frequencies to detect objects at a distance or in the road environment and obstacles around the vehicle. Other vehicles, pedestrians, roadblocks etc., are the primary targets for radar detection (Shawn , 2019). Advances in sensing technology and communication technology have transformed the classic car. Functions and safety systems for interacting with the environment have become essential features of most cars. For initial awareness of the environment, airborne radar provides a quick and decisive solution (Vazquez 2022). Automotive radar systems are an essential technology for vehicles in traffic safety. It is also the primary sensor used for automatic cruise control. With the development of technology, radar is also an essential and critical component of autonomous driving assistance systems (ADAS). It is mainly used to avoid collisions, detect the presence of objects around the car and ensure safe driving (Parker, 2017).

Early developments in the use of radar for automotive safety driving focused on frequencies in hertz: 17 GHz, 24 GHz, 35 GHz, 49 GHz, 60 GHz and 77 GHz. The aim was mainly to use radar to alert the driver and thus reduce the probability of a car collision (Schneider. 2005) Figure 2.15 Shows the operation of the radar on display.

Figure 2.15 The operation of the radar on display

Radar is the most mature and primary sensing system in the field of automotive safety technology, which is for detecting objects around cars widely; working well in most environments, and its reliability is unquestionable (Roriz, Cabral, and Gomes,2021).

In current technological developments, three main types of radar are used in automotive safety applications. They are as follows (Shawn 2018).

1. Short-range radar (SSR) The main applications are parking assistance functions and collision warnings for objects at close range. It is from this that we hear alarms when reversing or when parking.

2. Medium Range Radar (MRR) Mainly used for driver blind spot monitoring, safe lane changes or corner collisions on turns etc.

3. Long Range Radar (LRR) This radar sensor is generally installed only at the front of the car and is used for the automatic car cruise function, detecting objects in front of the car at a distance.

Figure 2.16 shows the range and position of the different waveform radars in the vehicle.

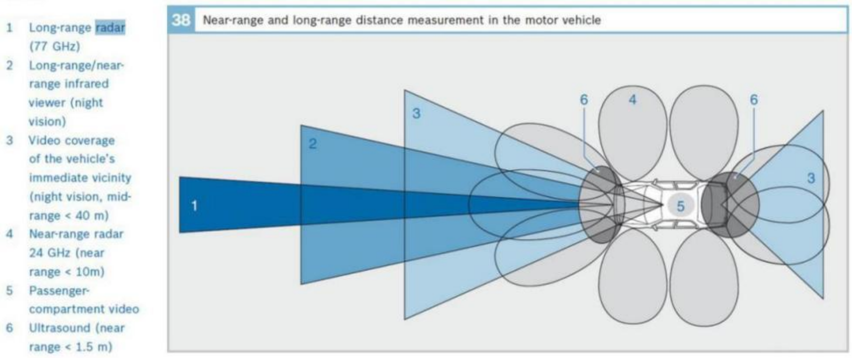

1 Long-range radar (77 GHz)
2 Long-range/near-range infrared viewer (night vision)
3 Video coverage of the vehicle's immediate vicinity (night vision, mid-range < 40 m)
4 Near-range radar 24 GHz (near range < 10m)
5 Passenger-compartment video
6 Ultrasound (near range < 1.5 m)
38
Near-range and long-range distance measurement in the motor vehicle
1
2
3
6
4
6
3
5

Figure 2.16The range and position of the different waveform radars (From google image)

However, RaDAR technology had to develop for a long time to become a more mature security system. However, the drawbacks of RaDAR are still evident. Although radar technology is constantly evolving and improving, the environmental information provided by radar still needs to be clarified and it provides insufficient sensing resolution to give the driver accurate information (Steinbaeck, Steger,Holweg, and Druml, 2017).

With the development of automotive technology, the disadvantages of using Radar systems in cars have become more apparent, the main ones being the following:

The atmospheric medium determines the response time of the radar, which takes more time to detect sandstorms or foggy weather.

If the detected object accelerates or decelerates by more than 1mph/s, the radar cannot track it. It is also difficult for radar to detect targets that change speed.

If the target object is too close to the radar, or if the target object is too large, it can saturate the radar receiver resulting in inaccurate information.

Radar cannot distinguish the type of target object and can only give feedback on the presence or absence of the target object within range.

Radar is inaccurate and ambiguous information can cause false alarms in the system

6. Over-sensitivity is also a disadvantage of radar (John. C. I 2019).

LIDAR

LiDAR systems work similarly to radar systems in that Radar transmits and receives radio waves, and LiDAR uses a laser. However, the feedback from the laser is more accurate and reliable because radio waves are absorbed or consumed in large part when they come into contact with an object, but the laser is not (Neal, A. 2018). This is essentially an advanced sensing method that detects objects and measures their distance by shining a laser pulse, invisible to the naked eye, in front of them and measuring the reflected signal. The width of the emitted laser pulses can be as small as nanometres. (Khader, and Cherian, 2020).

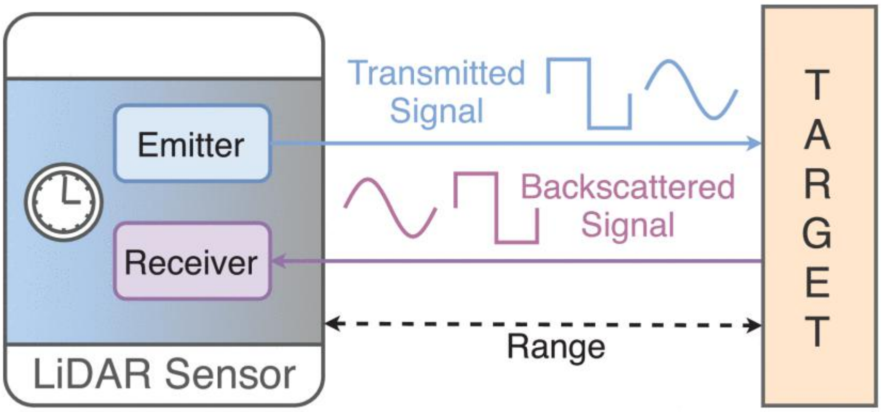
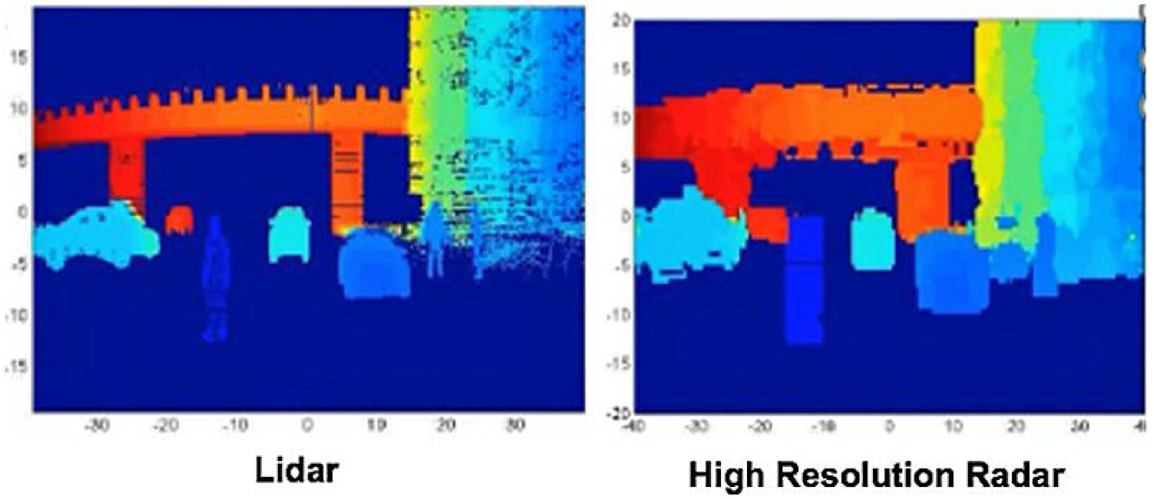


Figure 2.17 Operation principle of LiDAR technology

The LiDAR system can be divided into two parts, the laser ranging system scanning system. Laser ranging consists of a laser transmitter, which illuminates the target by modulating the laser. A photodetector produces an electronic signal by receiving photon reflections for photoelectric conversion. The optical element accurately focuses the received reflections onto the photodetector, and the signal processor calculates the predicted distance to the reflected target (Li, and Ibanez-Guzman, 2020).

LiDAR is an alternative development to laser detection radar, which is primarily used for distance detection. LIDAR works by firing a laser at surrounding objects that are obscured and measuring the time it takes for the laser to reach the object to calculate the corresponding distance. However, LiDAR is not widely available and only a few expensive cars are equipped with LiDAR for forward distance safety detection. (Kevin Lim, Paul Treitz, Michael Wulder, Benoˆıt St-Onge, and Martin Flood). LiDAR technology is evolving rapidly as automotive technology continues to advance, and new LiDAR technologies are constantly being updated. Along with the development of autonomous driving technology, LiDAR is one of the critical solutions to enhance the driver's perception of the vehicle's surroundings (Khader, and Cherian, 2020). The essential task of LiDAR is to measure distance. The LiDAR technologies are being improved, so that LiDARs gain camera-like vision in addition to the ability to measure distance. However, fundamentally LiDAR was developed for distance measurement based on Radar. LiDAR struggles to work appropriately in bad weather, and even cars with LiDAR still need Radar's aid. However, LiDAR produces data at a much higher level of detail and in a clear field of view than Radar, Figure 2.18 compares Radar's imaging with that of Lidar. At the same time, the clearer picture and more accurate data come with expensive LIDAR laser equipment and high maintenance costs. (Karpathy, 2021)



Lidar

High Resolution Radar

Figure 2.18 compares Radar's imaging with that of Lidar.(Fierce Electronics)

However, LiDAR systems are like an upgrade to radar systems, being more accurate and including distance detection capabilities that Radar does not have, but LiDAR still has many disadvantages. Dampness in the atmosphere can directly affect the feedback results and detection range of LiDAR, the severe rain, snow or fog can directly cause LiDAR to fail to detect objects at long distances, and the range of LiDAR is not even as wide as that of the human eye in some extreme conditions (Rasshofer, R.H. and Gresser, K., 2005). Although LIDAR is far more accurate and faster than ordinary radar, there are still difficulties in recognition due to atmospheric visibility or bad weather. The lasers can cause irreversible damage to the human eye, safeguarding the human eye is a requirement for LIDAR, which is more resistant to radiation at wavelengths greater than 1400 nm and is limited to 1550 nm(Warren,2019,). The high maintenance cost is an issue that must be considered in the car's development. Now, the LIDAR technology is challenging as vehicles on a large scale due to cost and the mentioned reasons. The need to reserve ample enough space in the vehicle for LIDAR devices is a significant challenge for vehicle design (Ibáñez,, Zeadally, and Contreras-Castillo, 2018.) With the introduction of new energy and environmental policies, battery energy is the way to go for the automotive industry. Nevertheless, unfortunately, LiDAR systems consume a lot of electricity and have a direct impact on the range of trams, which is one of the reasons why LiDAR is not popular (Karpathy, 2021).

2.4.3 Summary and Evaluation

In this section, through an extensive literature review, we analyse solutions for the driver's blind spot when sensing its environment. The application of additional mirrors and Radar is described in detail, and the working principles of Radar and LiDAR are analysed, as are the advantages and disadvantages of the current mainstream solutions. As society and the government take traffic safety seriously, car manufacturers are constantly developing and updating vehicle safety systems. From simple physical wide-angle mirrors to today's sophisticated millimetre wave radar and LIDAR, the level of safety in cars is constantly increasing. However, advanced LIDAR technology has been equipped in some cars, and the disadvantages of radar are hard to change. At the same time, additional wide-angle mirrors can give drivers more visibility in their blind spots, but it is difficult for the driver to notice the image in the blind wide-angle mirror while the car is in motion and while paying more attention to the front of the car.

2.5 Related Inspiration from Science Fiction Films

This section presents depictions of future cars from science fiction films that have already been released, focusing on autonomous driving functions and flat-screen display (HUD) functions

2.5.1 Autonomous Taxi

Many scenes depicting future technology that appear in science fiction films are primarily for cool effects, thus helping to boost the box office or as a plot necessity to help the protagonist explore. However, many of these novel technologies could well change the shape of the world we currently live in. As early as 1990, in the film Total Recall, self-driving taxis called "Johnny's cabs" (Figure 2.19) were introduced. They could be driven autonomously on the road without human intervention with a robot charged accordingly. In the film iRobot starring Will Smith, autopilot is standard in cars and can be switched between autopilot and manual control at any time by the driver (Craig Sheldon 2018).

Figure 2.19 Total Recall Johnny's cabs

All of the future technologies mentioned above have already been realised. In China, self-driving rental cars are already present in many cities and are generally accepted by society. The switch between autonomous and active driving has been implemented and is widely used in today's car brands, such as Tesla. The technology in science fiction films is not just a figment of the imagination some of them might well become a reality in part in the future.

2.5.2 Combine Object Detection with Driver Blind Spot

As time progresses, the car has become an essential part of everyday life as a means of transport for people travelling in the middle of their every day activities in their lives. However, traffic accidents break up people's ordinary lives with property damage, personal injuries and even life-threatening situations. The number of cars worldwide has reached one billion, and road safety has become a fundamental societal challenge. The rapid development of technology and the continuous innovation of vehicle safety systems have ensured traffic safety. New traffic intelligence systems that enhance the driver's awareness of the road and spatial environment around the car and allow the driver to interact with the car are one of the most critical directions in developing automotive safety (Procedia 2015).

As a safety measure for blind spots on the left and right sides of the car, some cars are now equipped with Blind Spot Monitors, which alert the driver to vehicles in the blind spot on both sides of the car by means of an icon in the rear view mirror, which is detected by radar. Figure 2.20.

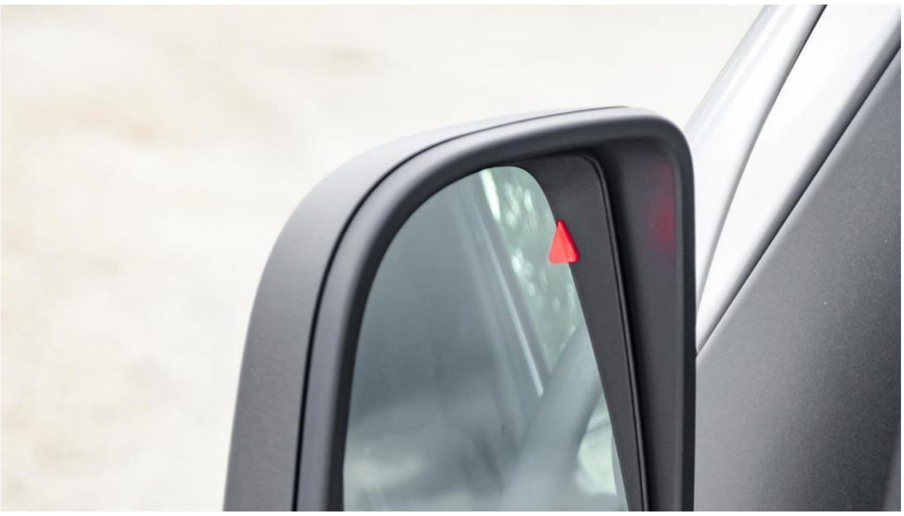


Figure 2.20 Blind spot monitors

While drawbacks may vary depending on the automaker, the biggest issue across the board is the ability of the technology to detect fast-moving vehicles efficiently. As mentioned previously, studies found that blind-spot monitoring systems would often alert drivers too late of an approaching vehicle in an adjacent lane. Since most blind-spot monitoring systems work on visual cues on the driver’s side mirror, it may be difficult for the driver to pick up in time. An outside circumstance such as bright lights can affect the driver ability to notice the signal (Ouyang 2022). This reminder is a vague concept for drivers and although it serves as a reminder, it is often difficult to give a precise and clear answer. Developing a blind zone alert safety system that can give precise answers is one of the critical challenges facing motor traffic today.

Here we find inspired by the technology shown in the film and try to apply AR and VR technology to the traffic safety system to subjectively remind drivers of the surrounding traffic environment and passing vehicles. Augmented reality (AR) and virtual reality (VR) technologies have gradually matured and entered ordinary people's lives. If this technology can work into the security system, subjective reminders will let the driver notice it. As a result, the dangerous situation gradually dramatically reduces the occurrence of traffic accidents. We are inspired by technology films depicting the future of automotive technology, to envisage cameras in the blind spots of cars. The screens are displayed near the driver's instrumentation or projected onto the car's front windscreen in combination with HUD technology so that the driver can see the road environment in a real blind spot. At the same time, we intend to use AI algorithms for blind spot detection so that when the camera captures objects in blind spot areas, it threatens driving safety. They will be detected and displayed to the driver with a coloured frame when the driver sees a special coloured box in the display to be alerted to the blind spot and can look ahead for the rest of the time.

2.6 Objects Detection

In this section, we will review and introduce the literature on Object Detection, analyze and discuss the development directions and applications of Object Detection, and with finalise with conclusions.

2.6.1 Introduction to Object Detection

Object detection (recognition) and AI deep learning techniques have attracted more attention as technology (Ashish Patel 2020). Object detection is also the basis for the development of most computer vision techniques and is one of the critical fundamental problems of artificial intelligence (Zou, Shi, Guo, and Ye, 2019). The development from the early days of radar detection to the current object detection in computer vision has been revolutionary. Object detection is the process of classifying and localising an object and ultimately determining the type of object and its current location (Choudhury, 2022). Candidate objects are extracted from the target task, and physical information about the object is predicted, i.e. the location, shape and size of the object. Object detection methods include comprising ground filtering and clustering in most traffic scenarios where the target object is kept perpendicular to the ground. The ground is first determined, non-ground objects are marked out, and then non-ground objects are grouped into different items using clustering methods (Li, and Ibanez-Guzman, 2020).

Figure 2.21 shows the result of the object recognition algorithm. We can see the vehicles framed and labelled with the type of vehicle.

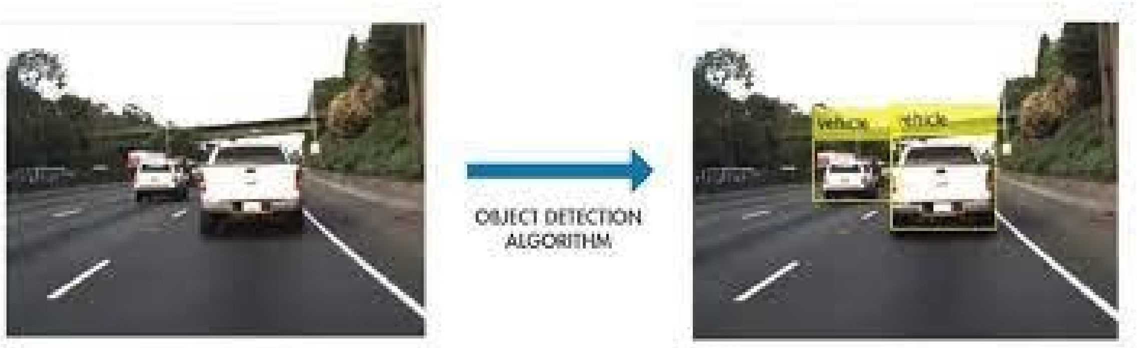


Figure 2.21 Object detection result

Object detection is an esoteric computer vision technique focused on identifying and labelling objects in images, video and even live footage. Object detection models are trained using a surplus of annotated visual objects to perform this process using new data. It becomes as simple as providing input visuals and receiving fully labelled output visuals. We will discuss the object detection model in more depth later. A key component is the object detection bounding box, which identifies the edges of objects labelled with clear quadrilaterals - usually squares or rectangles. They are accompanied by the object's label, whether a person, a car or a dog, to describe the target object. Bounding boxes can be overlapped to show multiple objects in a given shot as long as the model has a priori knowledge of the items it labels.

2.6.2 Deep Learning Review

Artificial neural networks are the foundation of deep learning, which has been developing rapidly as the age of intelligence has opened up (Du, Cai, Wang,and Zhang,2016).Deep learning is a branch of machine learning that learns abstract concepts from data through different layers of constructs and has a wide range of applications in artificial intelligence migration learning, computer vision (Ciresan, Meier, Schmidhuber, 2012), natrual language processing (Mkolov, Sutskever, Chen,2013), and semantic parsing (Bordes, X. Glorot,Weston2012).

Deep learning has grown rapidly and successfully in recent years (LeCun, Bengio, Hinton, 2015)

It has been used in various aspects of everyday life, As of now, we have applied it in our lives in the categories such as autonomous driving, object recognition, medicine, biology and statistics. Research in artificial neural networks are the primary theoretical and conceptual source of deep learning (Hong, 2011). The principle of deep learning is learning representations from a large, multi-level data abstraction. At the same time, Deep learning attempts to model levels beyond the target hierarchy of data, classifying in multiple layers of stacked information modules. This is a very stable type of hierarchical learning, where the system learns complex representations directly from all the data on the input (Dhillon, and Verma, 2020). The conditions that facilitate the development of deep learning are: 1. The increase in the speed and performance of computer chips, such as the popularity of GPU units. 2. The reduction in the cost of hardware, such as low-cost, high-performance graphics cards. 3. The development and updating of machine learning algorithms, such as the Yolo family of detection algorithms (Deng, 2014)

Figure 2.22 (Guo, Liu, Oerlemans, and Lew, 2016) which shows the branches of development of deep learning, which are discussed and summarized in this thesis, mainly in the direction of convolutional neural networks (CNN).

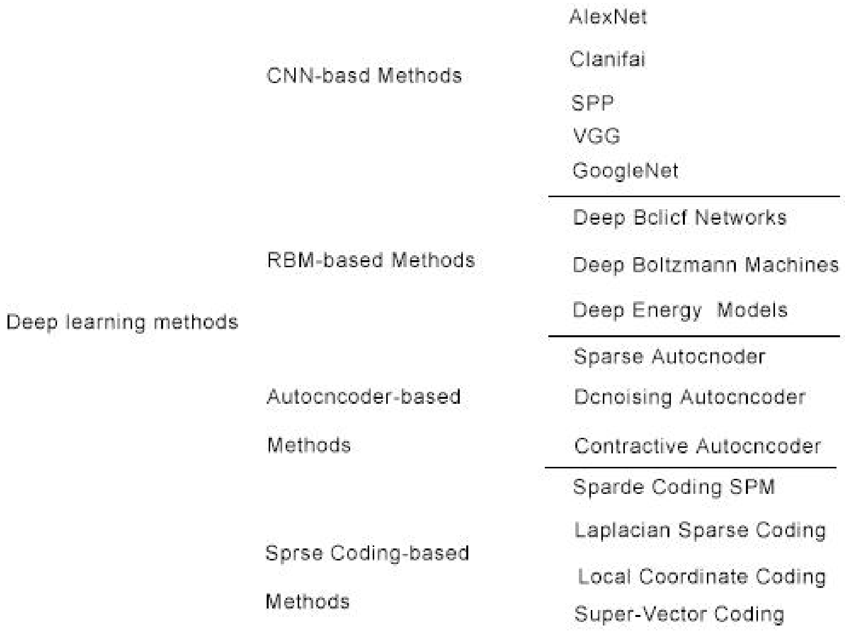


Figure 2.22 The developing progress for deep learning (Guo, Liu, Oerlemans, and Lew, 2016)

2.6.3 CNN Overview

Back in 1998 Convolutional Neural Networks (CNN) were first proposed by Fukushima (Fukushima, 1998), Convolutional neural networks have a wide range of applications, including activity detection (Papakostas, Giannakopoulos, Makedon, Karkaletsis, 2016), text detection(Kim,Y.2011), paragraph detection (hou, Gong, Fu, Du2017), face detection (Ranjan, Bansal, Bodla, Chen, Patel, Castillo, Chellappa, 2018), image characterization (Druzhkov,,Kustikova.2016), object detection (Milyaev, Laptev 2017), etc. Figure 2.23 (Zhao, Peng Zheng, 2018) Illustrates the development and stages of convolutional neural networks (CNN)

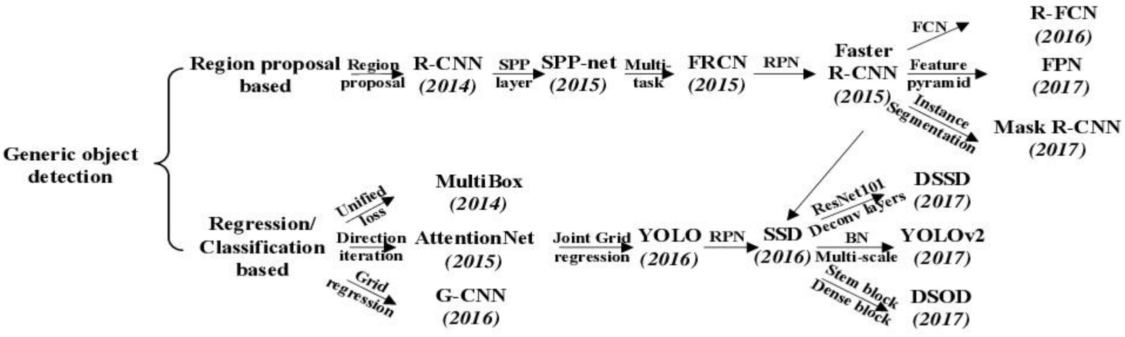


Figure 2.23 Object detection With deep learning(Zhao, Peng Zheng, 2018)

Deep learning with Convolutional Neural Networks (CNNs) is a mainstream AI learning system which uses multi-level distributed training (LeCun, Bottou, Y. Bengio, 1998). CNN is a deep learning model for the analytical processing or recognition of data with grid pattern features, such as images. The organisation of the animal's visual cortex inspires (Hubel, Wies, el ,1968) it. CNN which can learn spatially hierarchical features on their own and in order of rank from low to high. The CNN is also a mathematical structure typically consisting of three modules, namely the convolutional layer, The ensemble layer, The connection layer.

In Figure 2.24 the convolutional and ensemble layers are responsible for extracting features from the target. The final layer, the fully connected layer, is responsible for the final output of all the extracted features, such as item classification (Yamashita, Nishio, and Togashi, 2018)

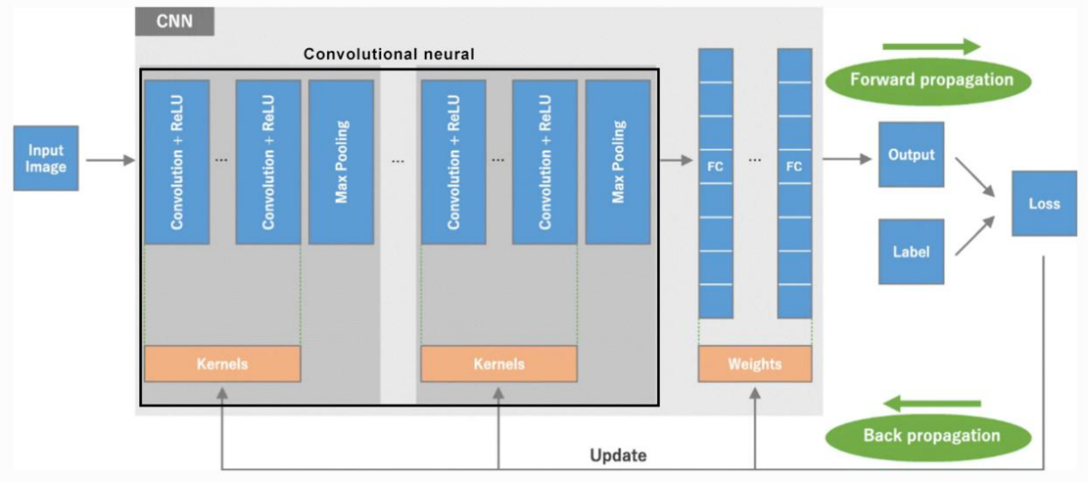


Figure 2.24 Convolutional neural network

The processing of the solid image case is in Figure 2.25 (Krizhevsky, Sutskever, Hinton. 2012). A flowchart for the recognition of the goldfish in the image.

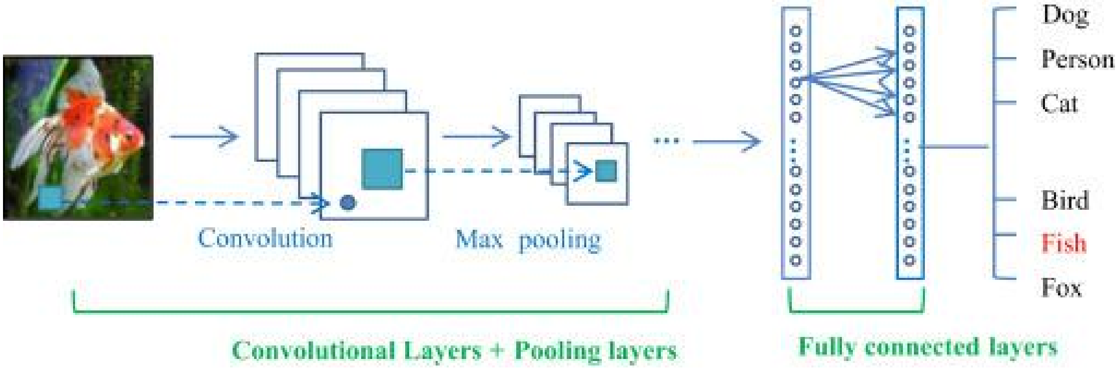


Figure 2.25 CNN image detection Case

2.6.3.1 Convolutional Layer

In the first convolutional layer, the CNN uses a kernel algorithm to convolve the prominent features of the entire image to generate the feature image Figure2.26 (Krizhevsky, Sutskever, Hinton. 2012)

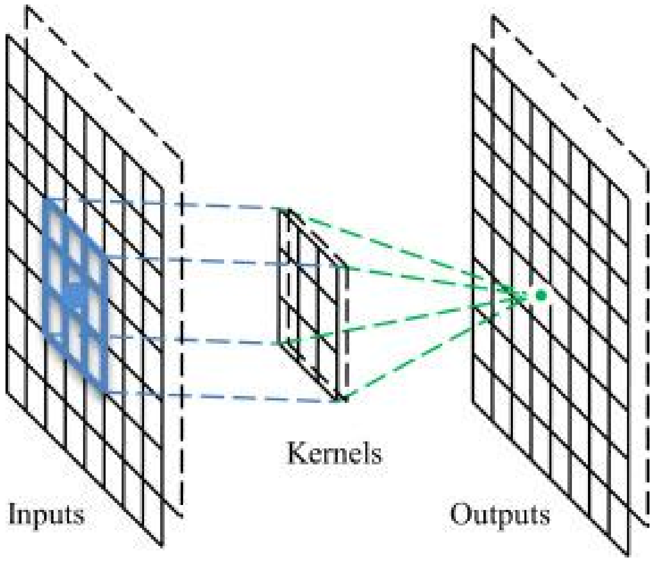


Figure 2.26 Convolutional layer

Convolutional operations (Zeiler Hierarchical 2014) are an important part of the convolutional layer. The main advantages of its operation are: 1. the sharing mechanism reduces the number of parameters (in the same feature map) (Ñanculef,Radeva,Balocco2020), 2. local connectivity sharing learns correlations between neighbouring pixels (Jeong, Pfister, Fatica, 2011) 3. the uniqueness of the position of the target object features that do not change due to the convolution operation (Szegedy, Liu, Jia 2015).

2.6.3.2 Pooling Layer

The pooling layer Figure2.27 of the convolutional neural algorithm is relatively widely developed and researched. There are three ways to obtain results for the pooling layer, each with a different research purpose.

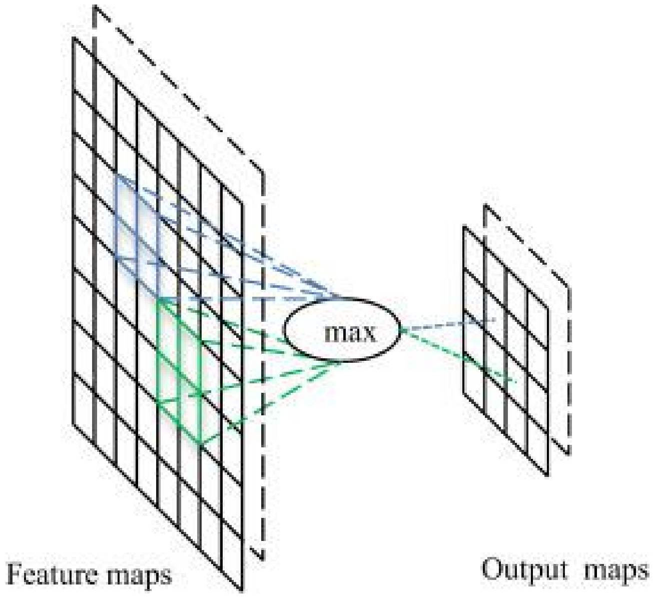


Figure 2.27 Pooling layer

Stochastic polling

At the pooling layer level, the stochastic pooling method addresses the disadvantage of maximum pooling leading to overfitting (Zeiler, Fergus, 2013). Instead of passing all pooling, a multinomial distribution with random selection is used to randomly select the activation samples for each pooled region, which reduces the workload and increases the speed while effectively solving the overfitting problem (Zeiler Hierarchical Convolutional).

Spatial Pyramid Pooling (SPP)

The input target in a convolutional neural network usually requires a fixed image size, and varying target images can affect the accuracy of the detection results. This problem can be solved by replacing the last layer of the pooling with a spatial pyramid pooling layer in the CNN base architecture. This is because spatial pyramid pooling can be used to represent a fixed length from any region, thus allowing images of different sizes and scales to be generated and processed. Pyramidal pooling can be adapted to any convolutional neural network and improves the structural performance (He, Zhang, Ren, 2014)

Def -Pooling

In the field of vision computers, object recognition is a challenge for the processing of target deformations.

Max pooling and average pooling both are effective in dealing with target deformations. However, they cannot learn with partial deformation and geometric models of objects (Friedenthal, Moore, Steiner, 2015).

To change the traditional pooling constraint on deformation, a new pooling layer, def-pooling, has been developed to handle deformation priming more efficiently, enriching the depth model by learning deformations of visual patterns. Def-pooling can also replace the traditional maximum pooling layer at any level of information abstraction. Improved CNN framework performance (Ouyang, Luo, Zeng, 2015).

2.6.3.3 Fully-connected Layers

Figure 2.30 CNN Image Detection Case, we can know the last layers named Fully-connected layers. The prominent role of this layer is to connect all the 2D features in the Pooling layer and finally transform them into a 1D feature vector, Figure 2.28.

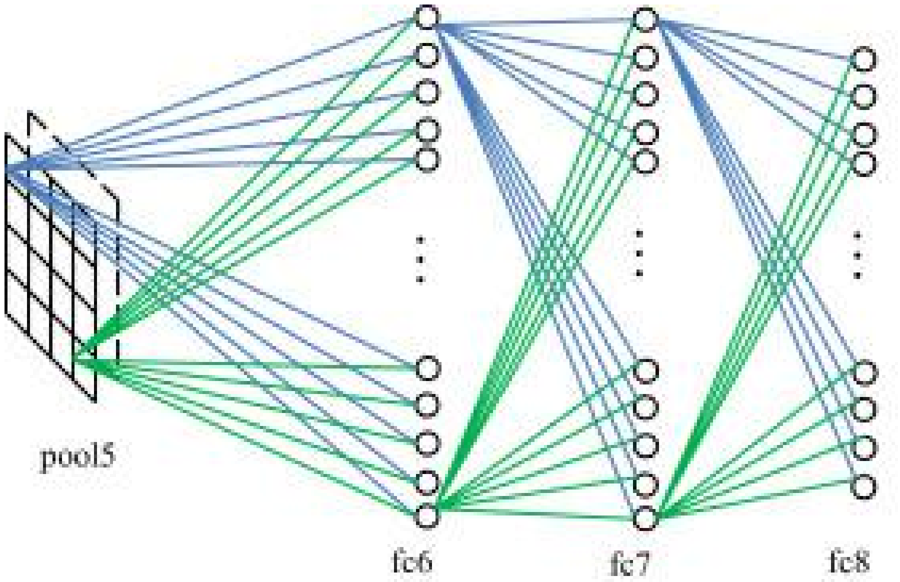


Figure 2.28 Fully-connected layer

The final fully connected layer contains a large amount of recognition information in the CNN. The results of the pooling layer are fed back into a pre-defined length vector. This length vector can be converted into numerical categories for image classification (Krizhevsky, Sutskever, Hinton2012). Or as a feature vector, which is the basis for the next operation (Girshick, Donahue, Darrell, 2014). The structure of the performance of the fully connected layers is rarely changed, and in some special visual recognition situations, the transfer learning approach is applied to the connections (Oquab, Bottou, Laptev, 2014).

The learning speed and results of convolutional neural networks and deep learning now far exceed those of traditional Machine Learning (Wang, X.J., Zhao, Wang, 2012), even Naive Bayes (which based on Bayesian principles and use knowledge of probability statistics to classify sample data sets) (Rish, 2021). With the development of computer vision and the widespread use of object detection, there will be newer structural developments in convolutional neural networks in the future.

2.6.4 Objects Detection Algorithms Review

Object detection algorithms are an important development in the field of computer vision. Different object detection algorithms have emerged with the widespread use of object detection algorithms and the expansion of the database of detection target models (Hu, Tan, Wang, 2004). The Viola-Jones framework which is working for face recognition detection (Viola, Jones 2001), this algorithm has made human detection universal. Examples include mobile phone face recognition or bank security systems (Padilla, Costa Filho 2012). Several branches have been developed with the update of the version, such as pedestrian detection (Trivedi, 2016), car detection (Sun, Bebis, Miller2006), etc.

With the rapid development of computer vision and the popularity of convolutional neural networks (CNN) and computer AI claim learning algorithms, visual object detection is entering a new era (He, Zhang, Ren. Sun 2016). With the continuous development of convolutional neural networks (CNN), faster and more accurate algorithms have emerged. R-CNN (R. Girshick, Donahue, Darrell .Malik 2014); Fast R-CNN (R. Girshick 2.15); Faster R-CNN (Ren, He, Girshick and Sun 2015); R-FCN (Dai, Li,. Sun) ; SSD (Liu, Anguelov, Erhan, Szegedy,,2015); YOLO (Redmon and Farhadi, 2017). The main difference between the CNN-based neural network framework detection algorithm with the Viola-Jones algorithm is that the CNN detection algorithm is flexible enough to train with several categories (tens or even hundreds) at the same time. (Padilla, Nettoda Silva, 2020) which allows object detection algorithms to become more accurate and faster. We present a review of several mainstream detection algorithms, including, R-CNN, Fast R-CNN and YOLO.

2.6.4.1 R-CNN

In 2014 Ross Girshick proposed the R-CNN algorithm, which achieved a much higher average precision of 54% than other algorithms at the time, compared to the best results of PASCAL VOC 2012 of less than 25% two years earlier. The R-CNN algorithm improves the candidate bounding boxes and extracts detailed features by depth framing (He, Gkioxari, Dollár, and Girshick, 2017)

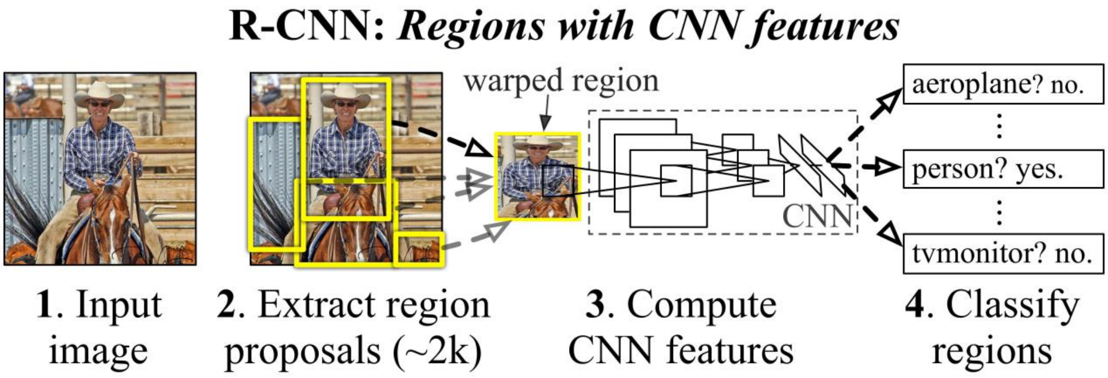


Figure 2.29 Object detection algorithm R-CNN

Figure 2.29 illustrates the three constructs of the R-CNN.

After the target region is generated, R-CNN takes a selective search, averaging 2000 initial target proposals per image. Next comes feature extraction, where each region is cropped to a fixed resolution, and the CNN standard model has been used to extract 4096-dimensional features as the final result. The features have a specific high-level semantic representation for each region of the area classification and localisation; each extracted feature is transferred to an SVM that simultaneously classifies the items in the candidate zone days.

To improve the accuracy of the boundedness, the R-CNN algorithm also predicts four additional values, which are not in the proposal for predicting objects (Bharati, and Pramanik, 2020).

Limitations:

1. The R-CNN works in a sequence where the corresponding features are extracted from the suggestions of different target regions and then temporarily stored on a disk. During this time, the deep internet, which used, e.g. VGG16 (Theckedath, and Sedamkar, 2020). It takes much time to process the small training set and a lot of disk space.
2. R-CNN uses selective search to generate region proposals in the target model. This approach leads to the algorithm generating redundant proposals (2k region proposals) and requires high time complexity. The whole process takes about 2 seconds.
3. The R-CNN tends to spend much time performing repetitive operations at detection time because the FC layer (Garipov, Podoprikhin, Novikov, Vetrov, 2016) requires a fixed size input of the target image.

2.6.4.2 Fast R-CNN

To address the limitations of R-CNN, a novel structural framework based on R-CNN Fast R-CNN was proposed by Grishick. This framework adds a multi-task loss function to the target bounding box regression. Firstly, same as R-CNN, a convolutional structure is introduced for image collation while generating a feature map. At last, vectors of equal magnitude are extracted from each target region in the region of interest (ROI, Chityala, Hoffmann, Bednarek, Rudin,,2002) pooling layer.

The main difference between Fast R-CNN and R-CNN is that the target features from the upper levels are combined into a whole larger region by using a RoI pooling layer. Figure 2.30. Then, the final softmax layer (Hu, Tian, Yin,and Wei,2018) predicts and speculates the region's class.(Rajeshwari, Abhishek, Srikanth, and Vinod, 2019). But here, the RoI return layer is a particular case of the SPP layer, which exists at only one pyramid level.

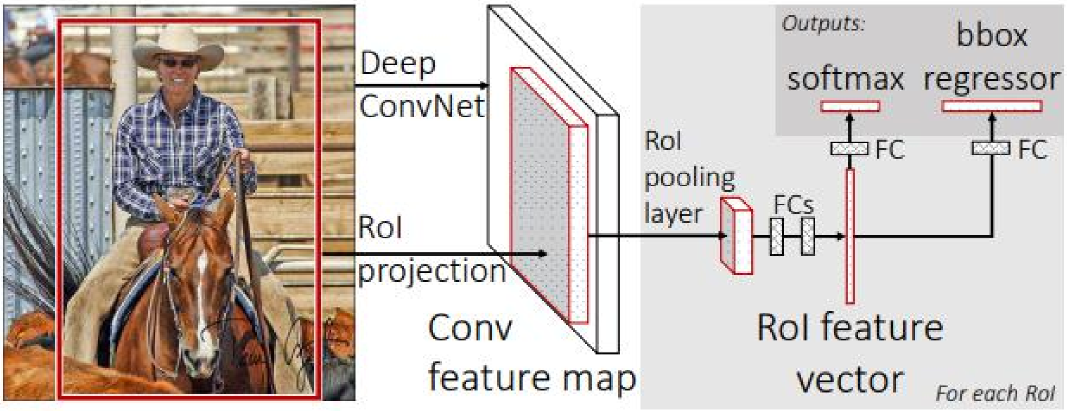


Figure 2.30 Fast R-CNN (Rajeshwari, 2019)

In the R-CNN described above, about 2k regions are processed by the convolutional neural network each time. However, the Fast R-CNN requires one convolutional operation per image and generates the target feature map. In fast R-CNN, single-stage processing training applied to the grid layer saves a lot of disk space and speeds up target detection accuracy. However, there is also a loss of task in generating region proposals.

Similarly, Fast R-CNN still uses the search for region suggestions, which is a slow and time-consuming process. It also relies on network performance (Ren, He,Girshick and Sun, 2015).

2.6.4.3 YOLO (You Only Look Once)

You Only Look Once (YOLO) is one of the most advanced object recognition algorithms available and is primarily used to process real-time object detection. The difference between YOLO algorithm and other object algorithms is detection area. Traditional object recognition uses regions to locate objects in an image, requiring the complete range of the target image. However, YOLO detects a range of regions in the target image where objects occur with high probability (Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi,2016). YOLO applies a single neural network to the complete image, divides the target image into regions, and predicts the bounding box probabilities for each region. Furthermore, all the bounding boxes had weighted from the predicted probabilities (Jiang, Ergu, Liu, Cai, 2022).

Figure 2.31 shows the basic framework architecture of YOLO, which is the latest recognition algorithm by combining the GoogleLeNet (Singla, Yuan, Ebrahimi 2016) image classification algorithm with CNN.

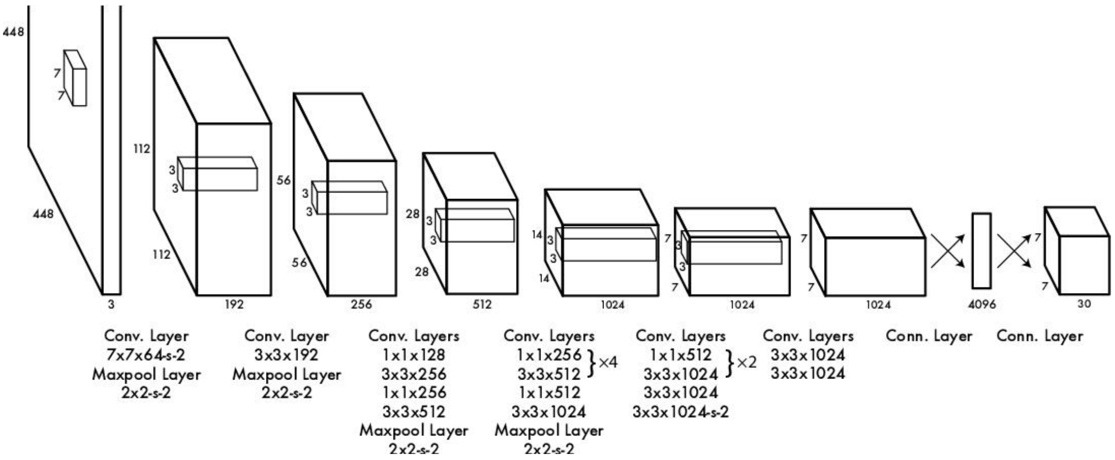


Figure 2.31 Framework rrchitecture of YOLO (Redmon)

YOLO divides the target image into an S X S grid, with each grid cell responsible for predicting only the objects centred on that grid. Each cell predicts the bounding box of the target and the corresponding confidence score Figure 2.32.

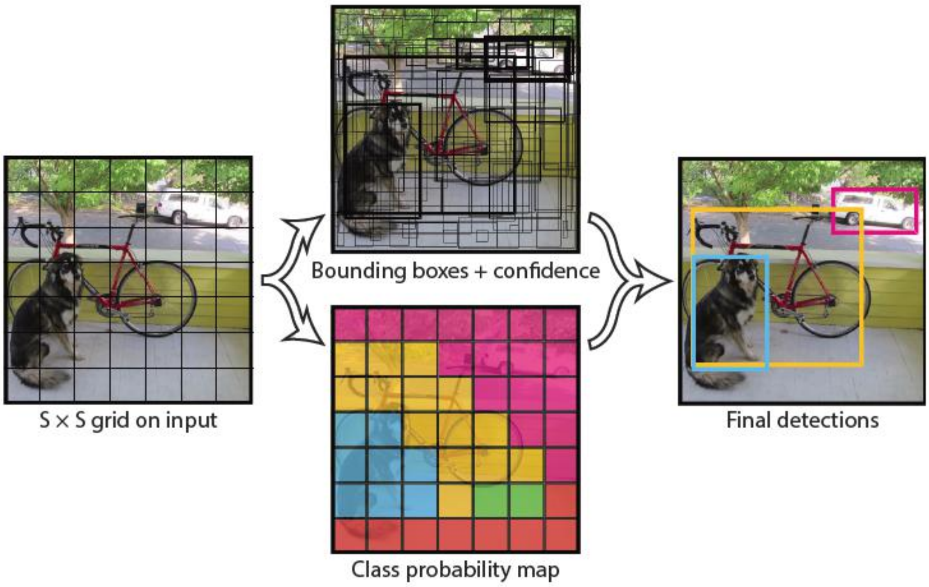


Figure 2.32 YOLO(*Mohan-2021)*)

YOLO is much faster than R-CNN or even Fast R-CNN. However, in the initial version of YOLO, the recognition of small objects is low due to the spatial constraint imposed by YOLO on the prediction of bounding boxes. Also, the initial version of YOLO had difficulty analysing objects with ill-defined boundaries. In later versions of YOLO, the above problems had been improved (Arya, M.C.Rawat, A., 2020).

This thesis focuses on the application of the YOLO system of algorithms to automotive blind spot detection. In chapter 4, we summarise and analyse YOLOv3 and YOLOv5 and discuss the details of the algorithm.

2.7 Summary

In this section, first we summarise the causes of traffic accidents through a large amount of traffic accident data and literature and identify the main reasons that influence traffic accidents. Most of these factors are due to improper driver observation leading to traffic accidents. An interesting phenomenon also emerged. Most accidents are minor cuts and scrapes that do not cause severe damage to cars or pedestrians. Both drivers (or one party) choose to settle or deal with the accident themselves and do not report it to the insurance company or the police. This can lead to the fact that the accidents we can learn about through information are often severe, but serious accidents are a tiny percentage of all accidents. Many more minor accidents are not reported or complained about by most drivers, which we cannot find out from statistics. This was not only confirmed in the questionnaire but is also very common. Most minor accidents are caused by blind spots (lack of observation) in cars. In the next section, we look at how drivers currently deal with blind spots in their vehicles. Since LiDAR is expensive equipment then it may only possible to equip some cars with it. The research reported in this thesis is inspired by the future of vehicles as depicted in science fiction films, convolutional neural algorithms detect the environment in the blind spot when an obstacle is present. When an obstacle had detected, marked on the screen with a red box, the type of obstacle is indicated to alert the driver.

The detection algorithm is based on a convolutional neural network (CNN) implementation. We begin with a literature review of convolutional neural networks while the concept of AI deep learning is explained and summarised. Then, the types of algorithms currently widely used by us are listed. Finally, examples are given to analyse the algorithmic process and the basic framework, and their advantages and disadvantages are summarised. Among them, the YOLO algorithm which is the central part of this thesis that we explore and apply without damage to property or injury to persons.

Chapter 3 3D Urban Modeling

3.1 Introduction

This Chapter is a research report on the virtual environment construction part of the project. In section 3.2, there is an overview of the Virtual City environment, a discussion of the reasons for our experiments with the virtual environment and a description of the software requirements and software selection needed to create the environment. Section 3.3 shows the lengthy process of creating the virtual environment and summary the available options which need to be considered in creating the environment. The data collection is significant in virtual environments. In the simulation tests, it is necessary to recreate as much as possible the realistic scenario, for which we did much fieldwork and finalised the initial environment model. Section 3.4 is about the statistics of the virtual environment. At last, section 3.5 concludes the 3d city modelling session.

3.2 Virtual Scenarios 3D Urban

In Chapter 2, we presented the idea of using an artificial intelligence recognition algorithm to detect blind spots in cars. We intend to test it in the initial phase with a virtual environment.

3.2.1 Virtual Versus and Real Scenes

The development of AI deep learning and neural networks requires large amounts of data to be trained and collected, but it is challenging and time-consuming to collect large amounts of data in real life without damage to property or injury to persons. Thus, conducting preliminary testing of projects in a virtual environment is one of the main ways to solve this problem (Liao, Song, Long. 2020). Figure 3.1 Virtual Versus Testing. Using randomly generated 2D images to test various tasks is one of the essential sources of test objectives, such as the generation of human poses and automated face generation (Zhu, Huang, Shi, 2017). 3D modelling of virtual scenes or 3D models for product testing or project inspection has had some recent success and is gaining traction in many areas (Varol, Romero, Martin, 2017), which are used for indoor or outdoor scene construction (Ros, Sellart, Materzynska, 2016), object detection (Hinterstoisser, Pauly Heibel H. 2019)

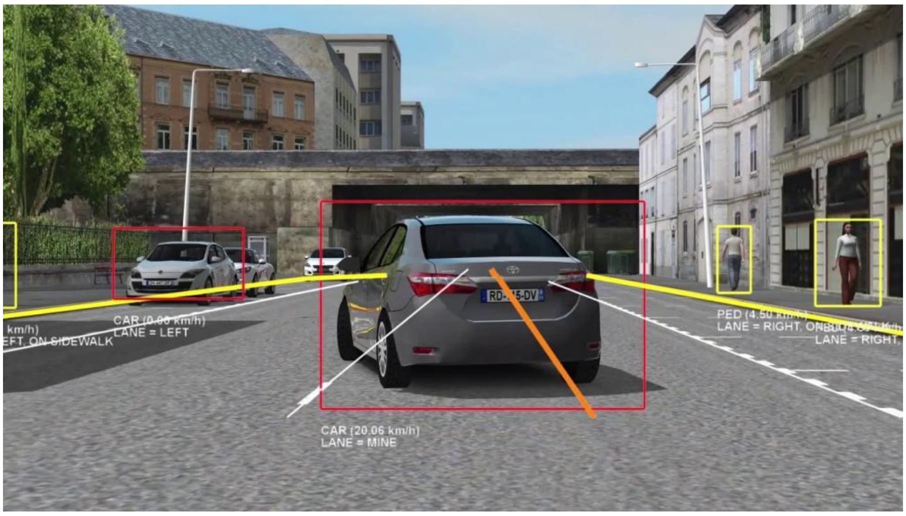


Figure 3.1 Virtual versus testing (Mérac, B.R.du 2021)

Virtual scenarios allow arbitrary changes to be made to the test environment and data to be collected (Makransky, Borre‐Gude, Mayer,2019). Virtual scenarios allow for testing unexpected situations that are impossible in reality and would often result in considerable casualties in real life. But virtual scenarios are a complete no-brainer (Lin, Guo, Shao, Chen Jiang, Zhu.2016). One of the main objectives of project testing is to deal with unforeseen situations. In real life, blind spots in cars often lead to accidents, and drivers avoid putting themselves in the blind spots of other vehicles for long periods on real roads. Deliberately creating blind spots can lead to severe accidents, so testing blind spots on existing roads is dangerous and undesirable which can lead to damage to property or injury to persons. Creating a virtual scene, creating blind spots in the virtual scene and performing blind spot detection are how we prepare for the test.

3.2.2 Software Application

This section focuses on the software used to build the 3D environment, including Unity 3D and Maya. We plan to use Maya software to create 3D modules, combine many different types of 3D building modules into a city model using Unity 3D and render them, and finally simulate and test the movement of a car model in a 3D environment.

Maya

Maya Figure3.2 is a 3D computer graphics software that is widely used in the entertainment industry for creating interactive 3D animations, models, simulations, and visual effects for film, video games, and television (Derakhshani, 2012).

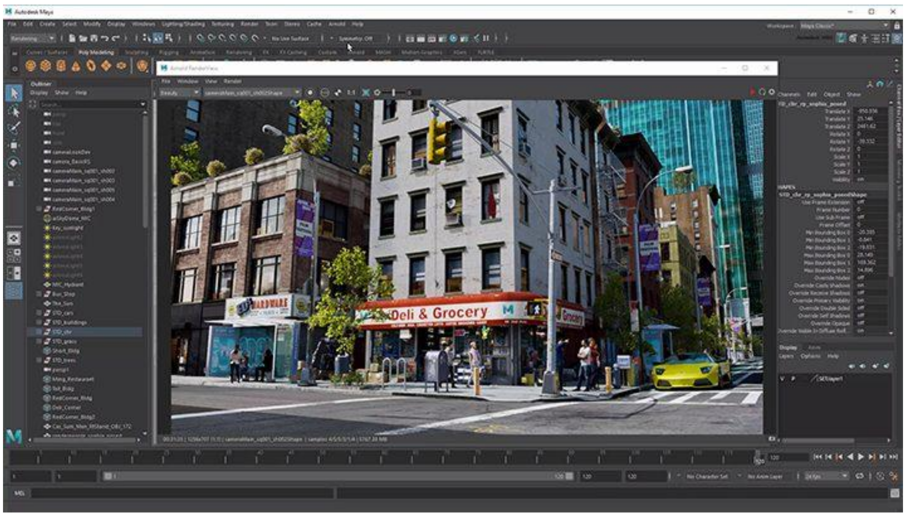


Figure 3.2 Maya interface(From Google Image)

Although the Unity3D engine is mighty, the manipulation and detailing of the model are more comprehensive in Maya than in pure model building (Liu, Li, 2011). Therefore, we use Maya for single-building construction and Unity3D for combined extensions.

Unity 3D

Unity 3D is our main production engine for building 3D virtual city environments. An introduction to the Unity3D engine is as the following walkthroughs.

Unity3D is a cross-platform game engine and development environment for creating 2D, 3D, AR and VR experiences. It is a popular choice for game developers as it offers various tools for creating interactive experiences, including a visual editor, physics engine, scripting capabilities and more (Messaoudi, Simon, and Ksentini, 2015). Unity 3D is a popular game engine and development environment that can be used to create a wide range of applications, including:

1. Video Games: Unity is widely used for creating 2D and 3D games across multiple platforms, including PC, console, mobile, and VR/AR devices.

2. Simulation and Training: Unity is used to create simulations and training programs for various industries, such as aerospace, construction, and medical.

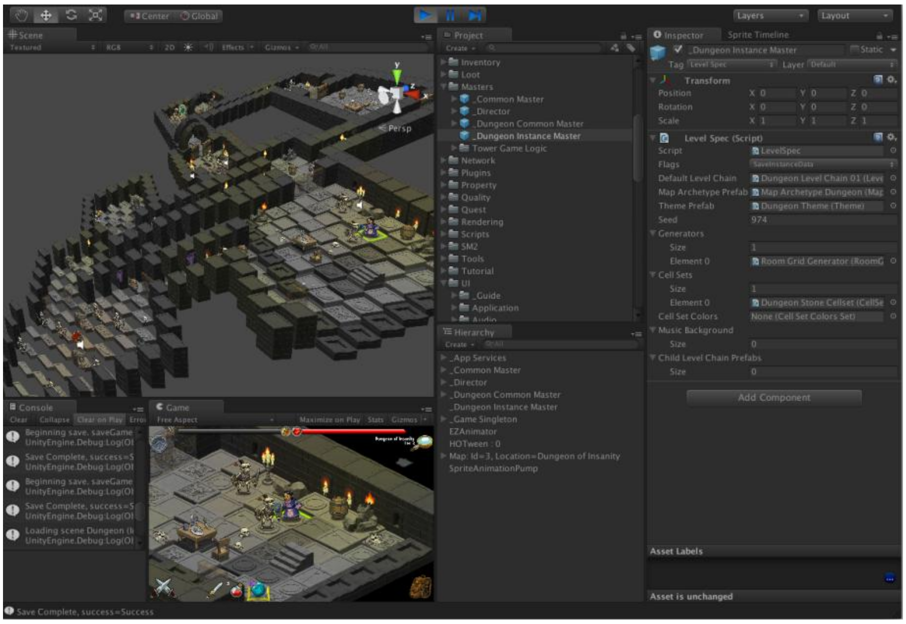
3. Architecture Vizualisation (Foffa, 2022.) and Real-time 3D: Unity is used to create real-time 3D visualization, architectural walkthroughs, and interactive product demonstrations.

4. Education and Research: Unity is used in educational settings to create interactive learning experiences and in research to create virtual environments for testing.

5. Advertising and Promotion: Unity is used to create interactive ads, promotions, and visualizations for the web and mobile devices.

6. Film and Animation: Unity is used to create short films, animated movies, and special effects for the film industry.

Overall, Unity 3D is widely used across different industries to create interactive and engaging experiences (Bartneck, Soucy, Fleuret, and Sandova, 2018).



_Dungeon Instance Master Static
Tag Level Spec Layer Default
Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Level Spec (Script)
Script LevelSpec
Flags SaveInstanceData
Default Level Chain Dungeon Level Chain 01 (Leve
Map Archetype Prefab Map Archetype Dungeon (Map
Theme Prefab Dungeon Theme (Theme)
Seed 974
Generators
Size 1
Element 0 Room Grid Generator (RoomG
Cell Sets
Size 1
Element 0 Dungeon Stone Cellset (CellSe
Cell Set Colors None (Cell Set Colors Set)
Music Background
Size 0
Child Level Chain Prefabs
Size 0

```
Beginning save. saveGame
UnityEngine.Debug:Log(Ol
Save Complete, success=S
UnityEngine.Debug:Log(Ol
Beginning save. saveGame
UnityEngine.Debug:Log(Ol
Save Complete, success=S
UnityEngine.Debug:Log(Ol
Loading scene Dungeon (I
UnityEngine.Debug:Log(Ol
Save Complete, success=Success
```

Figure 3.3 Unity 3D interface

In the test scenario, we planed the car model drive generally on a defined road, then make a traffic accident where a collision has occurred because another car blocked the driver's view or because of the car's blind spot. A camera is then set up from point of view of the accident vehicle to record the entire accident. The recorded video is saved and identified using the YOLOV5 algorithm. Unity3D can quickly solve this problem by using the object motion command module. In addition, Unity3D can also solve the problem of video recording. We can set up a camera anywhere and track the view of the car. Unity 3D is our main production engine for building 3D virtual city environments.

3D model architecture has been widely used in the last few years. It has a significant role in urban planning, simulated crash tests and other projects requiring target visualisation (Brenner, 2001). Roads, traffic signs, buildings, and trees are the main components of the virtual environment. Therefore, the alignment of buildings and traffic signs is essential, as is the accurate distribution of traffic roads and the correct placement of car models (Liu, Wang 2008). 3D city modelling is a lengthy process. Much time was spent on 3D model construction and virtual city layout to ensure accurate results.

3.3 Specific Requirements for Virtual Scenarios

Building a 3D city scene is a complex and lengthy process, and our team's scene requirements was first identified before starting to develop it.

Specific requirements for virtual scenarios for traffic testing depend on the type of simulation being performed, but some common requirements include:

1. Detailed road network: A detailed and accurate representation of the road network is essential for traffic simulations. This includes information about the layout of roads, traffic signs, road markings, road furniture and signals, and terrain features.
2. Vehicle models: The simulation should include detailed models of the vehicles that will be tested, including information about their size, weight, and performance characteristics.
3. Traffic flow data: The simulation should include accurate traffic flow data to reflect the expected conditions on the road network. This includes information about traffic volume, speeds, and patterns of movement.
4. Hardware and Software: High-performance computer with specialized software and hardware is required for processing large amounts of data and running the simulation in real-time.
5. Validation: The simulation should be validated using real-world data, such as traffic counts and speed measurements, to ensure its accuracy and reliability.

For our test-specific project, narrow sections of roads and junctions needed to be layed out that would encourage a more excellent range of blind spots for drivers. For the layout of the entire urban road, we designed and built mainly based on the two lanes single carriageway (Taylor, Baruya, 2002) Figure 3.4. We also added a small amount of single way Figure 3.5. Creating a 3D city scene is a complex and lengthy process, and we first identified our team's scene requirements before starting to develop it.

In our plans for the 3D environment build, we also optimised the details of the roads and the city's layout by adding road signs and street clutter, railings, and road construction sites, whitch tried to make 3D urban environment closer to the real-life road by bringing the test results closer to reality. Figure 3.6 Road Construction Sites and Figure 3.7 Junction Details are presented.

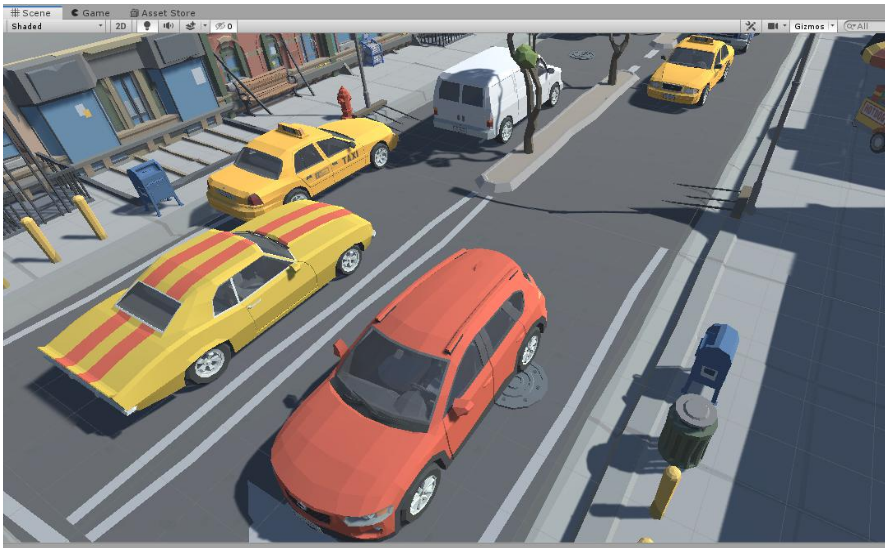


Figure 3.4 Two Lanes single carriageway

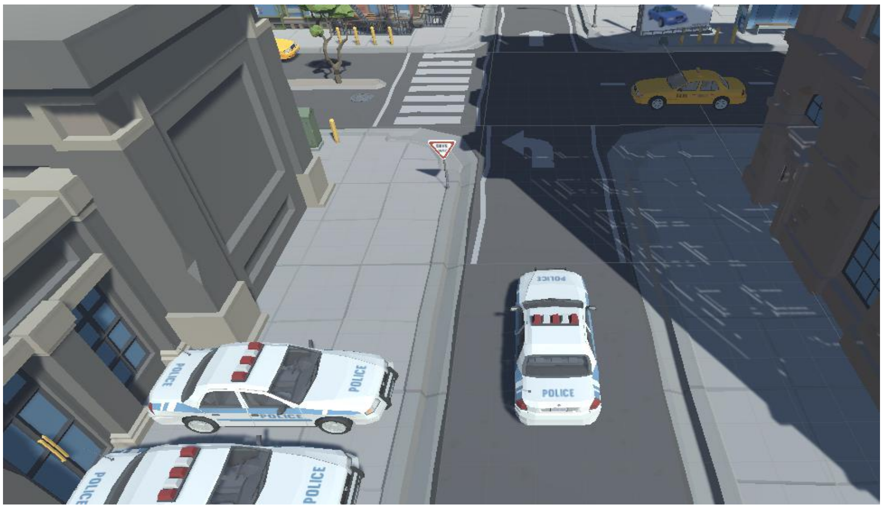


Figure 3.5 Single way

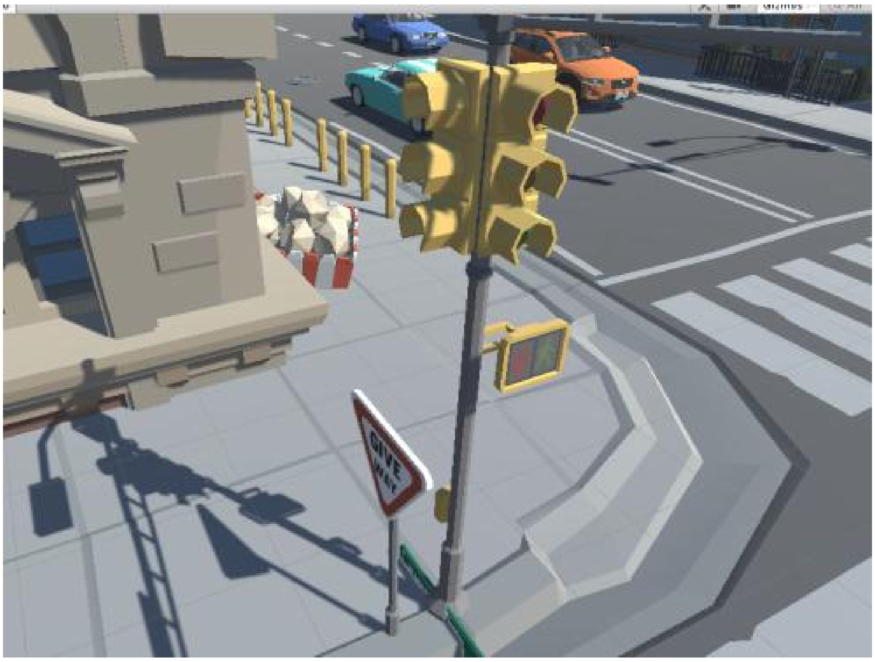


Figure 3.6 Road construction Sites

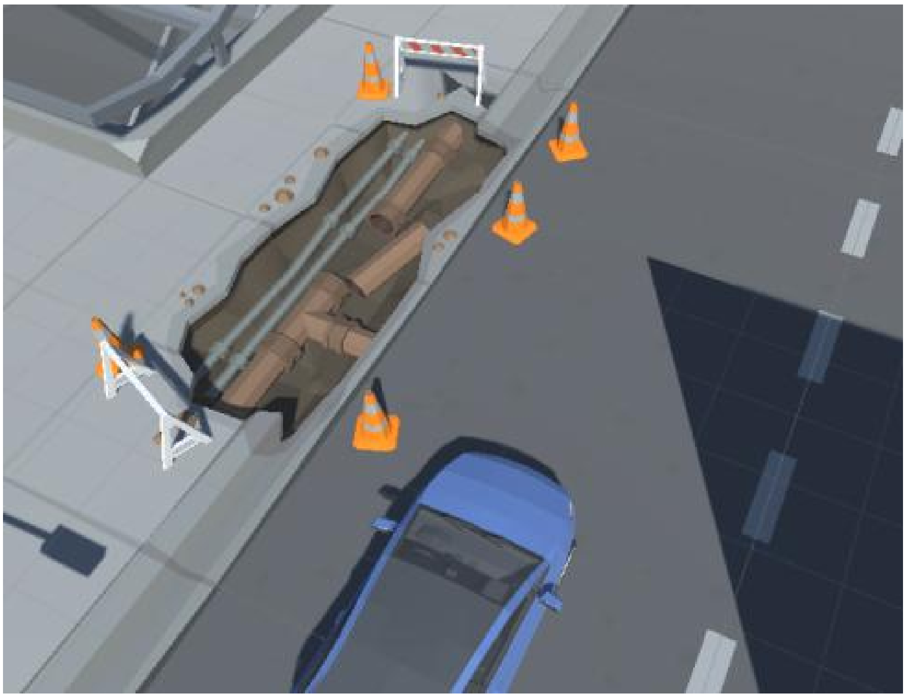


Figure 3.7 Junction details

3.3.1 Modeling Progress

Maya Part

3D modeling is a complex and demanding discipline requiring a combination of technical skill, artistic ability, and creativity (Huang, Lin, 2017). 3D modelling takes a lot of time to tweak the model, so we started with a single building model in the initial stages. First, we used Maya to design and create a simple model. Figure 3.8 combines two simple model buildings to form a small structure, mainly used to fill the gaps at the city's edge

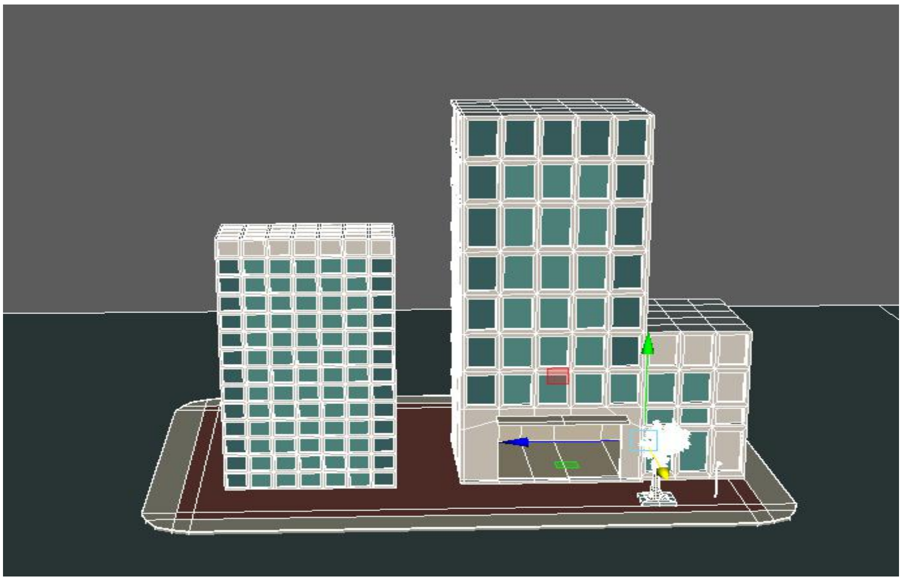


Figure 3.8 Single building

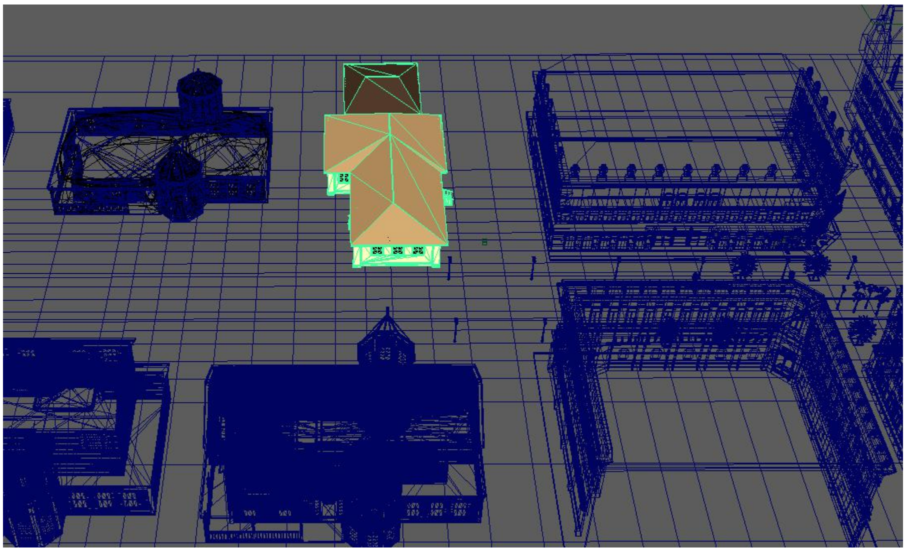


Figure 3.9 House area model

Figure 3.9 shows several architectural models of residential areas. These include general residential and educational buildings. Figure 3.10 shows the design of the building frontage details, which include the roadway trees and the base street lights.

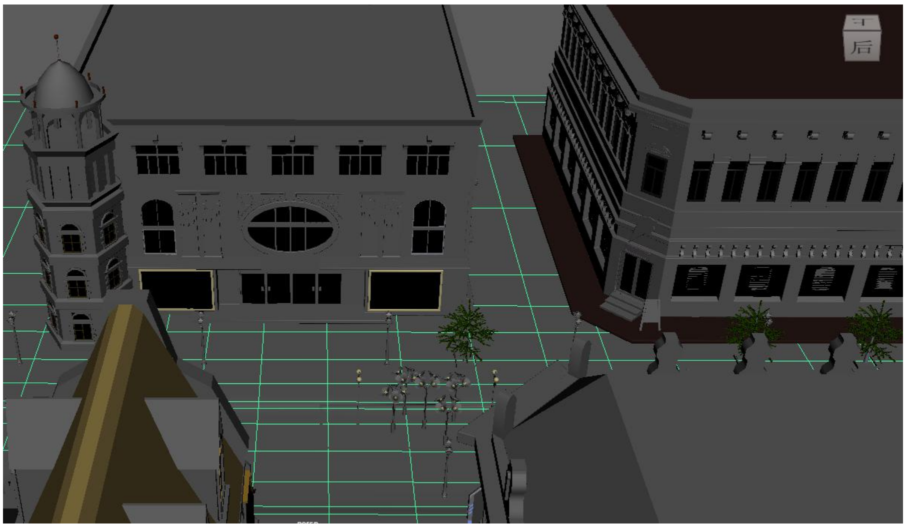


Figure 3.10 Building front view

3D modelling takes a lot of time, so we used a laptop and desktop computer to work together to speed up the early stages of the modelling process and assemble all the models. Although the two computers use the same Maya software, the initial presets have different system default length units due to the differences in computer systems and software versions. As a result, when we tried to put the models together, there was a significant deviation in the dimensions of the two sets, as shown in Figure 3.11

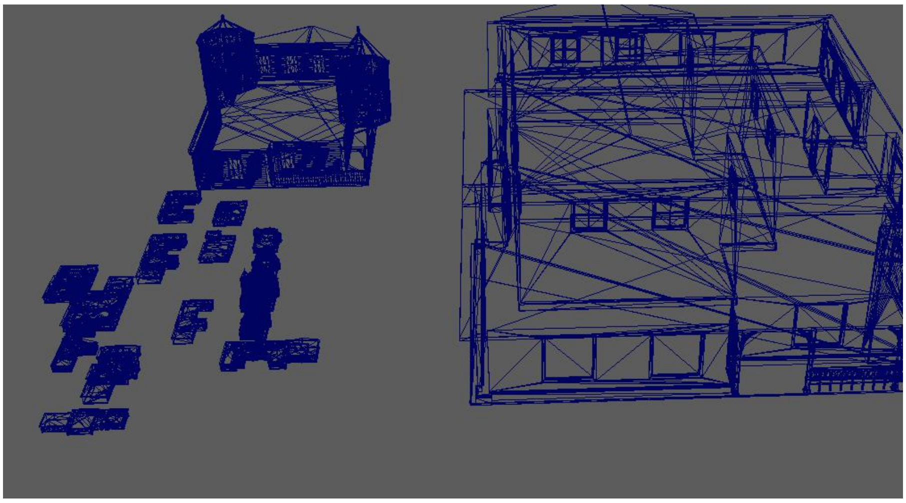


Figure 3.11 Unit deviation

Although the deviation in length units can be adjusted by zooming in and out in Maya, as each building has many differently shaped modules, selecting and scaling multiple modules simultaneously can cause the system to lag or even crash. As a result, building by building adjustments had to be made, which took a lot of time. During the subsequent modelling process the software version was updated to ensure all 3D models were in uniform length units.

3D city layout design involves creating a 3-dimensional representation of a city's physical structure, including buildings, roads, public spaces, and other urban elements. The design process may involve several stages, including gathering information about the city, conceiving a detailed 3D model, and refining the design through feedback and iteration (Rohil, and Ashok, 2022). 3D city layout design is often used in urban planning, architectural design, and virtual simulation, among other fields, to help visualize and understand the complex interactions between various urban element.

The 3D scene plays a vital role in machine learning, and rationalising the occupancy ratio of the grid in the 3D scene to bring the 3D scene closer to reality is beneficial in obtaining more accurate data (Thrun, Burgard, 2005). The road planning of the 3D city is also one of the issues that needs to be considered. Figure 3.12 shows the top view of the 3D city model, and Figure 3.13 shows the preliminary city model from different angles.

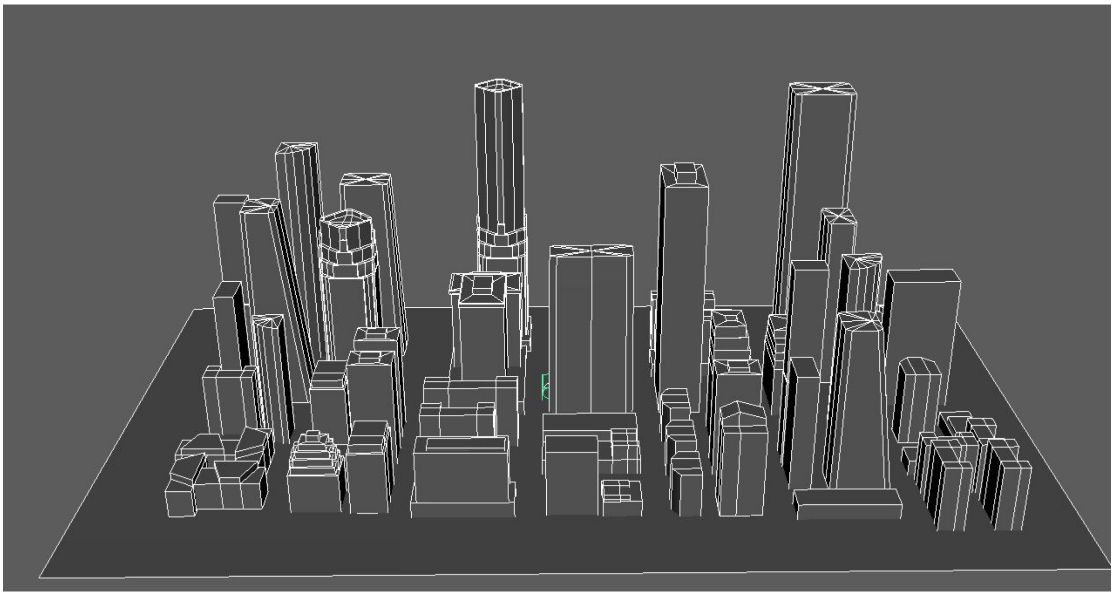


Figure 3.12 Top view of the 3D city model

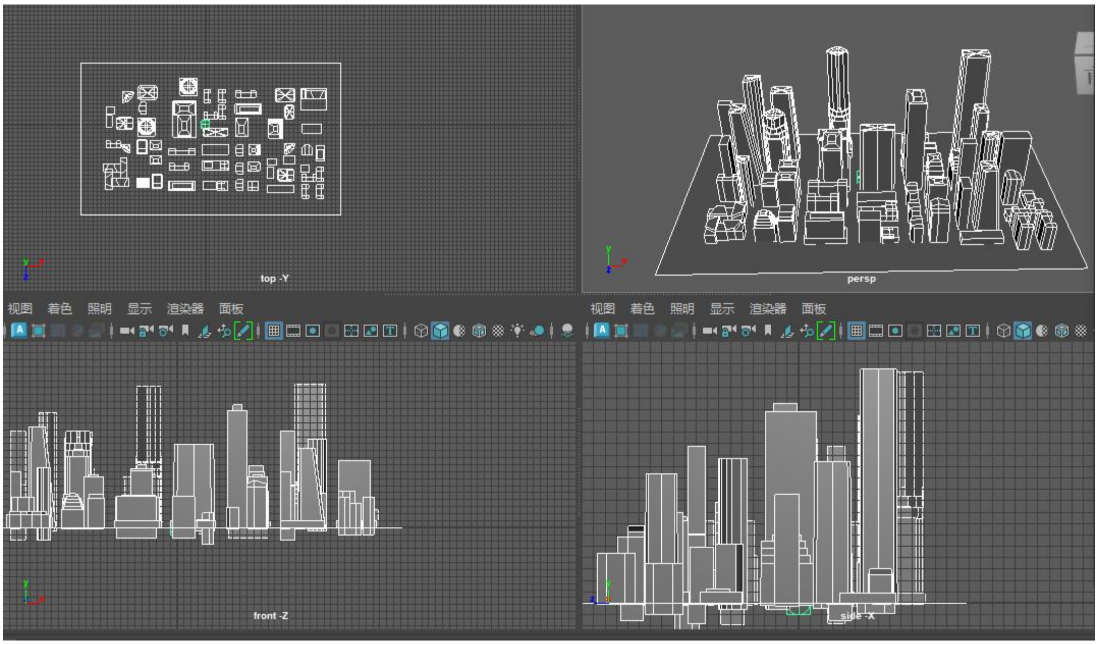

top -Y
persp
视图 着色 照明 显示 渲染器 面板
视图 着色 照明 显示 渲染器 面板
front -Z
side -X

Figure 3.13 Different view of 3D city mode

Unity 3D Part

After completing the initial building model and the basic 3D city framework in Maya, Unity 3D is used to optimize the details and the city road layout. The city model was imported from Maya into Unity. At the same time, a new arrangement was made for all the buildings. The residential buildings that had previously been made were added as shown in figure 3.14. However, the initial version of the city layout had many flaws; insufficient roads were reserved, and it was challenging to plan and fit the roads for testing.

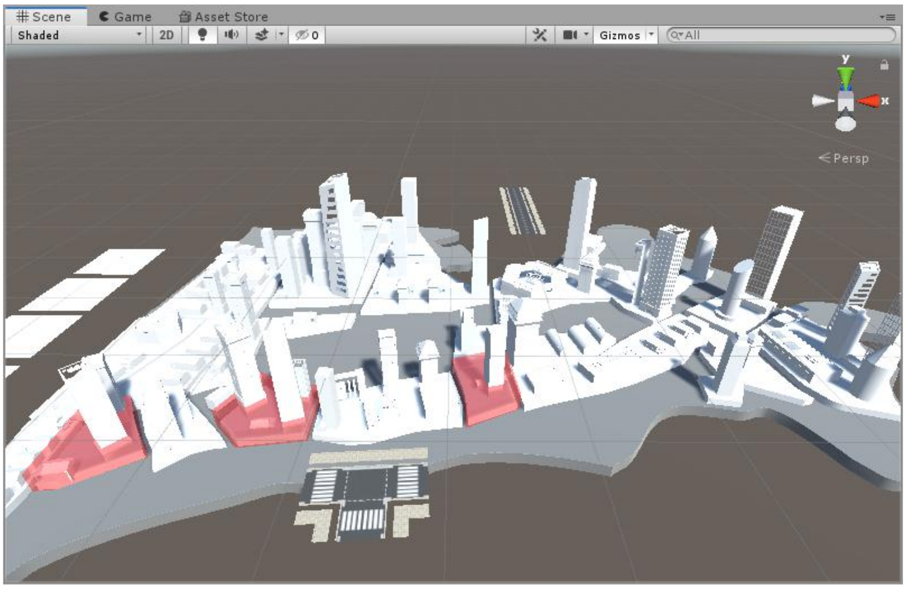


Figure 3.14 Unity3D urban first version

The city's layout was redesigned by removing low-rise residential areas and making substantial changes to the streets and buildings, considering the details of the London town plan. Details of the changes are shown in Figure3.15 and Figure 3.16

1. The city's edges have been redesigned in 3D to create enclosed areas, giving the city a more concentrated look.
2. The distance between each building has been increased.
3. Different types of roads have been added
4. Redesigned road elements such as traffic signs have been introduced

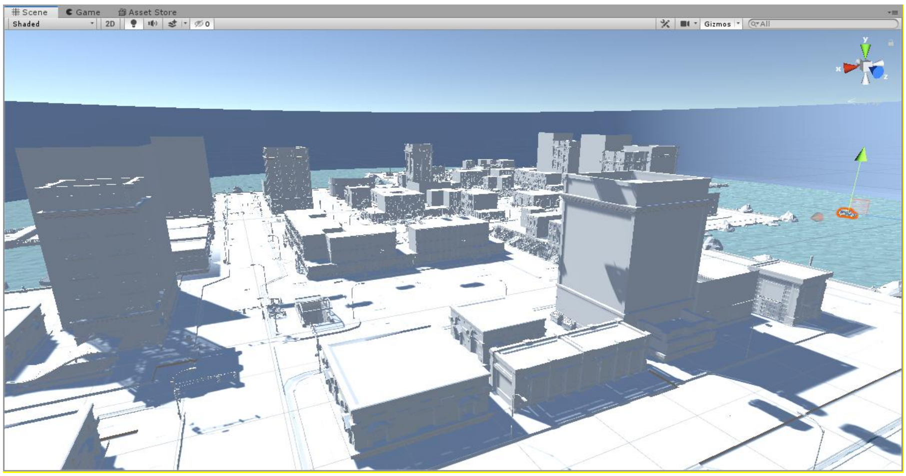


Figure 3.15 Urban street

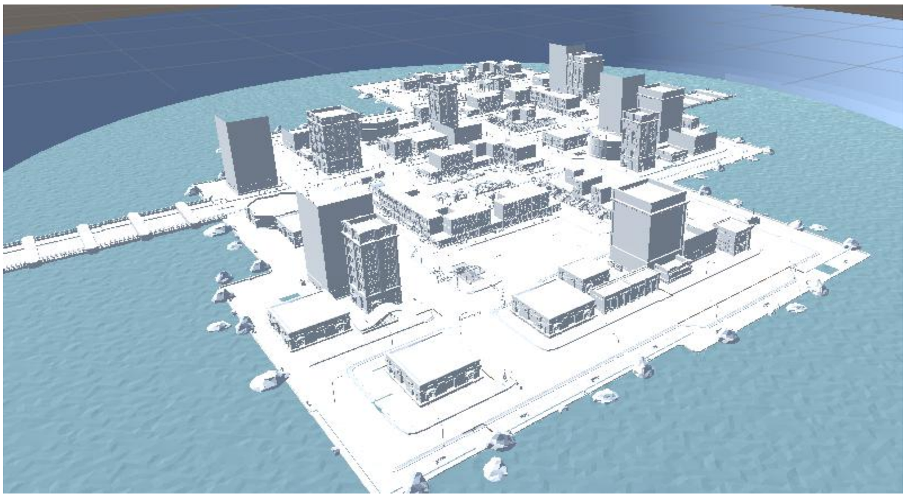


Figure 3.16 Urban overview

Our virtual urban road planning not only refers to reality, but also delves into road grid design (Cipriani, E., Gemma, A. Nigro, M., 2014). Urban road planning is not limited to the connection between different areas, but also includes the matching of roads to community needs (Cantarella, G. E., Pavone, G. and Vitetta, A, 2006). During the creation of our virtual environment, the layout of the roads and the different types of roads were the main requirements for testing the project. Different types of roads were designed at different locations, for example, single or dual carriageways.

Once the architectural framework of the city and the layout of the roads have been determined, adding car models is essential. Therefore, we have selected standard car models to add to the virtual city, including large vehicles and ordinary cars. Figure 3.17 shows several common types of small and large vehicles.

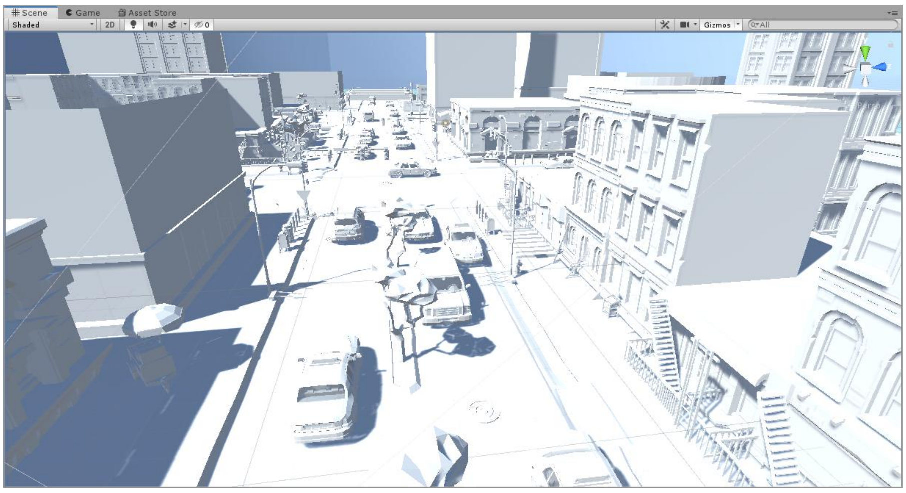


Figure 3.17 Urban overview

Materials and decals

Unity decals, which are materials that decorate the surface of other materials, are a way to add details to 3D objects in Unity, such as adding logos, text, or bullet holes. They are flat 2D images or textures projected onto an object's surface. Decals are overlayed on top of the objects' materials and blend with the underlying material's colour, roughness, and normal maps. They can be used to enhance the realism of a scene and add visual interest (Kumar, AKumar, 2021). Decals in Unity can be created using Unity's Standard/Particle Shaders or custom shaders.

Unity3D supports using most bitmap formats as image materials, even .psd format files with layers and layer effects. However, in practice, it is not recommended to use non-universal file formats directly. Unity3D automatically performs a conversion process when importing any image material in any format. Suppose we are dealing with a non-generic format. In that case, the corresponding third-party software function will be called to perform the conversion so that the actual conversion process will be slower. Some special file formats will only be recognized if we have the corresponding software installed. For this reason, it is recommended to first save the image material as a .png file (lossless and small) using PS and then use the .png file as the image material as shown in Figure3.18

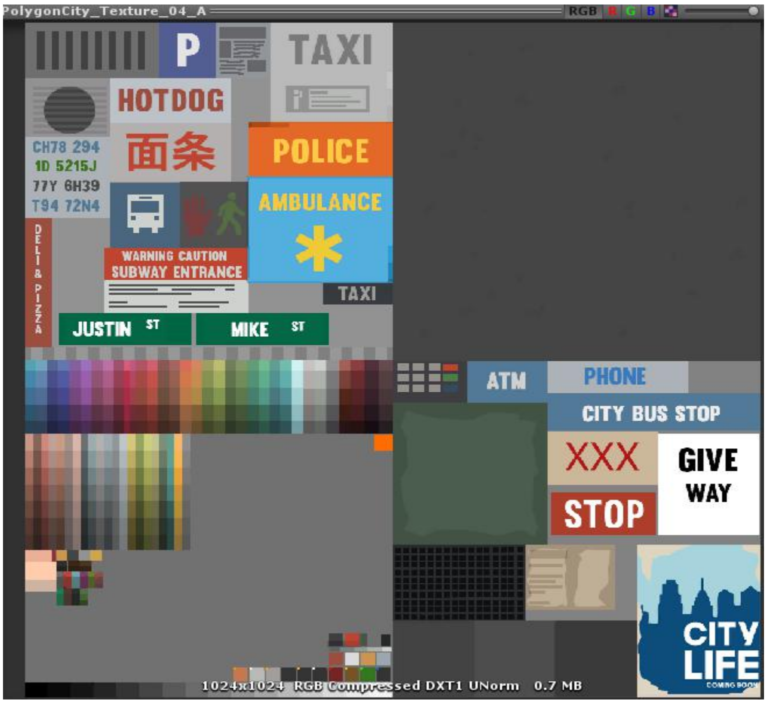


Figure 3.18 Building Decals

After much mapping, the initial effect of the city is shown in Figure 3.19. At this stage, our architectural model is complete, followed by the simulation of the car model track movements and the simulation of traffic accidents.

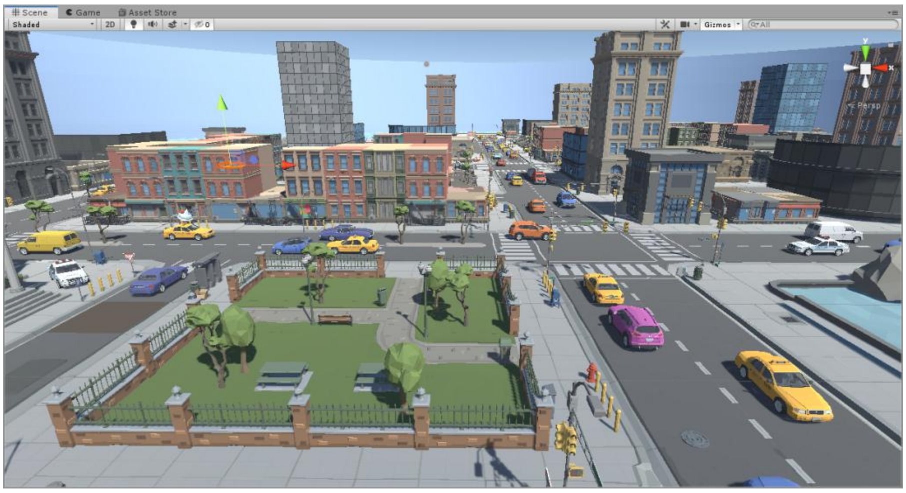


Figure 3.19 City Decals

3.4 Summary

This section focuses on the detailed process of building a 3D simulation environment. From the initial building model to the final virtual city was a very lengthy process. We had to constantly modify the pre-defined city prototype and compare it with real life to ensure that the design and distribution of roads were more realistic. At the beginning of the 3D modelling phase, we worked with different devices, and due to the lack of uniformity in the computer systems (Mac-OS and Widow's), the preset software units were different, and this led to significant discrepancies when combining the building models, which was one of the problems that affected our progress. We have reworked all the project units and are now using the Windows system to solve the problem of large deviations caused by different length units. Two versions of the city's first draft were used to finalise the final city model that we needed for the road simulation tests. We also carried out extensive research and a literature review on the distribution of roads in the city and the layout of different types of roads, identifying a variety of road divisions and layouts that correspond to reality. At last, we added many car models and coloured and mapped the 3D building models to complete the virtual city construction.

The virtual environment is based on Unity3D and is a city complex of a specific size and includes a complete road traffic system. Different areas have different types of road plans, and it is a virtual city based on a realistic environment. It can be used not only for our research projects but also for testing various virtual projects.

Chapter 4 Blind Spot Analysis of Vehicles and Virtual Scene Simulation Reproduction

4.1 Introduction

In order to obtain a more accurate picture of the actual data on the car's blind spot, a field trip was carried out in this chapter. The driver's blind spot was verified at different traffic sections using a filming and recording method, which was analysed in detail. A realistic test of the car's blind spot in the rearview mirror was carried out under controlled conditions, and the data was analysed and summarised. On the other hand, under uncontrolled conditions, a virtual scenario was recreated after filming in the field, simulating traffic accidents caused by blind overtaking in the presence of blind spots. The driver's perspective is recorded in the virtual scenario. Blind spot of vehicles was analysed and virtual scene simulation was reproduced.

Simulating traffic accidents in a virtual scenario involves creating a computer-generated environment and using software to simulate the events and consequences of a traffic accident, which can be used for research purposes, to study the effects of different variables on the outcome of accidents (Hadipriono, Duane, Nemeth, Won, 2003) without damage to property and danger to human well-being. In the subsequent progress of the project, we tried to design and guide some traffic accidents caused by traffic blindness, presenting them from the driver's point of view.

4.2 Traffic Accidents Caused by Blind Spots

Traffic accidents caused by a driver's blind spots occur when a driver cannot see other vehicles, pedestrians, or obstacles in adjacent lanes or areas around the vehicle due to the vehicle's design and the driver's limited field of vision (Thakurdesai, Aghav, 2021). These blind spots can cause accidents if the driver changes lanes or merges with traffic without being aware of other vehicles or obstacles in the blind spot.

Blind spot accidents typically involve a driver merging into the lane of another vehicle that was hidden in their blind spot, resulting in a collision. These types of accidents can also occur when a driver changes lanes and does not see a motorcycle or bicycle in their blind spot. Additionally, blind spot accidents can occur when a driver is backing up and does not see a pedestrian or an object behind their vehicle.

Traffic accidents caused by blind spots in the UK are a common occurrence. A blind spot is an area around a vehicle that the driver cannot see directly, which can lead to collisions with other road users. Blind spots are particularly dangerous when drivers are making turns or changing lanes, as they may not see other vehicles, cyclists, or pedestrians in their path. In this research thesis, the causes of blind spot accidents are discussed and how they can be prevented.

4.2.1 Causes of Blind Spot Accidents:

Blind spot accidents are caused by a number of factors, including:

1. Inadequate Mirrors: Drivers rely heavily on their rearview and side mirrors to see what is behind them. However, if these mirrors are not adjusted properly, they may not provide a clear view of the blind spot. It is important for drivers to adjust their mirrors correctly before driving.
2. Vehicle Design: Some vehicles have larger blind spots than others. For example, commercial trucks have very large blind spots due to their size and shape. Drivers of these vehicles must be particularly vigilant when changing lanes or making turns.
3. Distracted Driving: Distractions, such as using a mobile phone, eating or drinking, and adjusting the radio or GPS, can divert a driver's attention from the road. This can cause them to miss other road users in their blind spot.
4. Poor Weather Conditions: Rain, fog, and snow can make it difficult for drivers to see what is around them. This can increase the risk of blind spot accidents (Summerskill, Marshall, R, Lenard, and Richardson, 2016)

4.2.2 Preventing Blind Spot Accidents

There are several ways to prevent blind spot accidents, including:

1. Adjusting Mirrors: Drivers should adjust their mirrors so that they provide a clear view of the blind spot. This can be done by positioning the mirrors correctly and using a convex mirror to provide a wider angle of view.
2. Checking Blind Spots: Drivers should check their blind spots by turning their head to look over their shoulder before changing lanes or making a turn. This is particularly important when driving in heavy traffic.
3. Using Technology: Many modern vehicles are equipped with blind spot monitoring systems that use sensors to detect other road users in the driver's blind spot (Beresnev, Zarubin, Tyugin, and Pinchin, 2022). This technology can alert the driver to potential hazards and prevent accidents.
4. Eliminating Distractions: Drivers should avoid distractions such as using a mobile phone or eating while driving (Alosco, Spitznagel, Fischer, Miller, Pillai, Hughes, Gunstad, 2012). This can help them remain focused on the road and avoid missing other road users in their blind spot.

The blind spot accidents are a serious problem in the UK. They are caused by a number of factors, including inadequate mirrors, vehicle design, distracted driving, and poor weather conditions (Williamson, 2021). However, these accidents can be prevented by adjusting mirrors, checking blind spots, using technology, and eliminating distractions. By following these tips, drivers can reduce the risk of blind spot accidents and keep themselves and other road users safe.

4.3 Overtaking in Single Lanes

Blind spots can be an essential factor in overtaking accidents. A blind spot is an area around a vehicle or where a driver is unable to gather information about the environment in the blind spot due to the vehicle in front of them blocking their view. As a result, when drivers attempt to overtake another vehicle, they may not be able to see any approaching vehicles in the blind spot, which increases the risk of a collision. This is why drivers need to double-check their blind spots before attempting to overtake and to always be aware of their environment on the road. In addition, vehicles with large blind spots, such as trucks and buses, may pose a greater risk to other drivers when attempting to overtake. To help mitigate this risk, many vehicles are now equipped with advanced driver assistance technology, such as blind spot detection systems, which can alert drivers to the presence of vehicles in their blind spots. However, blind spots due to obscuration by the vehicle in front are difficult to identify and often only rely on the driver's experience to make predictions, which may be inaccurate.

Overtaking in single lanes refers to passing another vehicle on a road or highway with only one lane in each direction. This type of overtaking requires the driver to cross over the centerline into the opposing lane of traffic temporarily, making it a high-risk maneuver that must be done with caution. Before overtaking in a single lane, it is important to ensure it is safe, and that driver has a clear view of the road ahead. In our daily driving, the opposite lane is in the driver's blind spot because the car in front of us is blocking the view. Overtaking in such a situation is dangerous and can easily lead to traffic accidents. We have briefly described this situation in the previous section of the Literature Review.

We performed and recorded the data on a realistic road section. For example, in picture 4.1, the red bus was driving slow. Therefore, there is slight congestion behind the bus, and from our viewpoint, we cannot see the vehicles directly behind the bus, nor can the vehicles directly behind the red bus see us. At this point, if the car behind the numbered bus forces itself to overtake us, there is a risk of a collision with our vehicle. Figure4.2 is the view that blocked by a bus in same situation. We can not see oncoming traffic in the opposite lane . We are going to recreate this scene in a virtual scene and record it on video as shown in Figure 4.3, which is a screenshot of the video.

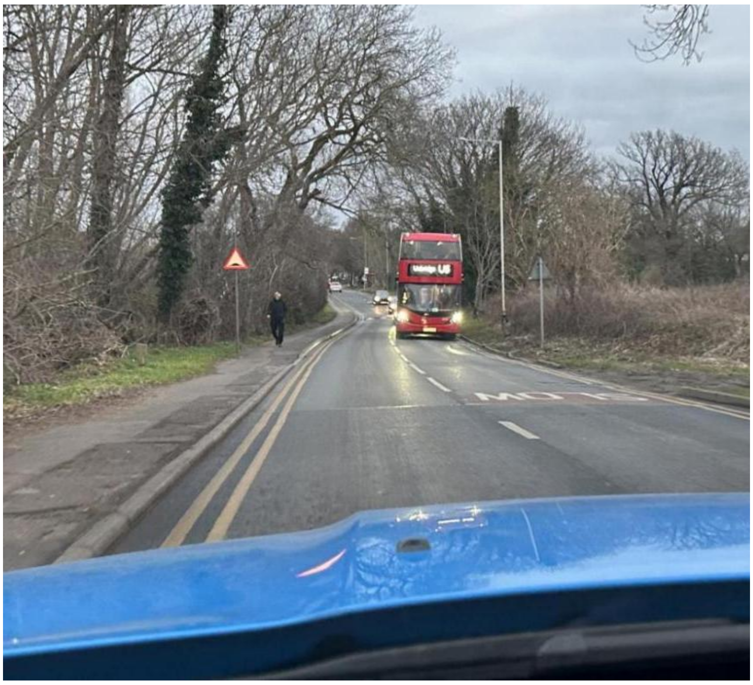
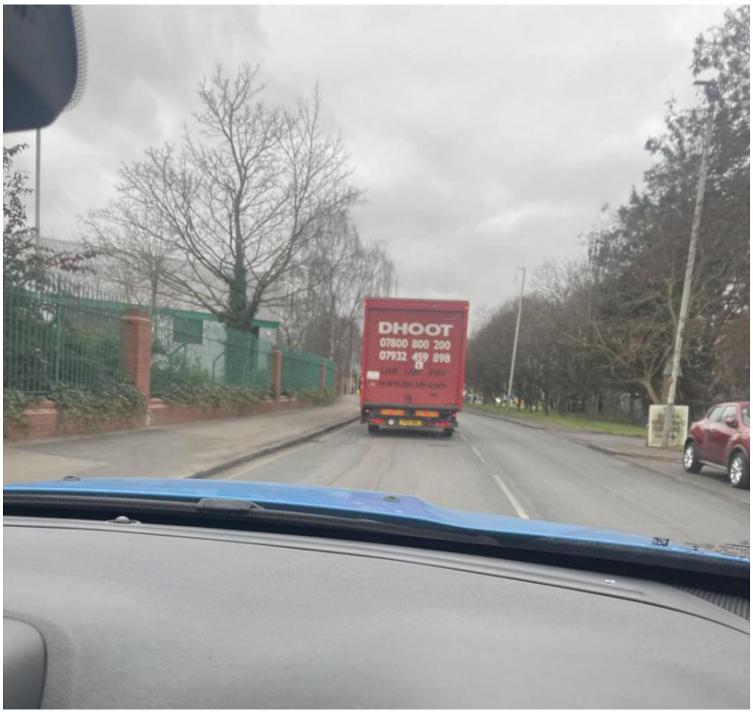


Figure 4.1 Blind spot behind the bus-front view



Fighre 4.2 Blind spot behind the bus-back view

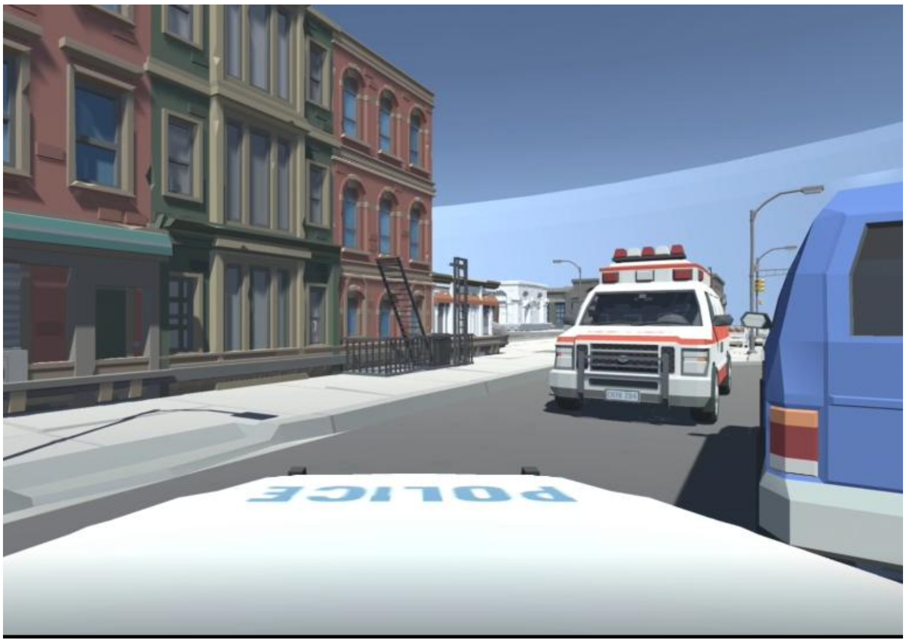


Figure 4.3 Accident by overtaking in singe lanes

Although it is hazardous to overtake when neither driver can see the other because of the cover of the car in front of them, most drivers will choose to overtake a slow-moving vehicle in front of them on particular long-distance roads.

4.4 Car drivers turning left(right) or change lanes

The blind spot for car drivers turning left (right) refers to the area behind and the side of a vehicle that is not visible in the driver's rearview or side mirrors. This is a common area where other vehicles, pedestrians, or cyclists may be located, and if the driver is not aware of their presence, it can lead to accidents. Figure 4.4 shows these driver’s blind spots.

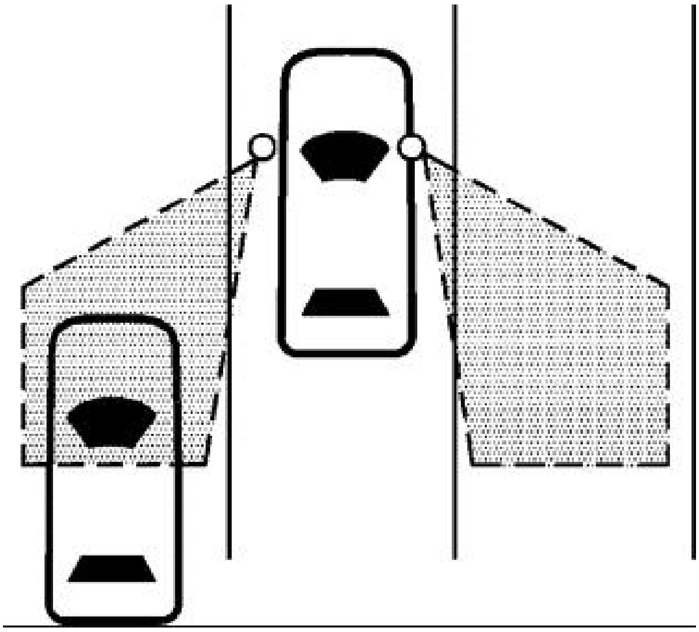


Figure 4.4 Bot side driver’s blind spots (Kiefer, R.J. and Hankey, J.M., 2008)

These blind spots pose a traffic hazard, mainly in the case of lane changes or two-lane turns at traffic junctions. A blind spot in a car refers to the areas around the vehicle that can't be seen directly through the mirrors or by looking forward (Hassan, and Zainal Ariffin, 2013). This includes the area behind the car, and to the side of the car, near the rear wheel. This can make it difficult for the driver to see other vehicles, cyclists, or pedestrians when changing lanes or reversing. In Figure 4.5 it shows the blind spot in a real-life rear-view mirror. In a standard rear-view mirror, it is almost impossible to see a car coming from behind the left side of the car. It is therefore very difficult for the driver to spot the car in the red circle without looking closely; it is by adding more mirrors that the driver can see the full outline of the car behind him on the side. However, most of the cars in our lives do not have these extra mirrors, and blind lane changes without a full view are very likely to lead to accidents.

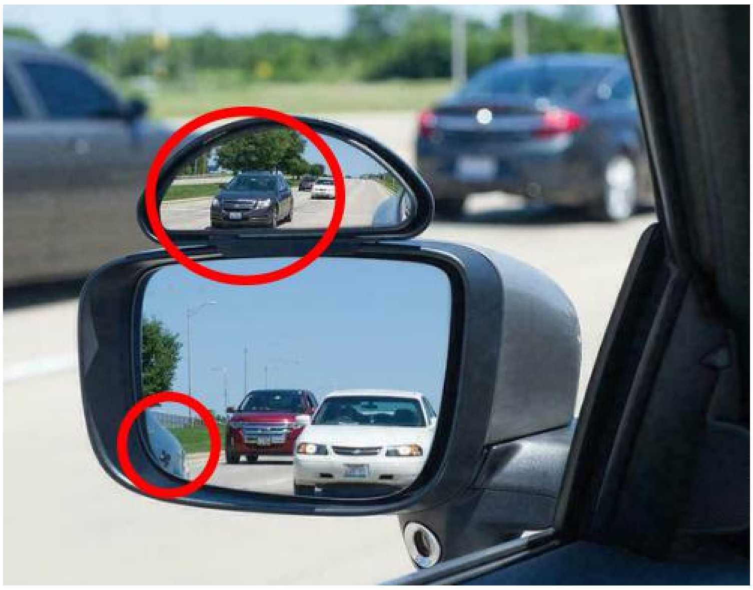


Figure 4.5 Rear-view mirror blind spot

We conducted a blind spot range test in a realistic and secure car park. The tester stands two-thirds of the length of the vehicle, 1.2 meters from the right side of the car as shown in Figure 4.6. We took a photo from the driver's view inside the car, a view that does not allow any information about the environment to be seen in the rear view mirror as shown in Figure 4.7. The view of the occupants of the car is shown in Figure 4.8.

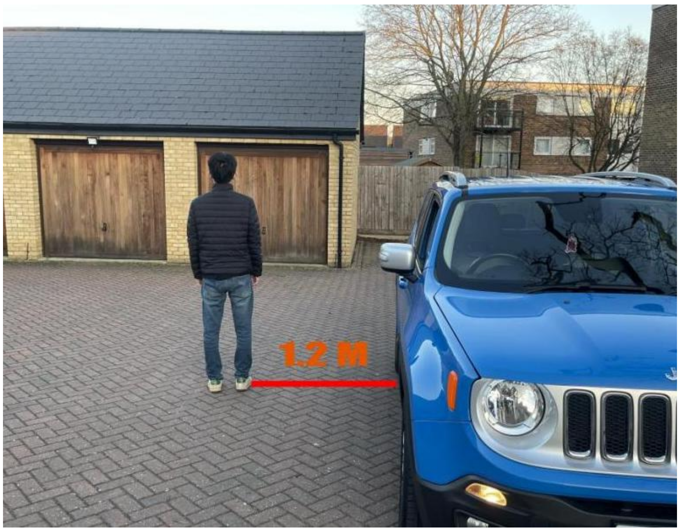


Figure 4.6 Blind spot test outside the car

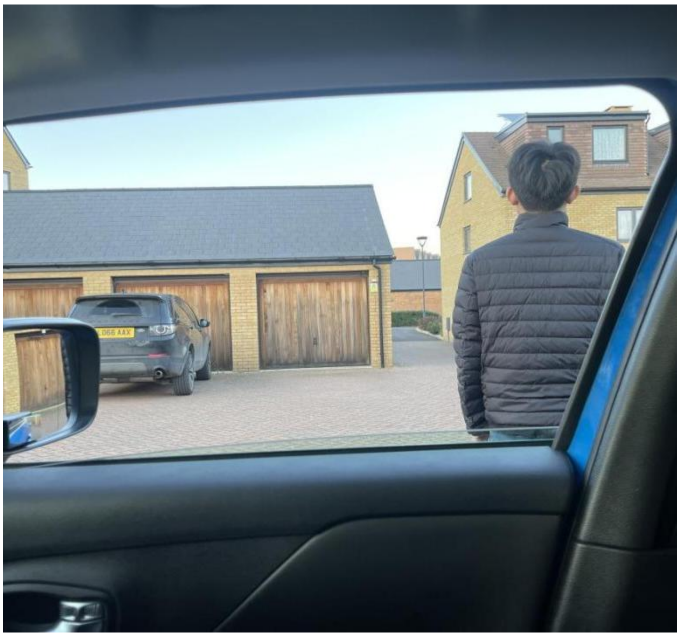


Figure 4.7 Driver's view of the rear view mirror-A

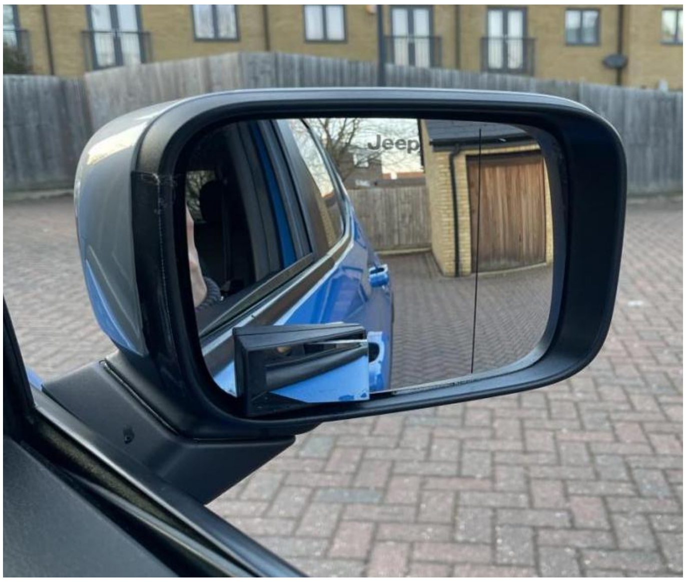


Figure 4.8 Passenger 's view

We can know the edge of the tester when the person tested is standing two-thirds of the way up the car but at a distance of fewer than 1.2 metres from the horizontal position of the car as shown in Figure 4.9.

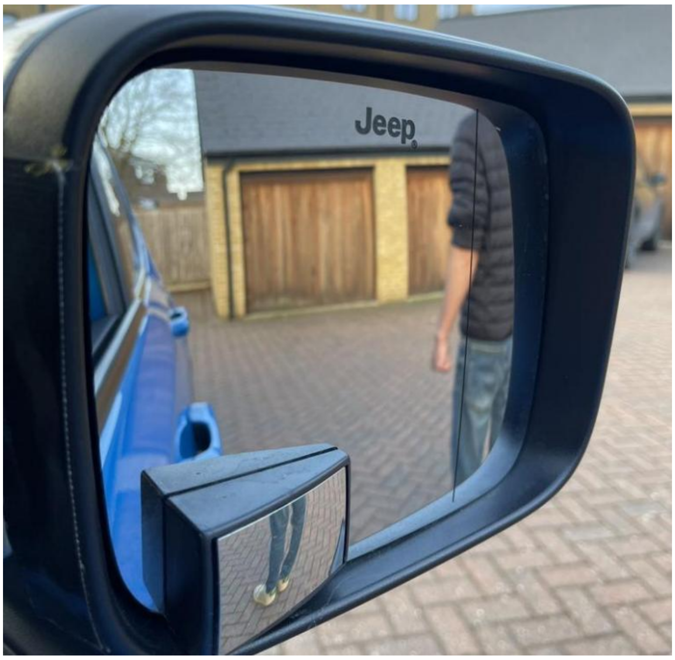


Figure 4.9 Driver's view of the rear view mirror-B

Based on the results of these field tests, we can conclude that the driver cannot obtain information about the environment through standard car mirrors in areas where the horizontal distance behind the side of the average small vehicle is more significant than approximately 1.2m. The standard UK road width for two lanes and above is 3.2m as shown in Figure 4.10. The average width of an average vehicle is 6 feet, approximately 1.83m (Design Manual for Roads and Bridges, 2021).

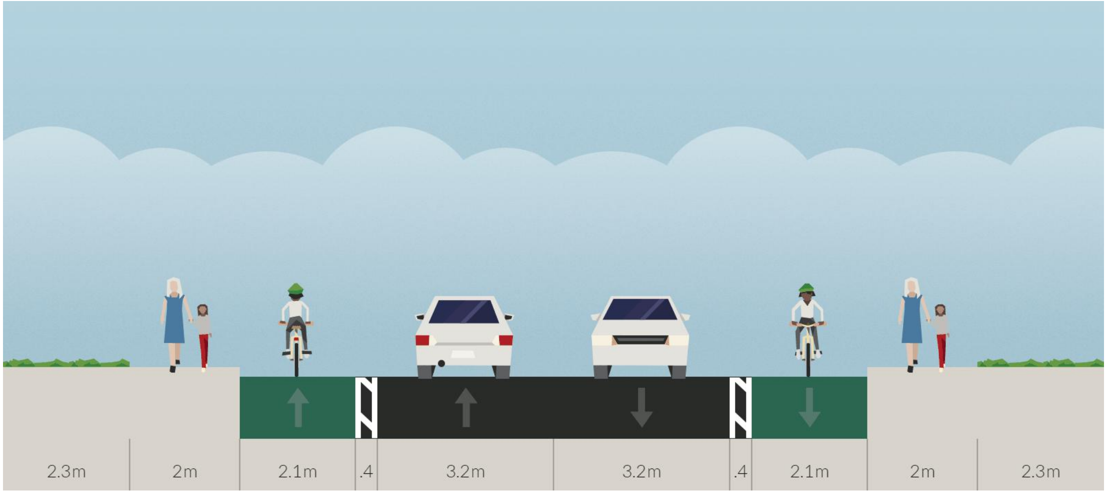


Figure 4.10 Standard UK road width(Response to Milton Rd and Histon Rd consultations 2017)

Assuming that both vehicles normally travel in the target lane, we know from the picture that the total width of the dual carriageway is 6.4 meters. We can calculate that the distance between the two vehicles parallel to each other is approximately 1.37 meters. From the results of our tests, a distance of more than 1.2 meters makes it impossible to observe oncoming traffic to the side and behind. It is therefore difficult for drivers to observe the vehicles to the side and behind them through standard car mirrors on a standard road.

This phenomenon occurs not only when changing lanes but also when both lanes can turn left or right because the driver in the outside lane cannot see the vehicle in the blind spot on the inside lane. This situation is even more common in large vehicles. Large vehicles turn at a greater angle than the average vehicle and may occupy other lanes when turning, which dramatically increases the probability of a traffic accident occurring without the driver being able to detect it. We simulated and recorded this situation, and the video screen shot in Figure 4.11 shows the traffic accident.

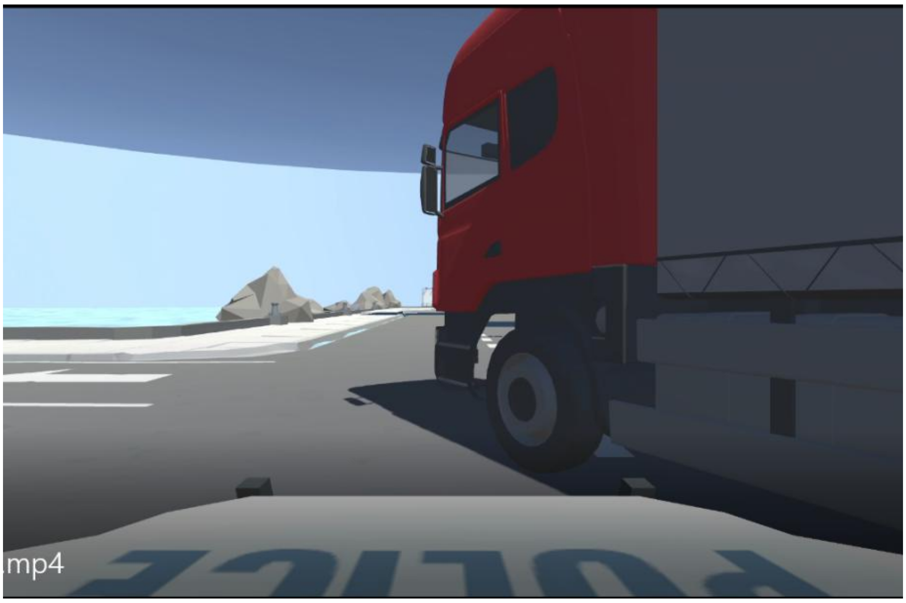
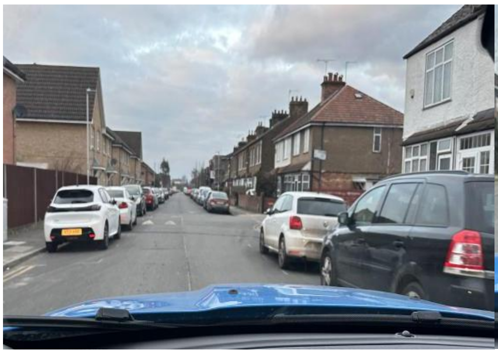


Figure 4.11 Blind spot for car drivers turning left

4.5 Both Vehicles in Blind Spot

At intersections or on narrow roads, not only is the target vehicle in the driver's blind spot, but the target vehicle is also often in the blind spot of other vehicles. When neither vehicle can see the other, it is straight forward to cause an accident, which is often the case in narrow neighbourhood roads, where cars parked on either side of the road obstruct the view when approaching an intersection.

Figure 4.12 and Figure 4.13 illustrate the data we have collected in real life.



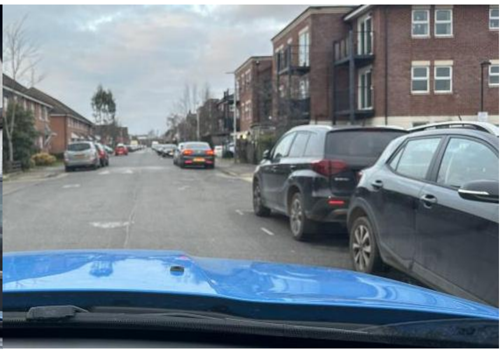


Figure 4.12 **Figure 4.13**

In both test photos, the car parked to the right of the vehicle blocks the view of the junction ahead to the right. As a result, when a vehicle pulls out in front of the right car, neither driver is able to observe the other due to the blockage of the view from the vehicle. This situation was reproduced in the test environment, where the target vehicle was unable to observe the blue car before pulling into the junction due to the blocking of the red car. Although all roads are planned as main roads or side roads, there is a need to wait and confirm that no other vehicles are passing on the main road when other vehicles are approaching. However, not all drivers strictly adhere to such rules and often only slow down slightly, making it easy for accidents to occur in such scenarios.

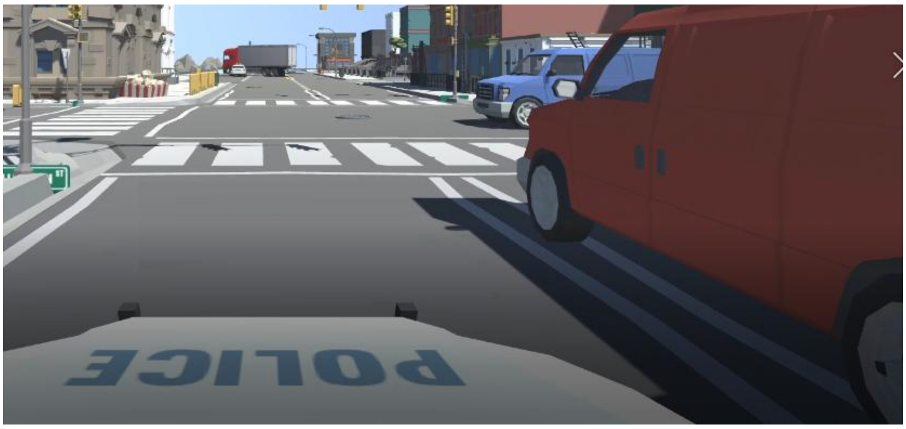


Figure 4.14 Both vehicles are in the blind spot

4.6 Blind Spots Around the Car in Low-Speed

Blind spots are areas around a vehicle where the driver's view is obstructed, making it difficult to see other vehicles, pedestrians, or obstacles. These blind spots can be especially dangerous in low-speed situations, such as when parking or driving in a crowded parking lot.

The blind spot varies depending on the size of the car, with approximately 5 feet in front of and 5 feet behind the car, with the blind spot on both sides of the car mainly behind the driver's side of the car and extending about 10 feet outwards (National

Safety Council ,2019). The range of a car's blind spot varies depending on several factors, such as the size and design of the vehicle, the height of the driver's seat, and the position of the side mirrors. However, in general, the blind spot of a car can extend up to several car lengths behind and alongside the vehicle. Most of the traffic accidents we have access to are those that have a severe impact on cars or pedestrians. However, according to the results of our questionnaire on traffic incidents at the beginning of the project, most traffic accidents are minor, minor collisions causing small scratches or bruises. These minor accidents or car hang-ups are difficult to record or count. The driver's ignorance of the road surroundings due to blind spots is one of the leading causes of these minor accidents.

Traffic jams are a common phenomenon in contemporary traffic and occur almost daily. In the London area, motorbikes often pass between cars close to each other when drivers are stuck in traffic, which makes it extremely easy for traffic hazards to occur. This is because cars do not stop in traffic. They move slowly. The driver is focused on the road ahead and always keeps a safe distance from the vehicle in front of him. A motorbike suddenly approaching from the rear of the vehicle is difficult to attract the driver's attention and will be in the driver's blind spot.

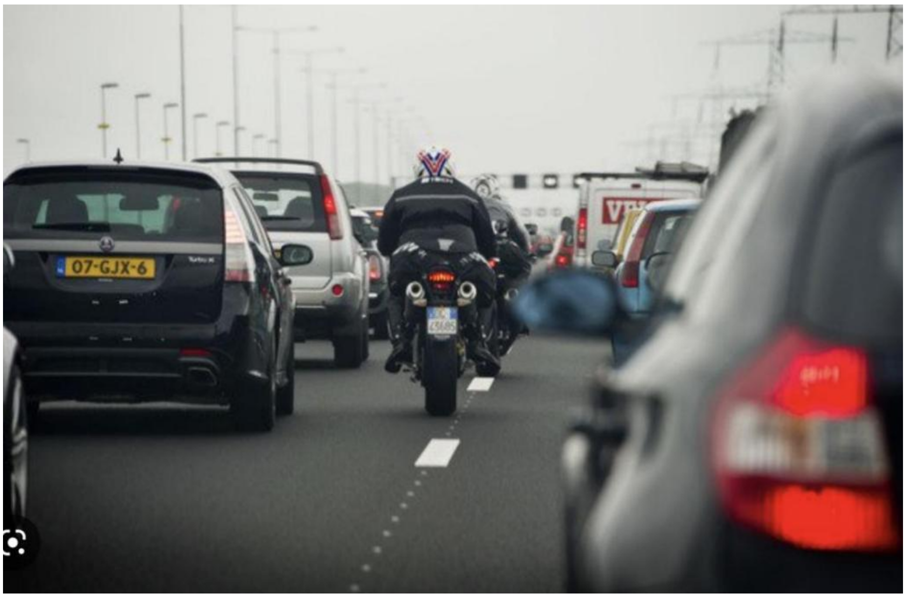


Figure 4.15 Motorbike passing between two vehicles

In image figure 4.15, if the driver of the car does not notice the motorbike coming from the rear gap and moves the vehicle directly in its way, the vehicle can easily collide with the motorbike, making even more congestion.

When turning, drivers should be aware of blind spots caused by the vehicle's A-pillars, which are the supports between the windshield and the front doors. These blind spots can make it difficult to see pedestrians or other vehicles approaching from the side.

The A-pillars of a vehicle are the vertical supports on either side of the windshield that help to hold up the roof (Obeidat, Altheeb, Momani, Theeb, 2022.). While they provide structural support to the vehicle, they can also create blind spots for the driver, particularly when turning or changing lanes.

When a driver looks forward through the windshield, the A-pillars can block their view of objects or pedestrians on either side of the vehicle. This can be particularly problematic when making turns or driving through intersections, as it can be difficult to see pedestrians or other vehicles that may be in the driver's path, as shown in Figure 4.16.

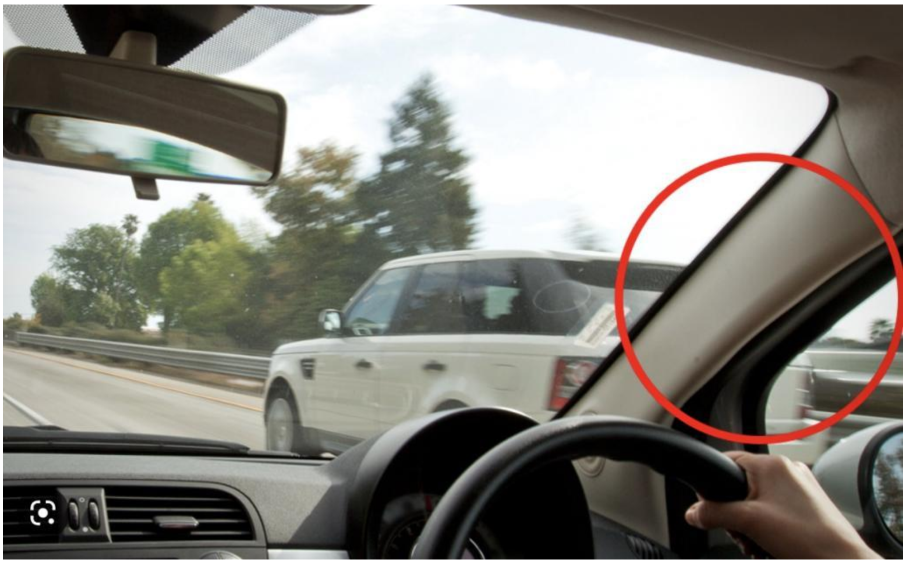


Figure 4.16 A-pillar blind spot

There are also blind spots in front of and behind the car, but these do not need to be considered when the driver is generally driving on the road, as the driver can observe the blind spot in advance from a distance and make a judgement. However, when driving at low speed in a car park or a narrow and curvy area, the driver cannot see what is about to pass over the road in advance, which can easily result in a minor collision.

4.7 Summary and Solutions

In this chapter, we have examined and verified the primary blind spots of a car in real life. The driver's blind spots, prone to traffic accidents, are recreated in a virtual environment, and the driver's perspective is recorded. These include the blind spot when overtaking in single lanes, the blind spot at junctions in residential areas due to roadside parking and the blind spot at the side and rear of the car. More field tests were conducted to measure and verify the extent of the driver's side and rear blind zones. The driver's blind spot was analysed at low speeds in cars, including motorbikes travelling between two cars in traffic jams, the car's front and rear blind spots in car parks or on narrow, curved roads, and the blind spot caused by the car's A-pillar.

In chapter 2, Literature review, we introduced traditional solutions to the driver's blind spot, including radar and auxiliary mirrors. Although these measures enhance the driver's awareness of the vehicle's surroundings, they do not entirely solve the driver's blind spot problem. We try to solve the driver's blind spot problem by using object recognition algorithms. Firstly, we intend to capture images of the car's surroundings with a camera, then use an object recognition algorithm to detect them and present the results on the in-car screen. This will improve the driver's awareness of the blind spot environment and reduce the damage caused by traffic accidents or car collisions.

Chapter 5 Object Detection - YOLO

5.1 Introduction

With the popularity and development of artificial intelligence, our ability to process image information has reached a level of deep understanding. The predicted target object's location in the image and multiple levels of culling and classification are performed to determine the target type in the image (Zhao, Zheng, Xu, and Wu, 2019.). This approach is known as a object detection algorithm (Felzenszwalb et al.2010). Object detection algorithms are computer vision algorithms designed to automatically identify and classify objects within images or videos (Fang, Love, Luo, Ding, 2020). These algorithms typically involve several stages of processing, including feature extraction, feature selection, and classification.

This chapter describes the object recognition algorithms that are currently being widely used and provides a cross-sectional comparison. The YOLO algorithm is the main target of our comprehensive research. We discuss the Yolo algorithm in depth, which includes the principles and implementation process of YOLO. Bounding Box Prediction, Loss Function, and Non-Maximum Suppression (NMS) are discussed in detail.

5.2 The Difference Between YOLO and Other Object Detection Algorithms

Object detection algorithms are computer vision algorithms that detect and locate objects within an image or video. These algorithms typically have two main components: object localization and object classification (Padilla, Netto, and Da Silva, 2020,). Object localization involves determining the location of an object within an image or video. In contrast, object classification involves identifying the type of object present in the image or video.

5.2.1 Mainstream object detection algorithms

Object detection is a computer vision task that involves identifying and localizing objects within an image or video stream (Athak, Pandey, and Rautaray, 2018). There are many different algorithms and techniques that have been developed for object detection, but some of the most popular and widely used are:

YOLO (You Only Look Once) figure 5.1: YOLO is a real-time object detection system that processes images in a single pass through a neural network (Du, 2018,). It divides the image into a grid and predicts bounding boxes and class probabilities for each grid cell (Morbekar, Parihar, and Jadhav, 2020). YOLO is known for its speed and accuracy, and it is widely used in applications such as self-driving cars and security systems.

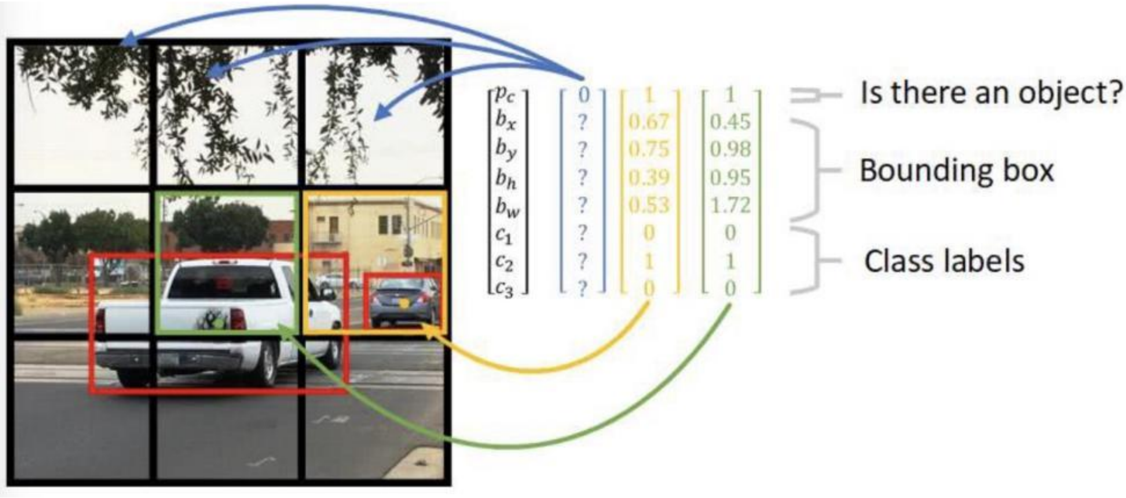


Figure 5.1 YOLO detection algorithms

Faster R-CNN (Region-based Convolutional Neural Network) figure 5.2: Faster R-CNN is a two-stage object detection system that uses a region proposal network to generate candidate regions for objects and then uses a convolutional neural network to classify and refine the bounding boxes for those objects (Jonathan Hui, 2017). Faster R-CNN is known for its accuracy and is commonly used in applications such as medical imaging and autonomous vehicles.

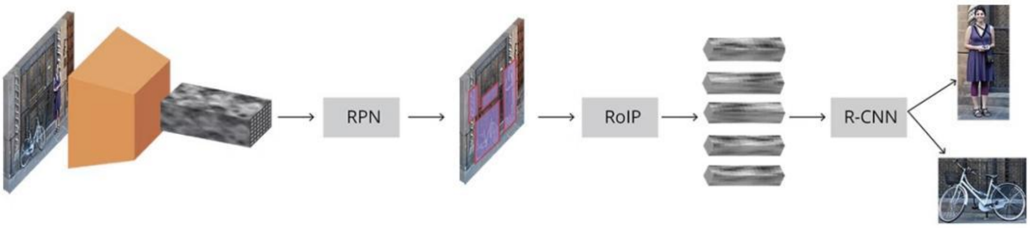


Figure 5.2 Fast R-CNN algorithm (Jonathan Hui, 2017)

SSD (Single Shot Detector. Figure 5.3): SSD is a real-time object detection system which uses a single neural network to predict bounding boxes and class probabilities for objects at multiple scales and aspect ratios (Valiati and Menotti, 2018). SSD is known for its simplicity and speed, and it is commonly used in applications such as robotics and video surveillance.

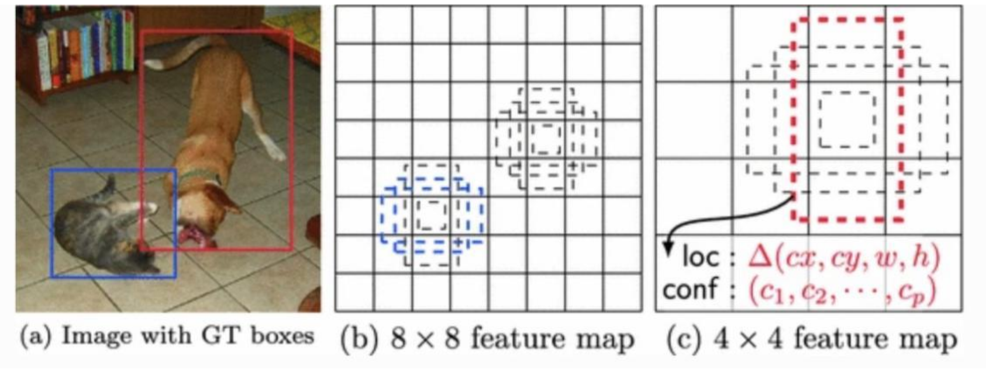


Figure 5.3 SSD detection algorithm (Liu, Anguelov, Erhan, Szegedy, 2016)

Mask R-CNN: Mask R-CNN is an extension of Faster R-CNN that adds a third branch to the network for predicting object masks in addition to bounding boxes and class probabilities (Bharati, P. and Pramanik, A., 2020). Mask R-CNN (Figure 5.4) is known for its accuracy and is commonly used in applications such as image segmentation and augmented reality.

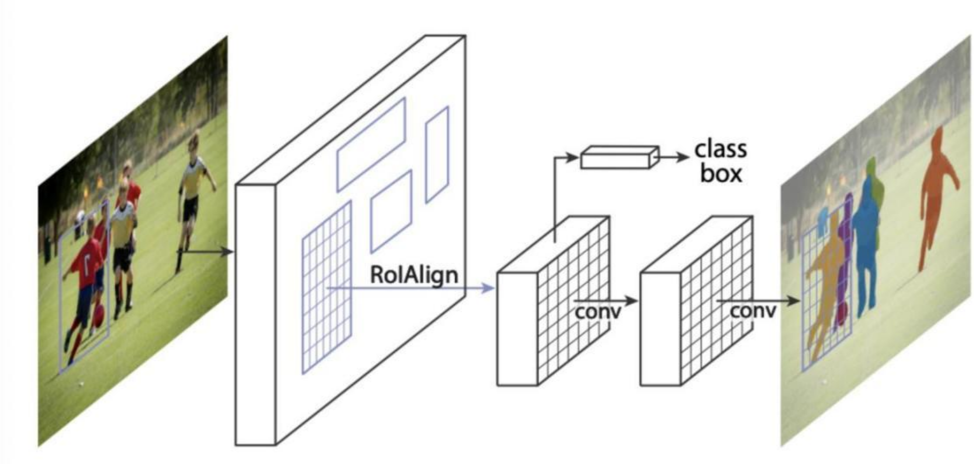


Figure 5.4 Mask R-CNN (Bharati, and Pramanik, 2020)

RetinaNet: RetinaNet (Figure 5.5) is a single-stage object detection system that uses a focal loss function to address the class imbalance problem that is common in object detection (Chen, and Qin, 2022). RetinaNet is known for its simplicity and effectiveness, and it is commonly used in applications such as pedestrian detection and satellite imagery analysis.

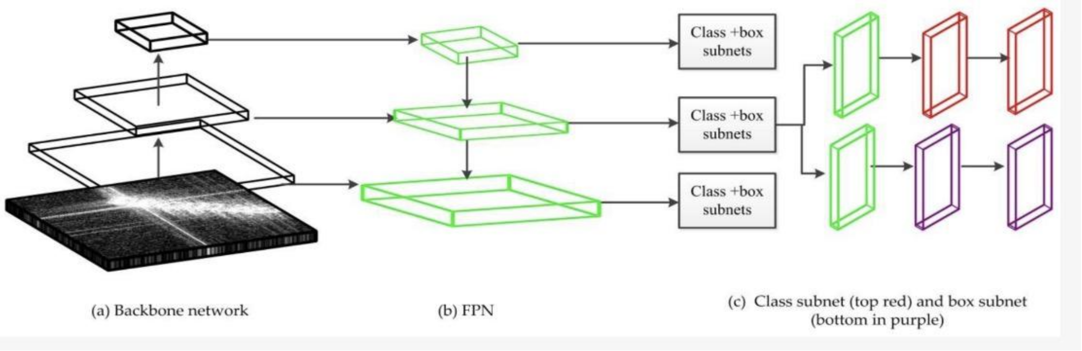


Figure 5.5 RetinaNet (Wang, Yuan Wang, Zhang ,2019)

In this subsection, we summarise most of the types of detection algorithms currently widely used. The following sections show deep research on the YOLO object detection algorithm analysis.

5.2.2 Advantages of the YOLO Object Detection Algorithm

Object detection is a popular field in computer vision that involves detecting objects within an image or video frame and classifying them into various categories (Paneru,, Jeelani, 2021). Among the various object detection algorithms available, YOLO (You Only Look Once) stands out due to its speed, accuracy, and efficiency.

Real-time Processing Speed

One of the most significant benefits of using YOLO is its real-time processing speed. Traditional object detection algorithms such as R-CNN and Fast R-CNN are slower as they detect objects in multiple stages (Xiao, Wang, Zhang, Meng, 2020). In contrast, YOLO detects objects in a single pass. This means that YOLO can detect objects in real-time, making it suitable for real-time applications such as autonomous driving, surveillance, and video analysis.

1. High Accuracy

YOLO has high accuracy in detecting objects compared to other object detection algorithms. It uses a single neural network to predict bounding boxes and class probabilities simultaneously, which makes it more accurate than other algorithms (Thuan, 2021). YOLO can detect objects with high precision and recall, making it an excellent choice for applications where accuracy is critical.

2. Fewer False Positives

YOLO has a lower false positive rate than other object detection algorithms. This means that it is less likely to detect objects that are not present in the image, which reduces the chances of generating false alarms (Redmon, Divvala, Girshick and Farhadi, 2016). YOLO also has a high detection rate, which means that it can detect objects that other algorithms might miss.

3. Handles Occlusion

Another benefit of YOLO is that it can handle occlusion, which is when an object is partially or fully obstructed by another object (Kim, and Cho, 2021). YOLO can detect objects even if they are partially occluded, making it suitable for applications where objects are likely to be partially hidden.

4. Handles Small Objects

YOLO is capable of detecting small objects accurately, which is challenging for other object detection algorithms. YOLO can detect objects as small as 10 pixels

(Kharchenko and Chyrka, 2018), making it suitable for applications where small objects need to be detected.

5. Supports Multiple Object Classes

YOLO can detect multiple object classes simultaneously (Ivašić-Kos, Krišto, M. and Pobar, 2019). This means that it can detect objects belonging to different classes, such as people, cars, and animals, in the same image (Pham, Courtrai, Friguet, Lefèvre, Baussard, 2020). YOLO can detect up to 80 different object classes, making it suitable for a wide range of applications.

6. Easy to Implement

YOLO is easy to implement, making it suitable for developers who want to add object detection capabilities to their applications (Kosuge, Suehiro, Hamada, Kuroda, 2022). YOLO is open-source and available on GitHub, making it easy for developers to access and use.

7. GPU Optimization

YOLO is optimized for GPU processing, making it faster and more efficient than other object detection algorithms (Huang, Pedoeem, and Chen, 2018). This means that YOLO can process large amounts of data quickly, making it suitable for real-time applications where data processing speed is critical.

8. Transfer Learning

Transfer learning is the process of reusing pre-trained models to solve new problems (Huang, Pedoeem, and Chen, 2018). YOLO can be used for transfer learning, allowing developers to use pre-trained models to solve new object detection problems. This makes it easier for developers to develop new object detection applications without starting from scratch.

Constant Improvement

YOLO is continually being improved by researchers and developers, making it more accurate and efficient over time. YOLO has now been updated with 7 versions from YOLO-V1 to YOLO-V7, of which YOLO-V3 and YOLO-V5 are among the more widely used versions. This means that YOLO is a future-proof object detection algorithm that will continue to improve as new research and techniques are developed. YOLO is an excellent object detection algorithm that offers many benefits, including real-time processing speed, high accuracy, fewer false positives, handling occlusion, handling small objects, supporting multiple object classes, easy implementation,

5.3 The Principle and Implementation of the YOLO Algorithm

This section mainly introduces the operating principle of YOLO. Target detection is a relatively simple task in computer vision used to find particular objects in a picture. Target detection requires us not only to identify the categories of these objects but also to label the locations of these objects. The categories are discrete data, and the locations are continuous data.

The full name of YOLO is you only look once, meaning you only need to look once to identify the class and location of an object in a picture. Because you only need to look once, YOLO is called a Region-free method. In contrast to Region-based methods, YOLO does not need to find the Regions where targets may be present in advance.

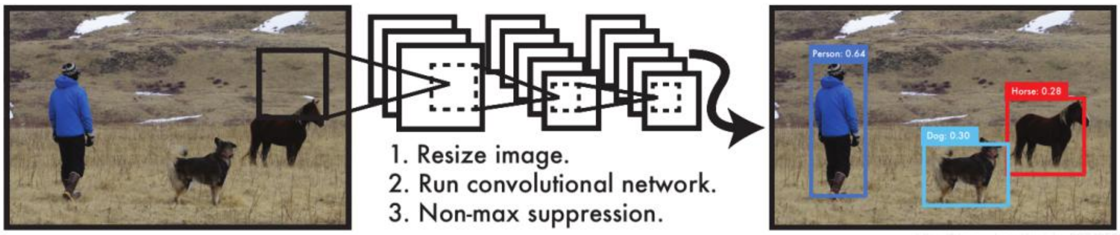


Figure 5.6 YOLO detection vision

In the image above Figure 5.6, there are three tasks in computer vision: classification, target detection, and instance segmentation. Overall, these three types of tasks range from easy to complex, with the target detection we will discuss being in the middle. The first classification task is the basis for our target detection.

5.3.1 Image Input

The first step is preprocessing, which is an essential step in YOLO object detection, and one of the tasks involved in preprocessing is resizing and normalizing the input image (Benedict, 2022.). The reason for resizing the image is to ensure it is compatible with the input size expected by the YOLO model. Different versions of YOLO may have different input size requirements, but generally, the input images are resized to a fixed size, such as 416x416 pixels or 608x608 pixels. Resizing the image also helps to reduce the computational overhead and speed up the object detection process.

Normalization is another important preprocessing step, and it involves scaling the image's pixel values to a fixed range, which helps improve the neural network's performance by ensuring that the input data is in a consistent range of values. Normalization is typically performed by subtracting the mean pixel value of the dataset from each pixel value of the input image and then dividing the result by the standard deviation of the pixel values. Resizing and normalization are necessary preprocessing steps in YOLO object detection (Jeong, Park, and Ha, 2018). They help ensure that the input image is compatible with the YOLO model and is in a consistent range of values. Figure 5.7

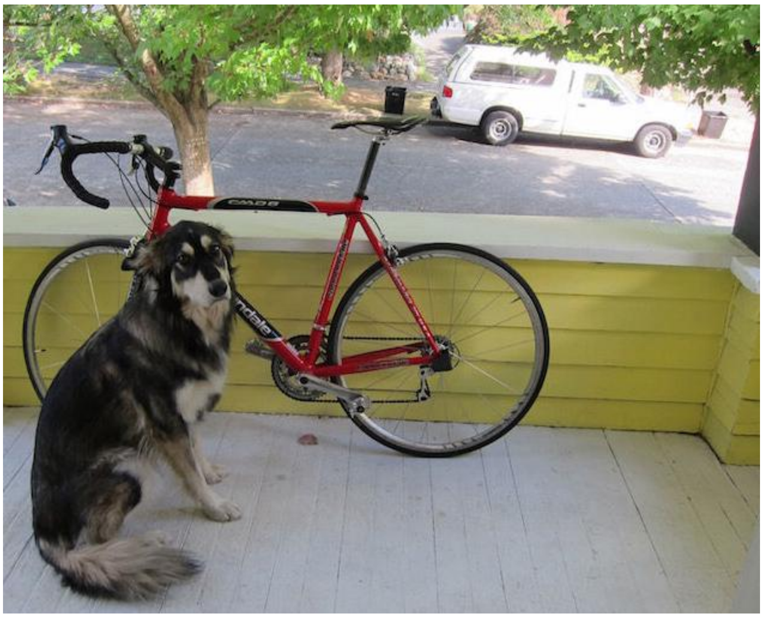


Figure 5.7 Image input

5.3.2 Grid-Based Approach

YOLO uses a grid-based approach for object detection.

Grid Cells: After the input image is processed by CNN Backbone (Maity, Banerjee and Chaudhuri, 2021) is divided into a grid of cells. The grid size depends on the specific YOLO variant, typically 13x13 or 19x19 (Ghimire and Horanont, 2017). Figure 5.9. Each cell is responsible for detecting objects that fall within it.

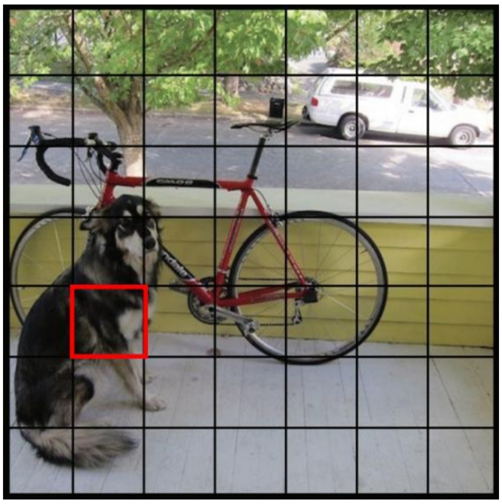


Figure 5.8 S x S gird on input

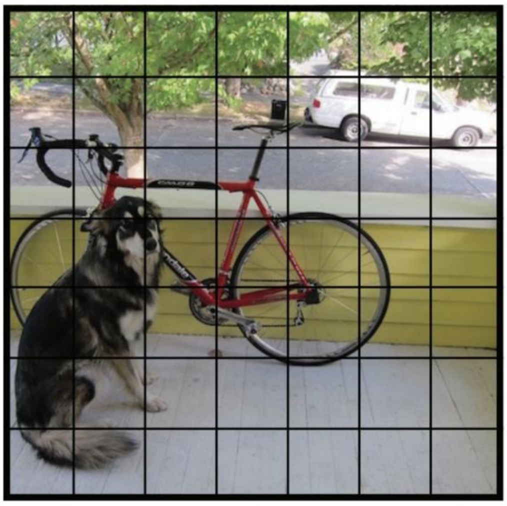


Figure 5.9 Target center point

5.3.3 Bounding Box Prediction

For Bounding Box Prediction, when the input image is divided into S*S cells, each cell detects the target whose centre point falls within the cell. For example, in Figure 5.10, the red cell is the centre point of the target dog, so this cell is responsible for the prediction of the dog.

Each grid predicts B bounding boxes and the confidence score of the bounding box, which has two aspects: the probability that the bounding box contains the target and the accuracy of the bounding box. Each cell is predicted to have B bounding boxes and a confidence score. The so-called confidence level consists of two aspects: the probability that the bounding box contains the target and the accuracy of the bounding box. When there is no target in the bounding box but the background, Pr(object)=0. Conversely, When there is no target in the bounding box but only background, Pr(object) = 0. Conversely, when the bounding box contains a target, Pr(object) = 1. Then markers as:

$$\mathrm{IOU}_{pred}^{truth} \qquad \text{（equation 1）}$$

Thus the confidence level can be defined as:

$$P_{\gamma}\left(object\right) * \mathrm{IOU}_{pred}^{truth} \qquad \text{（equation 2）}$$

The confidence in YOLO is not the probability that the bounding box contains a target but the product of the other two factors. The accuracy of predicting the bounding box is also reflected in it. The size of the bounding box is usually represented by four values: (x, y, w, h), where (x. y) is the center coordinate of the bounding box, and w and h represent the width and height of the bounding box, respectively. It should be noted that the predicted value of the center coordinates (x, y) is the offset value relative to the upper left corner of each cell, aligned with the grid cell (i.e., the offset value relative to the current grid cell). In contrast, the predicted values of w and h of the bounding box are relative to the ratio of the width to the height of the whole picture, so the size of the four elements should theoretically be in the range of [0,1] (Jiang, Ergu, Liu, Cai, and Ma, 2022). The final prediction of each bounding box contains five elements: (x, y, w, h, c), the first 4 of which characterize the size and position of the bounding box, while the last one is the confidence level.

For the classification problem, for each cell it also gives the predicted probability value of C categories, which characterizes the probability that the target belongs to each category of the bounding box for which the cell is responsible. But these probability values are actually conditional probabilities at the confidence level of each bounding box, as:

$$P_{\gamma}\left(class_i \middle| object\right) \qquad \text{(equation 3)}$$

It is worth noting that no matter how many bounding boxes a cell predicts, it only predicts a set of class probability values, which is a drawback of the YOLO algorithm. In later improved versions, the YOLO bound the class probability prediction values with the bounding boxes. At the same time, we can calculate the class-specific confidence scores for each bounding box:

$$P_{\gamma}\left(class_i \middle| object\right) * P_{\gamma}\left(object\right) * \mathrm{IOU}_{pred}^{truth} = P_{\gamma}\left(class_i\right) * \mathrm{IOU}_{pred}^{truth} \qquad \text{(equation 4)}$$

In particular, the confidence level of the bounding box categories characterises the likelihood that the targets in the bounding box belong to each category and how well the bounding box matches the target.

Each cell requires a prediction of (Bx5+C) values. If the input image is divided into S × S grids, the final predicted value is a tensor of size S × S × (Bx5+C). For the PASCAL VOC data, which has 20 categories, if S=7, B=2, then the final prediction is a tensor of size 7×7×30.

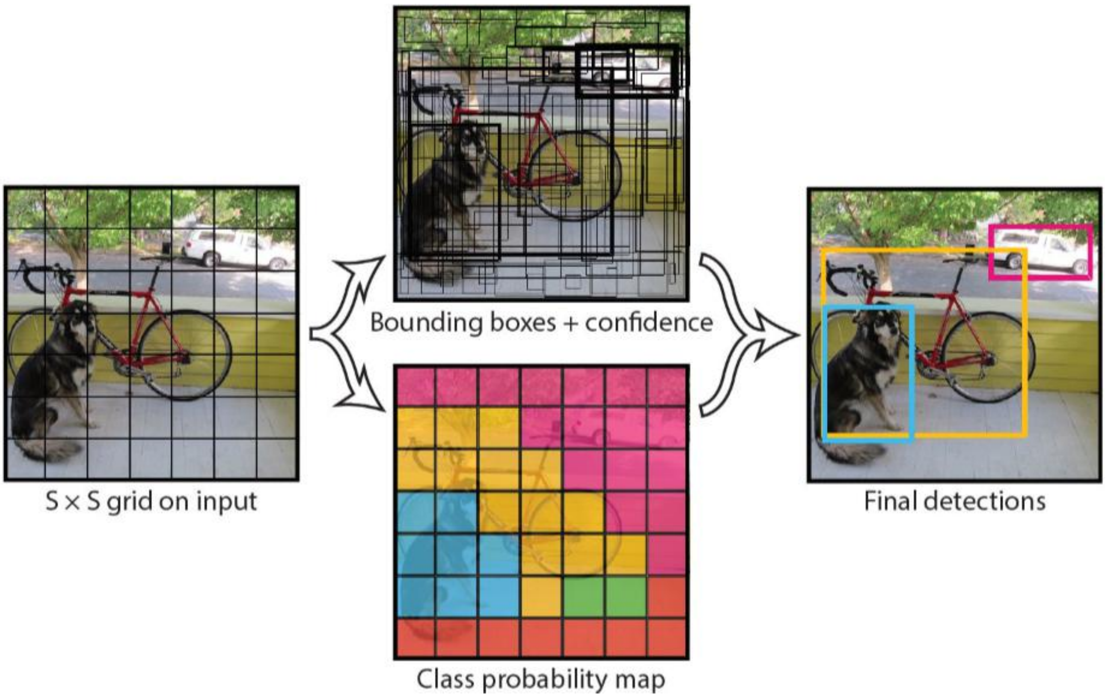


Figure 5.10 Predictive value structure of the model (Redmon, Divvala, Girshick and Farhadi, 2016)

We can understand as the aim of Yolo is to find an object in a picture and give its class and position. Target detection is based on supervised learning, and the supervised information for each image is the N objects it contains, with five pieces of information for each object, namely the object's centre position (x, y) the height (h), width (w) and confidence score(C).

5.3.4 Network Design

Yolo uses a convolutional network to extract the features and then uses fully connected layers to obtain the predicted values. The network structure is based on the GooLeNet model, with 24 convolutional layers and 2 fully connected layers, Figure 5.11. For the convolutional layers, 1x1 convolution is mainly used to do channel reduction, followed by 3x3 convolution. For the convolution and fully connected layers, the Leaky ReLU activation function is used: max(x, 0.1 x). The last layer, however, uses a linear activation function.

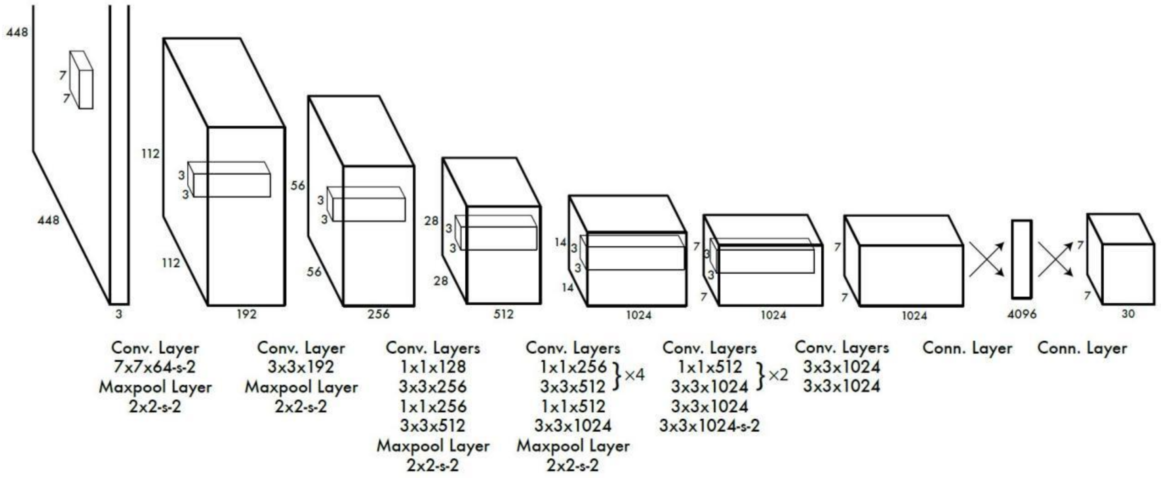


Figure 5.11 Network overview (O.D.S.C.- O.D.2018)

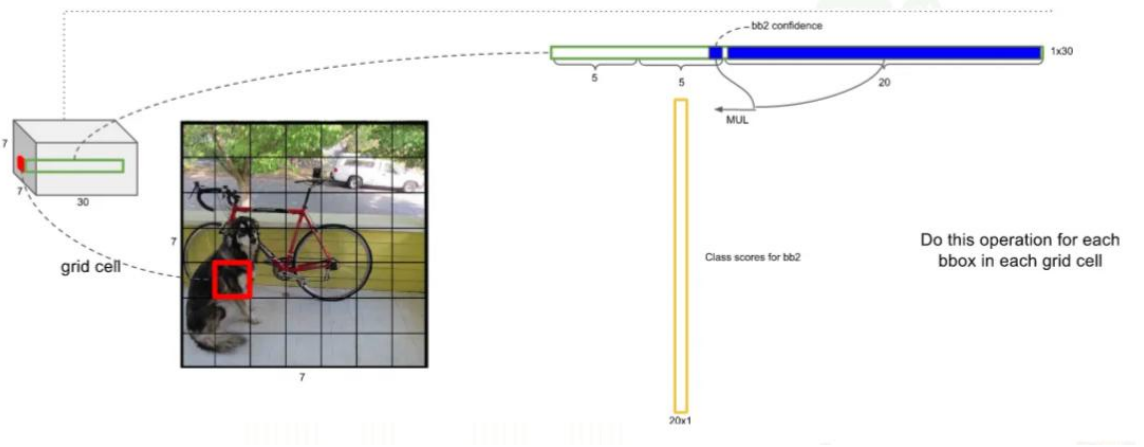


Figure 5.12 Network progress (Odsc,2018)

From figure 5.12, we can obtain the network's final output as a tensor of size 7 x 7 x 30. This is consistent with the previous discussion. For each cell, the first 20 elements are the category probability values, then two elements are the bounding box confidence levels, which are multiplied to obtain the category confidence levels, and the last eight elements are the bounding box (x, y, w, h). The bounding box also separates the confidence c and (x, y, w, h) so that each component can be easily extracted. The predicted value of the network is a two-dimensional tensor P with a shape of [batch,7x7×30]. Using slicing, then P[:,0:7x7x20] is the category probability part, while P[:,7x7x20:7x7x(20+2)] is the confidence part and the final remaining part P[:,7x7x(20+2):] is the prediction of the bounding box. In this way, it is very convenient to extract each part, which will facilitate the computation during training and prediction later on.

5.3.5 Network training

Before formal training, the classification model was pre-trained on ImageNet using the first 20 convolutional layers in Figure5.13, followed by adding an average-pool layer and fully-connected layers. After pre-training, four convolutional layers and two fully-connected layers were randomly initialised on top of the 20 convolutional layers obtained from pre-training. As the detection task generally requires higher-resolution images, the input to the network was increased from 224x224 to 448x448. The flow of the entire network is shown in the Figure 5.13

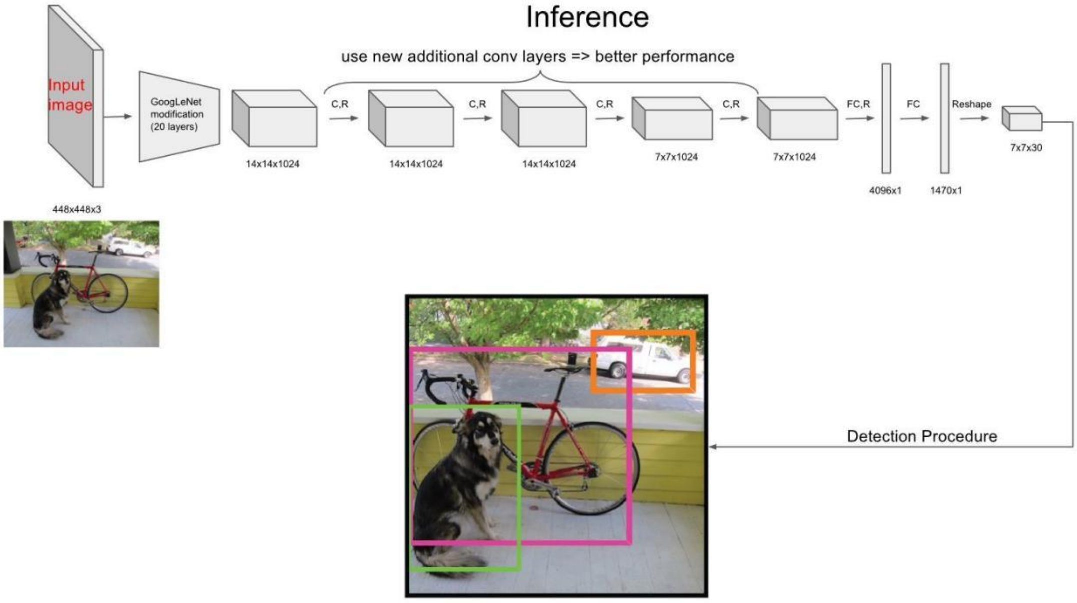


Figure 5.13 Flow of the network

5.3.6 Loss Function

The Yolo algorithm treats target detection as a regression problem. A mean squared difference loss function used in YOLO (Wang, Yang, Zhang, 2020).

Different complex weights are also used for the different components. First, a distinction is made between the localisation and classification errors. The localisation error (the prediction error in the bounding box coordinates) is given a more significant weight: λcoord = 5。The smaller weight λnoobj = 0.5 was used to differentiate the confidence level of the bounding boxes without targets from those with targets. All other weights were set to 1. The mean square error was then used to treat the different sizes of the bounding boxes equally. Since the coordinate errors of smaller bounding boxes are usually more sensitive than those of larger bounding boxes, the width and height predictions of the bounding boxes are changed to square root predictions, and the final predictions can be expressed as:

$$\left(x, y, \sqrt{\omega}, \sqrt{h}\right) \qquad \text{(equation 5)}$$

Each cell predicts multiple bounding boxes. However, its corresponding category is only one. So then, if a target exists in that cell during training, only the bounding box with the large Intersection over Union (IOU) to the ground truth is selected to predict that target. In contrast, the other bounding boxes are considered not to have a target. The result will be a more specialised bounding box for each cell, which can be applied to targets of different sizes and aspect ratios, thus improving model performance. When there are multiple targets in a cell, the Yolo algorithm can only select one for training, which is one of the disadvantages of the Yolo algorithm. Also, for bounding boxes with no corresponding target, the error term is only the confidence level, and the coordinate term error cannot be calculated. The classification error term is only calculated if a cell has an actual target, otherwise, it is not calculated either. Figure 5.14 shows the loss function.

Regression loss

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

$$+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

Confidence loss

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$

Classification loss

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

Figure 5.14 Loss function (Ahmad, Ma, Yahya, Ahmad, Nazir, and Haq, 2020)

In this loss function, the first term is the error term of the centre coordinates of the bounding box, and the presence of the target in the I-th cell is denoted as: 1_{ij}^{obi} , and the J-th bounding box in that cell is responsible for predicting that target.

1_{i}^{obi} The second term is the error term for the height and width of the bounding box.

The third term is the confidence error term for the bounding box containing the target. The fourth term is the confidence error term for a bounding box that does not contain a target. Furthermore, the last term is the classification error term for the cells containing the target. In classification loss the

Which refers to the presence of a target in the I-th cell. In particular, if the target is not present, Ci=0 since Pr(object)=0.

If a target exists, Pr(object)=1; at this point, it is necessary to determine the

$\mathrm{IOU}_{pred}^{truth}$ We can take the IOU to be 1 so that Ci=1. In the YOLO implementation, a control parameter rescore (default 1) is used; when it is 1, the IOU is not set to 1, but the true IOU between truth and pred is calculated.

YOLO's forecasting uses the non maximum suppression (NMS) algorithm. The following is a detailed description of the NMS algorithm and the application of the NMS algorithm in YOLO.

Non-maximum suppression (NMS) is a technique used in computer vision and image processing to reduce the number of overlapping detections in an image (Hosang, Benenson and Schiele, 2017), which often works in object detection tasks to filter out redundant bounding boxes around objects.

The basic idea of NMS is to suppress all but the best bounding box (with the highest confidence score) that covers a particular object in the image. The algorithm works as follows:

1. Sort the bounding boxes by their confidence scores in descending order.
2. Pick the bounding box with the highest confidence score and output it as a detection.
3. Remove all the bounding boxes that have a high overlap (measured by intersection over union, or IOU) with the selected bounding box.
4. Repeat steps 2 and 3 until there are no more bounding boxes left.

This is a step-by-step example of how NMS works:

Assume we have 5 bounding boxes with confidence scores and their corresponding IOU scores with other boxes:

Box 1: confidence score = 0.8, IOU with box 2 = 0.4, IOU with box 3 = 0.5, IOU with box 4 = 0.1, IOU with box 5 = 0.2

Box 2: confidence score = 0.7, IOU with box 1 = 0.4, IOU with box 3 = 0.6, IOU with box 4 = 0.2, IOU with box 5 = 0.3

Box 3: confidence score = 0.6, IOU with box 1 = 0.5, IOU with box 2 = 0.6, IOU with box 4 = 0.3, IOU with box 5 = 0.4

Box 4: confidence score = 0.5, IOU with box 1 = 0.1, IOU with box 2 = 0.2, IOU with box 3 = 0.3, IOU with box 5 = 0.5

Box 5: confidence score = 0.9, IOU with box 1 = 0.2, IOU with box 2 = 0.3, IOU with box 3 = 0.4, IOU with box 4 = 0.5

Sort the bounding boxes by their confidence scores in descending order:

Box 5: confidence score = 0.9

Box 1: confidence score = 0.8

Box 2: confidence score = 0.7

Box 3: confidence score = 0.6

Box 4: confidence score = 0.5

Pick the bounding box with the highest confidence score (Box 5) and output it as a detection. Remove all the bounding boxes that have a high overlap with Box 5 (IOU > 0.5):

Box 5: confidence score = 0.9 (output as a detection)

Box 1: confidence score = 0.8, IOU with box 5 = 0.2 (remove)

Box 2: confidence score = 0.7, IOU with box 5 = 0.3 (remove)

Box 3: confidence score = 0.6, IOU with box 5 = 0.4 (remove)

Box 4: confidence score = 0.5, IOU with box 5 = 0.5 (remove)

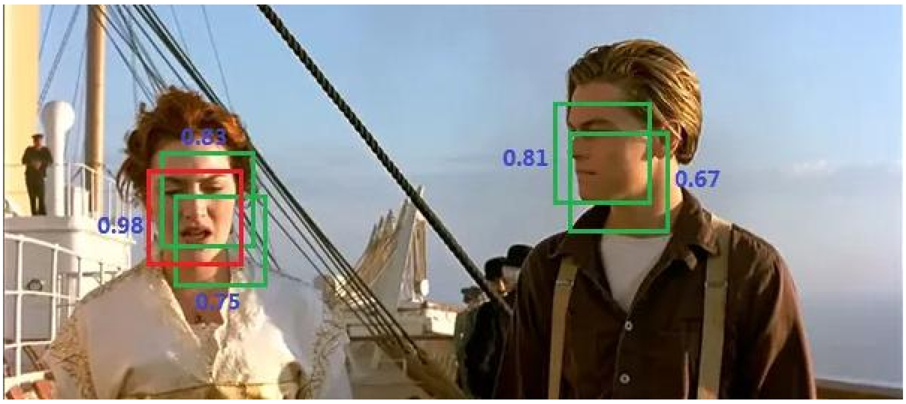


Figure 5.15 Non maximum suppression (NMS) in YOLO

The following is a case study of image processing by Non-maximum suppression (NMS) in YOLO. Not only does YOLO sport the NMS algorithm, but most of the detection algorithms use NMS. As shown in Figure 5.15, the face is detected a number of times, but we want to output only one of the best-predicted frames. We only need the detection result of the red frame for the lady in the picture. Then the NMS algorithm can achieve this effect: first, find the frame with the highest confidence level from all the detected boxes, then calculate its IOU with the remaining box one by one. If the value is greater than a certain threshold (overlap is too high), then the box is eliminated; then the above process is repeated for the remaining boxes until all the boxes are processed.

In figure 5.16 (same as Figure 5.11), the model input image has a shape of [448,448,3], and after several convolutional and pooling layers, the output feature map has a shape of [7,7,1024]; this feature map is then flattened into a fully-connected layer with 4096 neurons, and a 4096-dimensional vector is an output; this vector is then fed into a fully-connected layer with 1470 neurons, and a 1470-dimensional vector is an output; finally, this vector is reshaped into a [7,7,30] feature map. The vector is then fed into a fully connected layer with 1470 neurons, resulting in a 1470-dimensional vector; finally, the vector is reshaped into a [7,7,30] feature map.

In the prediction phase, the YOLOV1 model is equivalent to a black box, with an input image of [448,448,3] and an output feature map of [7,7,30]. The output tensor contains all prediction boxes' coordinates, confidence levels, and category results.

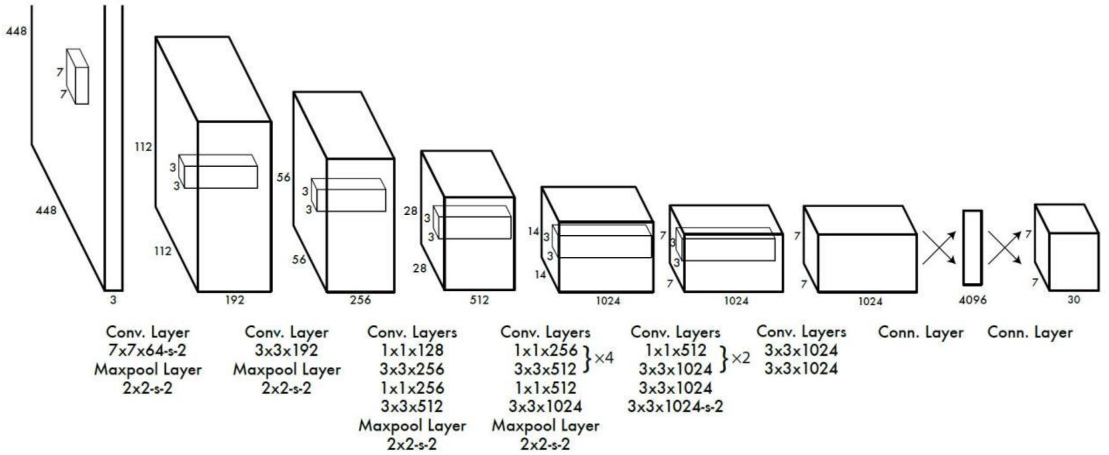


Figure 5.16 Network overview (O.D.S.C.- O.D.2018)

In the first two subsections, we learned that the final network output is 7 × 7 × 30 .We can understand as the network divides the image into SxS grid cells, S=7 in Figure5.16, so each image is divided into a 7x7 grid. Each grid cell predicts b bounding boxes, b=2 in YOLOv1, and each grid predicts two bounding boxes. The two prediction boxes may vary significantly in size and shape, but as long as the centroid of the box falls within the grid, the box is generated by that grid. So the centroids of the two prediction boxes generated by each grid must fall in that grid.

Each prediction box contains the centroid coordinates (x, y), width and height (w, h) which are the four positioning coordinates that determine the position of the prediction box; the confidence c that the object in the prediction box is the target object; and the conditional probability of containing all categories, assuming that the object is of a particular category if the prediction box already contains the target object. For example, the probability of being a dog given that the target object is included.

Multiplying the confidence level of each prediction frame by the conditional probability of the category gives the probability that each prediction frame belongs to each category.

The number of channels in the output feature map is 30, which can be interpreted to mean that each grid generates two prediction frames, each containing five parameters (x, y, w, h, c), so two prediction frames would have ten parameters and contain 20 categories in the VOC dataset, i.e. each grid contains the conditional probabilities of these 20 categories.

Thus, each grid contains 5+5+20 parameters, and each image is divided into a 7x7 grid, with one image having 7x7x30 parameters. Figure 5.17

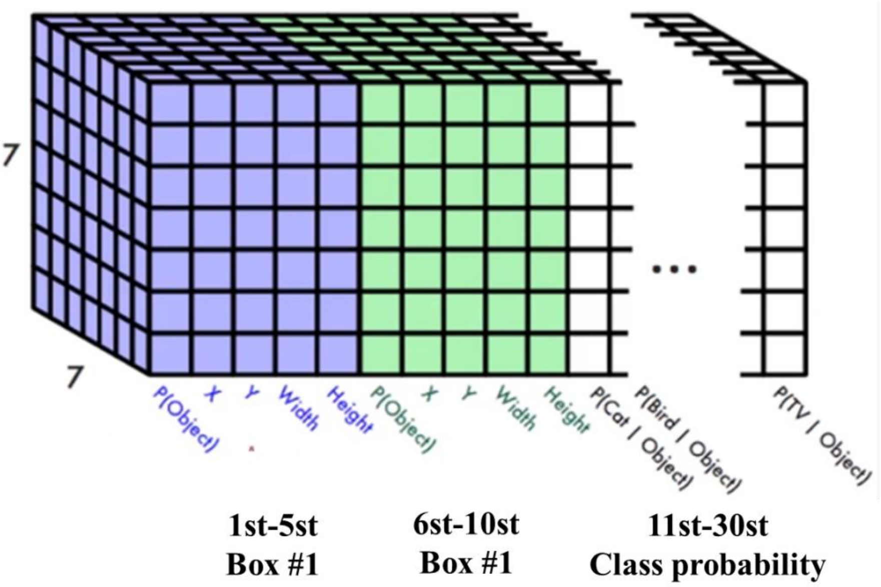


Figure 5.17 30 Parameters

Each grid predicts two prediction boxes: the prediction box with high confidence, represented by a thick line and the one with low confidence, represented by a thin line, keeping the box with high confidence. Figure 5.18. Each grid also generates conditional probabilities for 20 categories, as shown on the right, showing the grids occupied by the categories with high conditional probabilities. For example, the green colour represents the area with a high conditional probability for dogs. Each grid has only one category, and the one with the highest of the 20 conditional probabilities is selected.

Only one target object can be detected per grid, whereas a 7x7 grid can predict up to 49 objects.Which is one of the reasons why the YOLOv1 version does not perform well in small, mini-target detection.

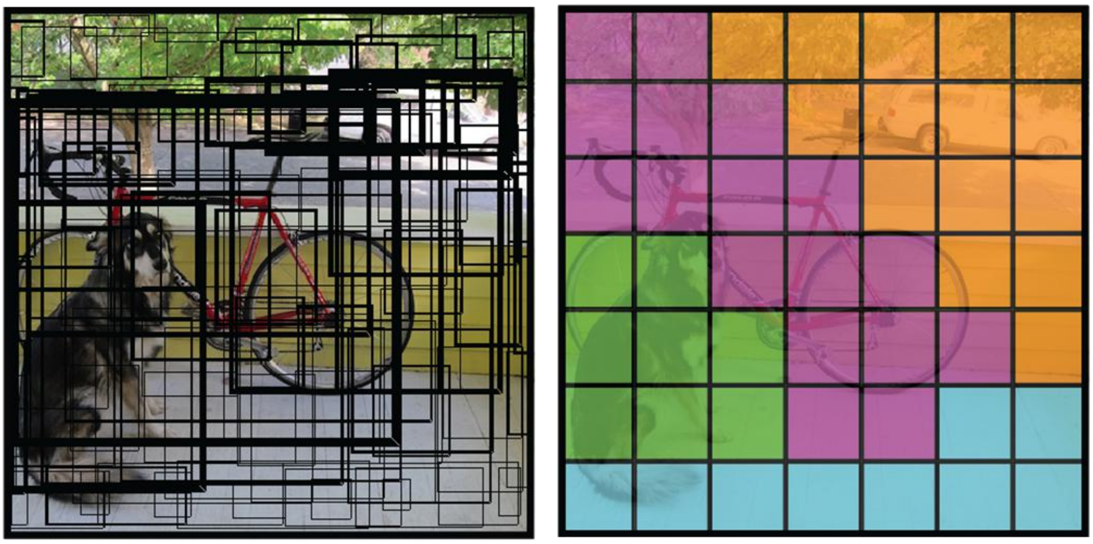


Figure 5.18 Prediction boxes with confidence

At last, the 7x7x30 feature maps output by the network are turned into the final target detection results. One grid is taken from the 7x7 grid and studied. In contrast, each grid contains two predictor parameters (x, y, w, h, c) and 20 conditional probabilities of the category (assuming the predictor contains the target object, the probability that it is a particular category), i.e. 5+5+20 parameters, figure 5.19.

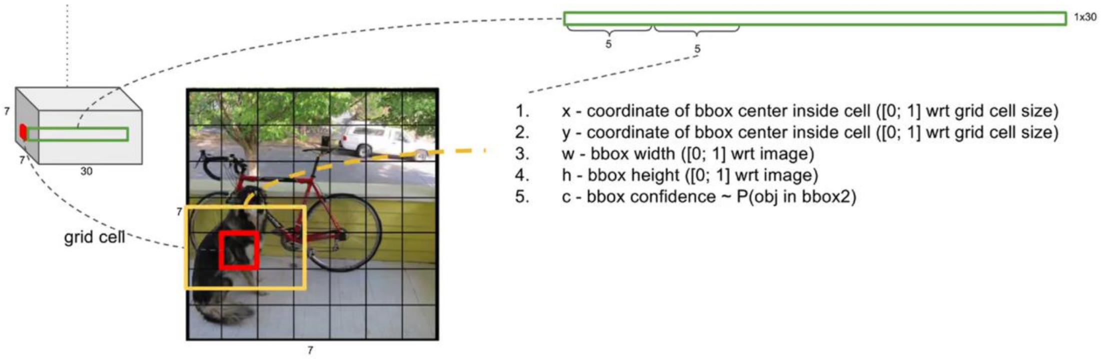

1x30
5
5
7
7
30
grid cell
7
7
1. x - coordinate of bbox center inside cell ([0; 1] wrt grid cell size)
2. y - coordinate of bbox center inside cell ([0; 1] wrt grid cell size)
3. w - bbox width ([0; 1] wrt image)
4. h - bbox height ([0; 1] wrt image)
5. c - bbox confidence ~ P(obj in bbox2)

Figure 5.19 5+5+20Parameters

Next, the confidence of each prediction box is multiplied by the conditional probability that each grid belongs to one of the 20 categories to obtain the probability that the grid belongs to a category. The confidence of the first predictor box is multiplied by the conditional probability of the 20 categories to obtain the total probability of the first predictor box belonging to the 20 categories. One 20-dimensional vector represents the probability that a predictor box belongs to each category. Each grid then has two probability vectors, each with 20 elements. 7x7 grids have 98 vectors (7x7x2=98) Figure 5.12

However, using the NMS algorithm in YOLO adopts a different mechanism from other algorithms. First, for each prediction box, the category with the highest confidence level is selected as its prediction label. After this layer of processing, we obtain the prediction category and the corresponding confidence value for each box, all sizes [7,7,2]. In general, a confidence threshold is set, meaning that boxes with confidence less than this threshold are filtered out, so after this layer of processing, the predictor boxes with higher confidence remain. Finally, the NMS algorithm is applied to these boxes; the final result is the detection result. A point worth noting is that NMS is applied to all predictor boxes equally and distinguishes between each category, using NMS separately.

In YOLO (Figure 5.20), NMS was first used before determining the category of each box. Then, for each of the 98 boxes, values less than the confidence threshold is first returned to 0, and then NMS is applied to the confidence values on a case-by-case basis, where the result of the NMS process is not a rejection but a return of the confidence value to 0. Finally, the category of each box is determined, and the detection result is output when its confidence value is not 0.

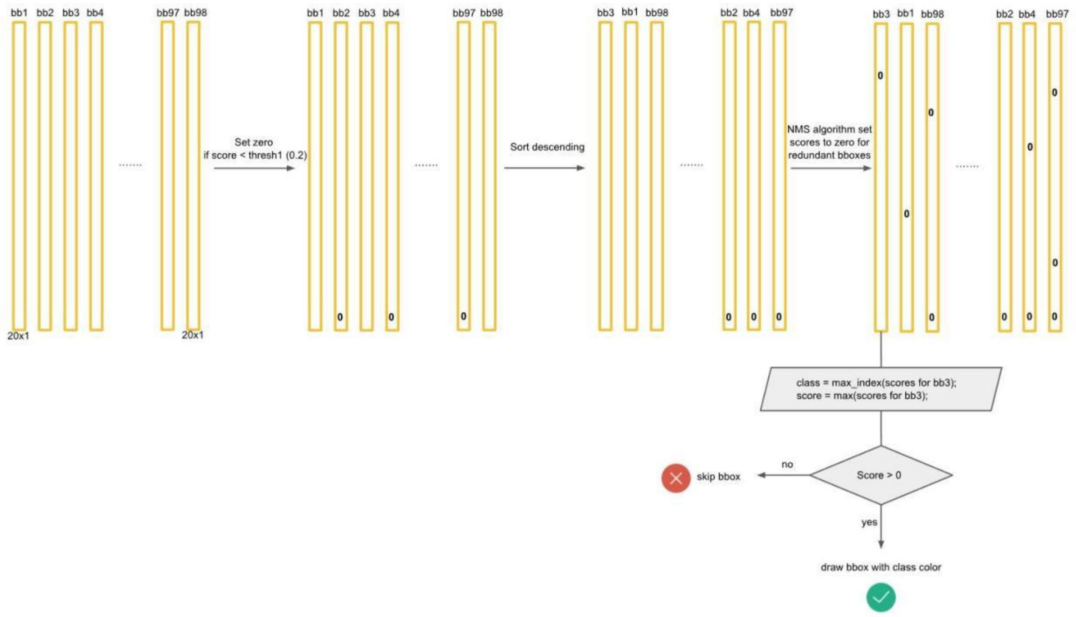


Figure 5.20 Non maximum suppression in YOLO

5.4 Summary

YOLO (You Only Look Once) is a popular computer-vision object detection algorithm. The main idea behind YOLO is to divide the image into a grid and predict the object's class and location in each grid cell (Nie, Sommella, O'Nils, Liguori and Lundgren, 2019). The algorithm uses a single convolutional neural network (CNN) to perform this task, making it a single-stage detector.

The critical steps involved in the YOLO algorithm:

1. First, divide the input image into a grid of cells.

2. For each cell, predict a bounding box and associated confidence score. The

bounding box specifies the object's location in the cell, and the confidence score represents the algorithm's confidence in the detection.

3. Assign a class score to each bounding box, representing the probability of the object belonging to a particular class.

4. Use non-max suppression to remove overlapping bounding boxes and keep only the most confident detection.

In this chapter, we analyse the YOLO algorithm. Three of the modules, bounding box prediction, loss function and non-maximum suppression (NMS), we investigate and interpret its operation in detail. In the following research, we will construct YOLO to experiment with blind zone detection of cars in a virtual environment.

Chapter 6 The Construction and Application of YOLO

6.1 Introduction

In this chapter, we focus on building the YOLO object detection algorithm using the images and videos we recorded in the virtual environment by constructing virtual accidents without damage to property and human well-being, for object detection recognition by the YOLO algorithm. We discuss and summarise image detection and video detection in two parts, namely YOLOv3 image detection and YOLOv5 video detection. In subsection 6.2, we describe the differences between YOLOv3 and the original YOLO, then use YOLOv3 for image detection, summarise the errors that tend to occur in the YOLOv3 environment built for successful image detection, and suggest modifications. Sub-section 6.3 focuses on the YOLOv5 recognition algorithm. After comparing YOLOv5 with YOLOv3 at the beginning and summarises the points of YOLOv5. Finally, the video we recorded in the virtual scene was successfully recognised, and the feasibility of YOLO for virtual object detection was verified. A summary of this chapter is presented in subsection 6.4.

The YOLO family of object detection algorithms is updated rapidly, with each update bringing different algorithm performance improvements. Therefore, as the project progressed, we used different versions of the YOLO detection algorithm..

6.2 YOLO v3 With Image Detection

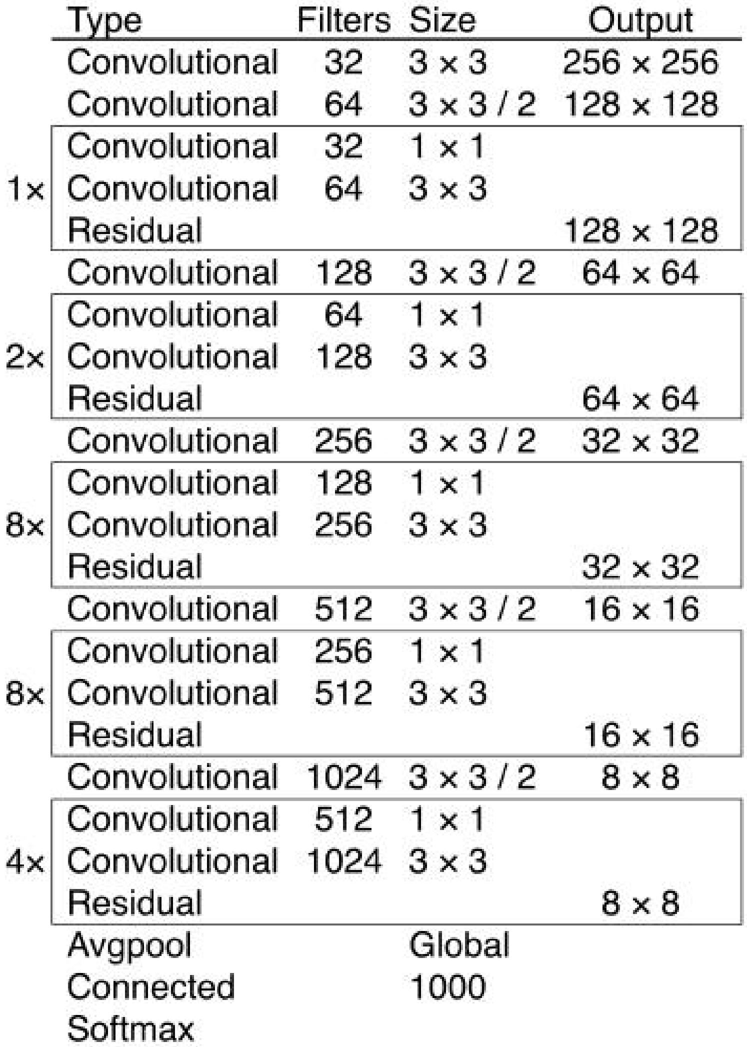
YOLO v3, the third iteration of the YOLO architecture, was released in 2018. It uses a deep neural network with over 53 convolutional layers, including residual blocks and skip connections, to detect objects in images and videos (Mujahid, Awan, Yasin, Mohammed, Damaševičius, 2021). The architecture is designed to optimize for both accuracy and speed, achieving real-time detection speeds of up to 30 frames per second.In this section we arr focus on YOLO v3 architecture application in image part.

6.2.1 YOLO Version 3 Updates

YOLOv3 has been upgraded from the original YOLO version and is faster, more accurate and more stable. From YOLOv1 to YOLOv3, each generation of performance improvements has been closely linked to improvements in the backbone (the backbone network).

1. YOLOv3 uses a new backbone architecture called Darknet-53, which is deeper and

more powerful than the previous Darknet-19 backbone used in YOLO. This allows YOLOv3 to extract more high-level features from the input image and improve detection performance. Fugure 6.1 .Darknet-53 consists mainly of 1×1 and 3×3 convolutional layers, each followed by a batch normalization layer and a Leaky ReLU, which are included to prevent overfitting. The convolutional layers, the batch normalization layer and the Leaky ReLU together form the basic convolutional unit DBL in Darknet-53, so called because there are 53 such DBLs in Darknet-53.



	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Fugure 6.1 Darknet-53 (won, Lee,and Lin, 2019)

2. Multi-scale fusion, the feature map can be 13x13, 26x26 or 52x52, which is beneficial for detecting small object category probabilities. Figure 6.2

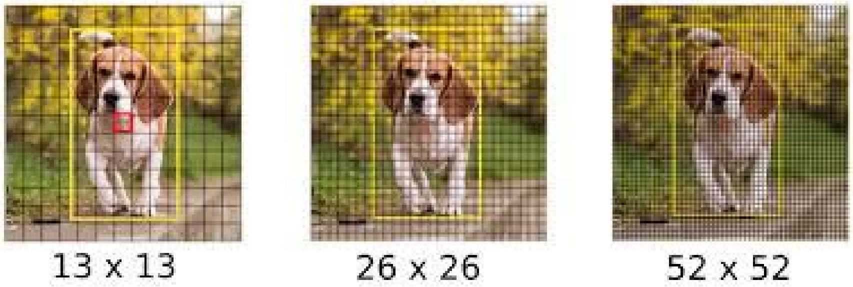


Figure 6.2 Multi-scale fusion

3. Multi-label classification, replacing the first two versions of the softmax function with multiple independent (logistic) classifiers to calculate the probability of inputting a particular label.

4. One of the core ideas of YOLOv3 is to fuse different feature maps for prediction, e.g. up-sampling before fusion. YOLOv3 also uses the idea of Resnet (Targ, Almeida and Lyman, 2016), stacking more layers for feature extraction. Figure 6.3 shows the network structure of YOLOv3.

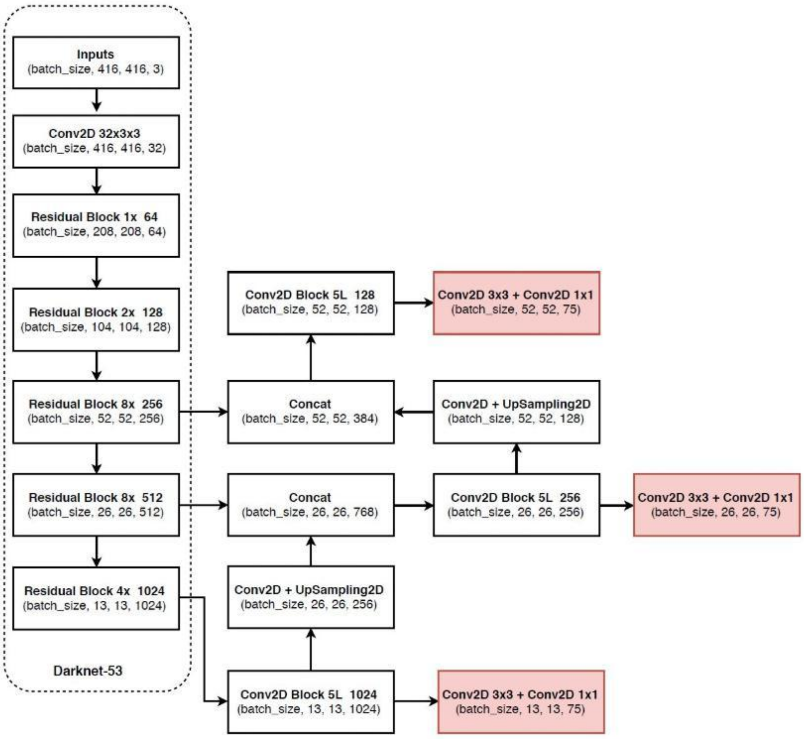


Figure 6.3 Network structure of YOLOv3 (Mao, Sun, Liu, and Jia, 2019)

6.2.2 YOLOv3 Installation and Summary

Installation

We use the Mac Os system for the initial tests in the YOLO image recognition test phase. At the same time, all our operations will be carried out on the system terminal.

1. On the homebrew website “https://brew.sh/ ” copy the terminal download link “/bin/bash -c "$(curl -fsSLhttps://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh” to run in the terminal and download the required files.

2. Download the wget command. In put ”brew install wget ” without which we will not be able to download the files required for the next yolo

3. Installing darknet and compiling

The darknet code: “git clone https://github.com/pjreddie/darknet”

When we have cloned the repository, we can compile the code using the following commands:

“cd darknet” and then input “make”

This will create an executable file called darknet that we can use to run YOLOv3.

Now we need to verify that all the programs installed are correct. Input: “./darknet”, if the output usage:. /darknet <function>, which means we are ready for the next step. Whenever we need to use YOLOv3, we need to cd darknet first and then run

4. Next step is to download the YOLOv3 weights:

YOLOv3 requires pre-trained weights to make predictions. Therefore, we need to download the weights from the official website using the following command:

“wget https://pjreddie.com/media/files/yolov3.weights”

5. Run YOLOv3:

At last, run YOLOv3 on an image or a video using the following command:

“./darknet detector test cfg/coco.data cfg/yolov3.cfg yolov3.weights -thresh 0.25 data/xxx.jpg.”

The image can be any name, and we will use XXX instead. This will detect objects in the XXX.jpg image using the YOLOv3 model with a confidence threshold of 0.25. Here we have randomly selected images for testing and obtained the full detection results, figure 6.4

The final output image is in the darknet directory and is automatically named predictions.jpg. It is important to note that each time an image is detected, it will automatically overwrite the previous one.

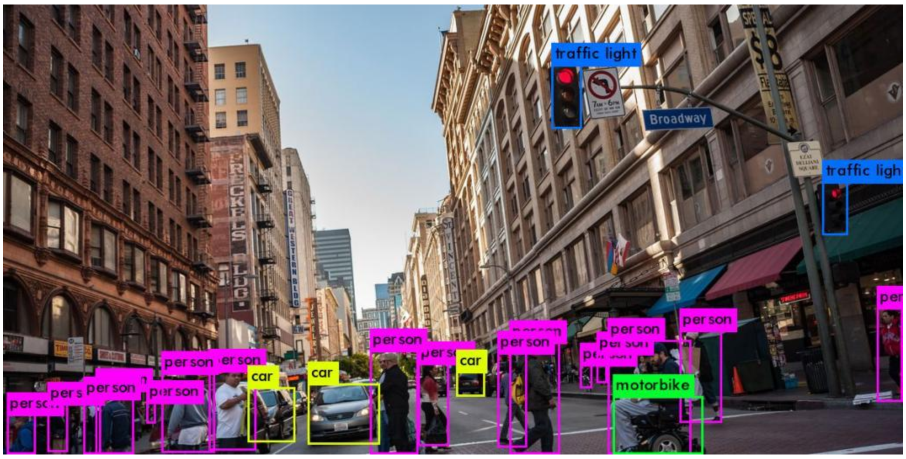


Figure 6.4 YOLOv3 Image detection test-1

YOLOv3 Image Detection Results

Inflow are the results of our random image inspection Figure6.5 and the results of our virtual city screenshot inspection figure6.5 and figure6.7

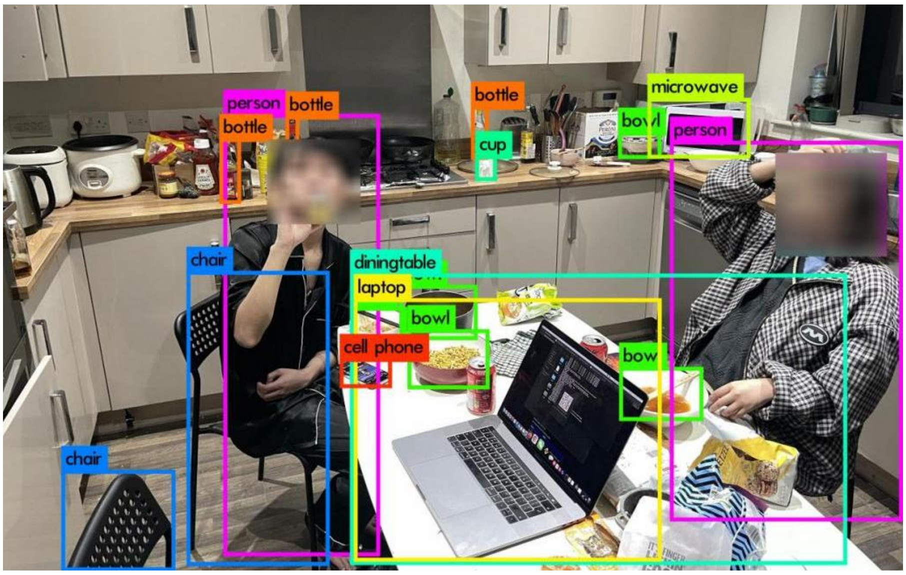


Figure 6.5 Test-2

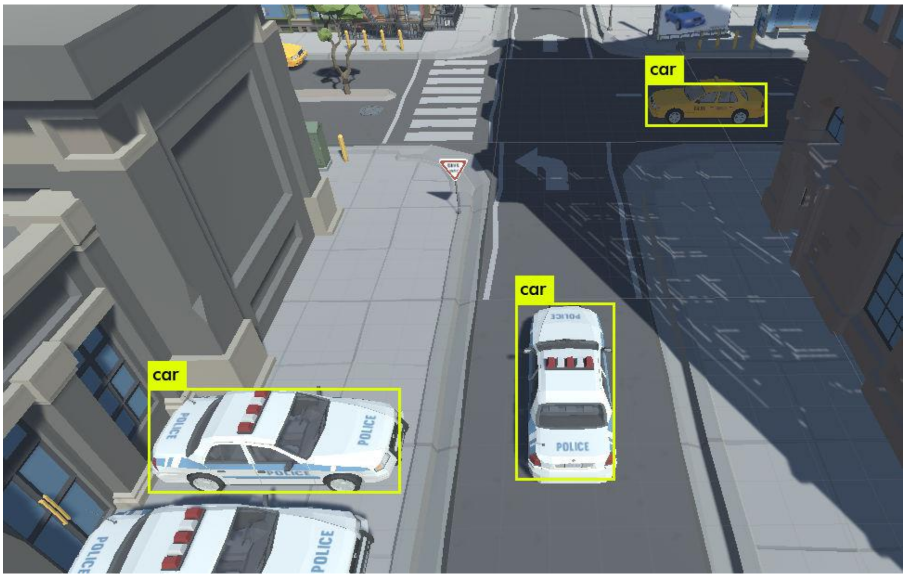


Figure 6.6 Test-3

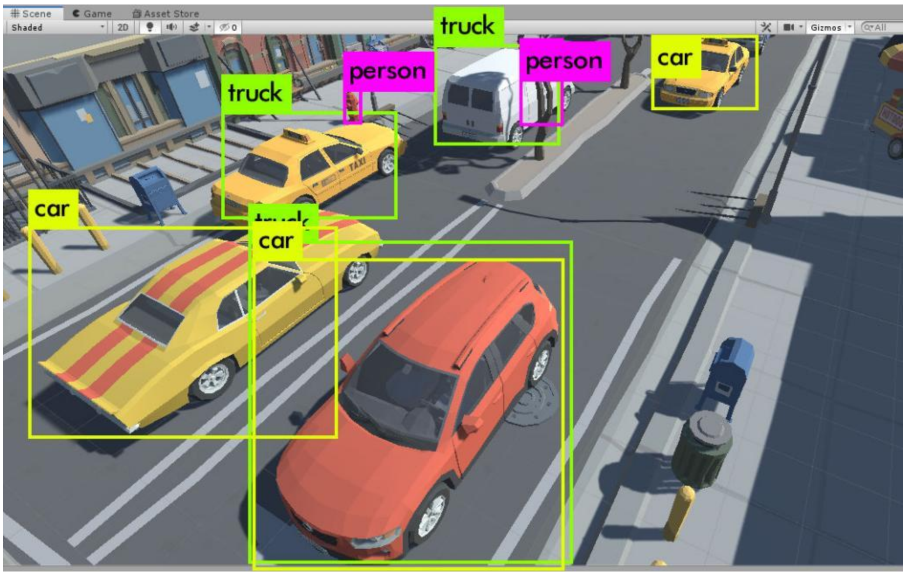


Figure 6.7 Test-3

Errors and Summary

Errors

We encountered these main problems in the code session when building the YOLOv3 environment, and solutions are provided.

When we finish downloading Darknet input make, if we get: XCRUN: ERROR: INVALID ACTIVE DEVELOPER PATH (/LIBRARY/DEVELOPER/ COMMANDLINETOOLS), MISSING XCRUN AT: Execute: xcode- select --install *

If we got: error: RPC failed; curl 56 SSLRead () return error -36 Execute: Input: git config --global http.postBuffer 524288000 or 2.Input: sudo xcode-select -switch / Applications/Xcode.app/Contents/Developer

If we only have one xcode, then this command is helpful; but if we have multiple xcodes, we need to change it to the following: sudo xcode-select -switch /Applications/Xcode 7.3.1.app/Contents/Developer.

Smmary

YOLOv3 is the most widely used of the earlier versions of the YOLO algorithm. However, there are still some problems with the results from our data, with a detection error in Figure 6.9, where two fire hydrants are treated as two pedestrians. Also, in figure 6.7, it was not possible to recognise all the information in the photo for two main reasons: 1 YOLOv3 uses a cubic detection algorithm (Dhyanjith, Manohar and Raj,2021), so it is inaccurate or infeasible to recognise very dense targets in the area. 2 As this is a preliminary trial, we only tried a little data training, resulting in some detected targets being out of the algorithm's recognition range. As we tried YOLOv3 for more training, the more powerful YOLOv5 came into view. The YOLO family of object detection algorithms is updated quickly, with substantial improvements at each stage. After discussion, we finally decided to experiment with the even faster and more accurate YOLOv5 in the direction of video recognition research.6.3 YOLOv5 With Video Detection

6.3 YOLOv5 Video Detection

The YOLOv5 algorithm uses a single neural network architecture to perform object detection on images and videos (Chen, Cao and Wang, 2022). It consists of a backbone network, neck network, and head network (Lawal, 2023). The backbone network is typically a convolutional neural network (CNN) that extracts features from the input image (He, Ma, Wang, 2017). The neck network and head network are responsible for generating bounding boxes and performing object classification.

One of the key improvements of YOLOv5 is its speed. The algorithm can process images and videos at high frame rates, making it suitable for real-time applications(Xiaoping, Jiahui, Zhonghe and Shida, 2021). YOLOv5 is also highly accurate, achieving state-of-the-art performance on several benchmark datasets. Another advantage of YOLOv5 is its flexibility. The algorithm can be trained on a wide range of datasets and can detect a variety of object classes. Additionally, the YOLOv5 algorithm has been optimized for deployment on a variety of platforms, including mobile devices and embedded systems (Nguyen, Q.T., 2022). Compared to YOLOv3, YOLOv5 is a better match for our experimental needs.

6.3.1 The Difference Between YOLOv3 and YOLOv5

YOLOv5 is implemented using the PyTorch deep learning framework, which provides better flexibility and performance than Darknet (Nguyen, Q.T., 2022). However, YOLOv5 is still based on the same fundamental ideas as the original YOLO and Darknet (Thuan, 2021.), which include dividing the input image into a grid of cells and predicting the object classes and bounding boxes for each cell (Redmon, Divvala, Girshick and Farhadi, 2016). So, while YOLOv5 does not use the Darknet framework, it builds on the foundation that Darknet and YOLO laid. Figure 6.8

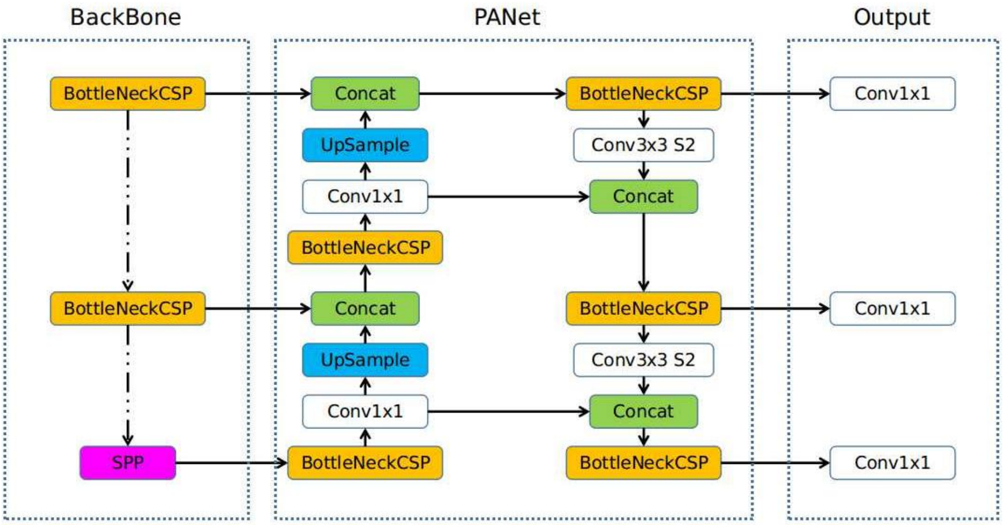
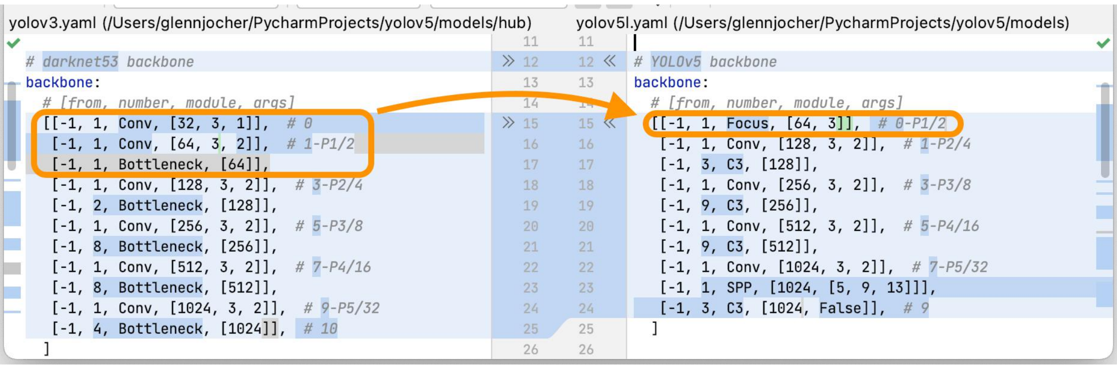


Figure 6.8 Over view of YOLOv5

One of the critical difference with YOLOv3 is that YOLOv5 uses the Focus layer, figure 6.9. Furthermore, the Focus layer is used in the YOLO algorithm for the first time; the Focus layer replaces the first three layers of YOLOv3. (Nepal and Eslamiat, 2022) The use of the Focus layer reduces the number of layers and the memory needed for CUDA and thus increases the speed.



yolov3.yaml (/Users/glennjocher/PycharmProjects/yolov5/models/hub)

```
# darknet53 backbone
backbone:
  # [from, number, module, args]
  [[-1, 1, Conv, [32, 3, 1]],  # 0
   [-1, 1, Conv, [64, 3, 2]],  # 1-P1/2
   [-1, 1, Bottleneck, [64]],
   [-1, 1, Conv, [128, 3, 2]],  # 3-P2/4
   [-1, 2, Bottleneck, [128]],
   [-1, 1, Conv, [256, 3, 2]],  # 5-P3/8
   [-1, 8, Bottleneck, [256]],
   [-1, 1, Conv, [512, 3, 2]],  # 7-P4/16
   [-1, 8, Bottleneck, [512]],
   [-1, 1, Conv, [1024, 3, 2]],  # 9-P5/32
   [-1, 4, Bottleneck, [1024]],  # 10
  ]
```

yolov5l.yaml (/Users/glennjocher/PycharmProjects/yolov5/models)

```
# YOLOv5 backbone
backbone:
  # [from, number, module, args]
  [[-1, 1, Focus, [64, 3]],  # 0-P1/2
   [-1, 1, Conv, [128, 3, 2]],  # 1-P2/4
   [-1, 3, C3, [128]],
   [-1, 1, Conv, [256, 3, 2]],  # 3-P3/8
   [-1, 9, C3, [256]],
   [-1, 1, Conv, [512, 3, 2]],  # 5-P4/16
   [-1, 9, C3, [512]],
   [-1, 1, Conv, [1024, 3, 2]],  # 7-P5/32
   [-1, 1, SPP, [1024, [5, 9, 13]]],
   [-1, 3, C3, [1024, False]],  # 9
  ]
```

Figure 6.9 Focus layer (Glenn-jocher. 2021)

Details of the other changes to YOLOv5 are as follows:

Architecture: YOLOv3 uses a Darknet-53 architecture as its backbone network, while YOLOv5 uses a CSPDarknet53 architecture (Shetty, Saha, Sanghvi, Save, and Patel, 2021). CSPDarknet53 is a more efficient and accurate version of Darknet-53.

Speed and Accuracy: .YOLOv5 has improved performance in terms of mAP (mean average precision)and FPS (frames per second)() on various benchmark datasets (Wu, Wang and Liu, Y2021).YOLOv5 is faster and more accurate than YOLOv3

Object Detection Capability:

YOLOv5 has better object detection capabilities than YOLOv3. It can detect smaller objects with greater accuracy and can detect objects that are partially hidden or occluded.

Training Process: YOLOv5 has a simpler and more streamlined training process compared to YOLOv3. Which uses a single-stage training process with anchor-based object detection (Huang, Cheng, Yang, Lv and Xu, 2022), while YOLOv3 uses a two-stage training process with anchor-free object detection.

Model Size: YOLOv5 has a smaller model size compared to YOLOv3(SZhao, Wang, Liu, Peng yan, 2022). This means that YOLOv5 can be easily deployed on devices with limited memory and computing power.

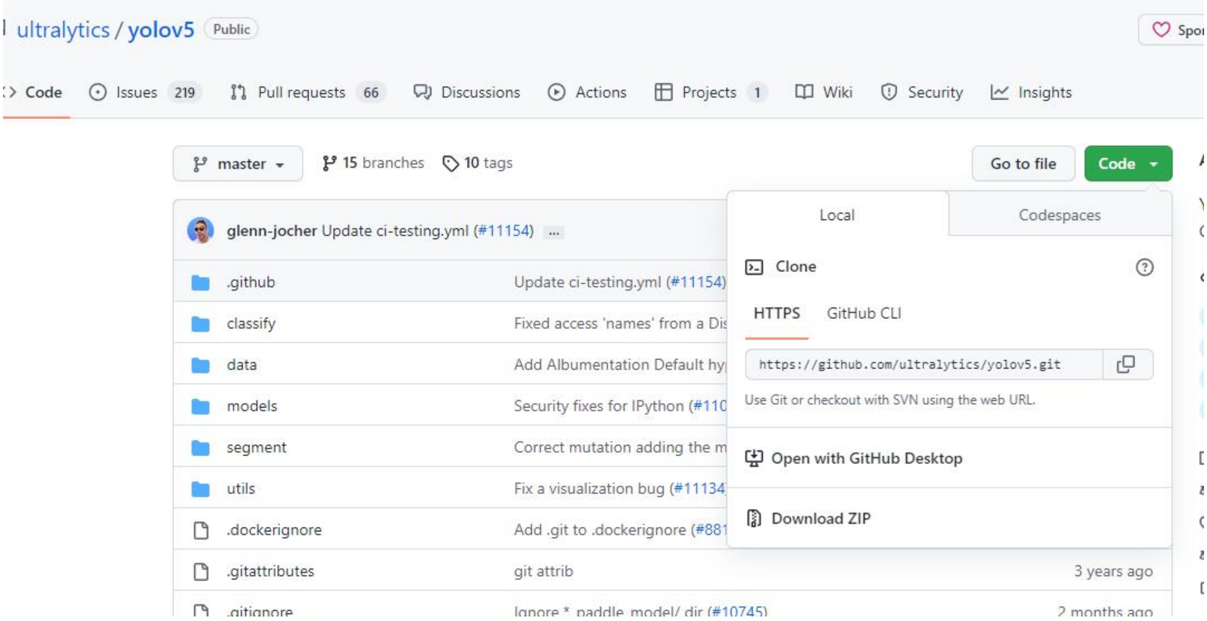
All changes to YOLOv5 we can understood as an update from YOLOv3, which the most widely used version of the YOLO detection algorithm up to 2022.

6.3.2 YOLOv5 Installation and Testing

YOLOv5 Installation

The installation steps for YOLOv5 are tedious, starting with installing Anaconda with Pycharm. Anaconda is an open-source Python package manager containing over 180 scientific packages (Rolon-Mérette, Ross, Rolon-Mérette and Church, 2016) and their dependencies and is a Python data science and machine learning development platform based on conda. Anaconda can be used for environment isolation in addition to providing rich databases. When we install a Python program or library, these packages are installed by default under the site-packages directory of the current Python environment. However, only one copy of the same library can exist in an environment. What often happens is that two or more programs need to rely on the same library, but have different library version requirements: program A relies on library xxx, which requires version >= 1.23.0, and program B also relies on library xxx but requires a version lower than <= 1.21.0, which can lead to version conflicts. So the point of a virtual environment is to prevent conflicts caused by different python projects relying on different versions.

The YOLOv5 source code from Github, (https://github.com/ultralytics/yolov5), figure 6.10



glenn-jocher Update ci-testing.yml (#11154) ...

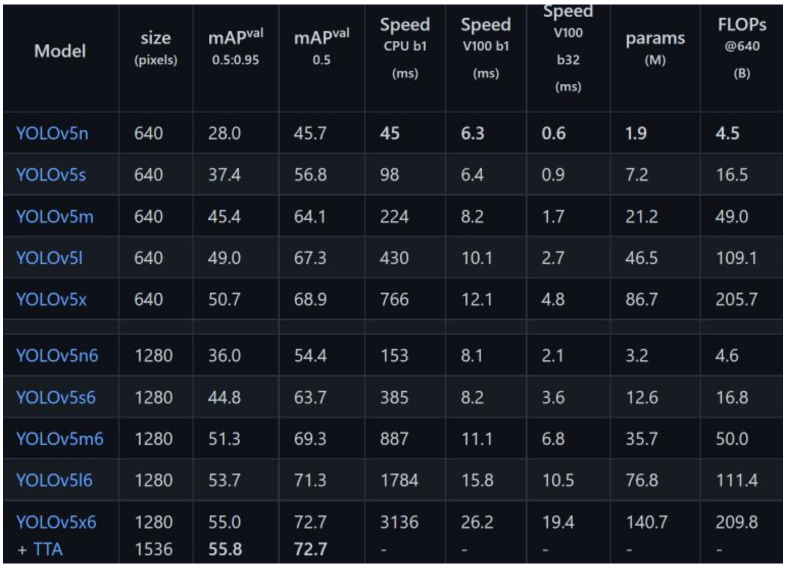
https://github.com/ultralytics/yolov5.git

Use Git or checkout with SVN using the web URL.

.github	Update ci-testing.yml (#11154)	
classify	Fixed access 'names' from a Di	
data	Add Albumentation Default hy	
models	Security fixes for IPython (#110	
segment	Correct mutation adding the m	
utils	Fix a visualization bug (#11134	
.dockerignore	Add .git to .dockerignore (#881	
.gitattributes	git attrib	3 years ago
.gitignore	Ignore * paddle_model/ dir (#10745)	2 months ago

Figure 6.10 YOLOv5 source code

Next, the pre-trained model needs to be downloaded. In order to shorten the training time and achieve better accuracy, we usually load pre-training weights for the network. yolov5 version 6.2 provides us with several pre-training weights, and we can choose different pre-training weights according to our different needs. Figure 6.11. The installed pre-trained model should be placed in the YOLO folder. In our project, we have chosen the widely used YOLOv5s version.



Model	size (pixels)	mAP^{val} 0.5:0.95	mAP^{val} 0.5	Speed CPU b1 (ms)	Speed V100 b1 (ms)	Speed V100 b32 (ms)	params (M)	FLOPs @640 (B)
YOLOv5n	640	28.0	45.7	**45**	**6.3**	**0.6**	**1.9**	**4.5**
YOLOv5s	640	37.4	56.8	98	6.4	0.9	7.2	16.5
YOLOv5m	640	45.4	64.1	224	8.2	1.7	21.2	49.0
YOLOv5l	640	49.0	67.3	430	10.1	2.7	46.5	109.1
YOLOv5x	640	50.7	68.9	766	12.1	4.8	86.7	205.7
YOLOv5n6	1280	36.0	54.4	153	8.1	2.1	3.2	4.6
YOLOv5s6	1280	44.8	63.7	385	8.2	3.6	12.6	16.8
YOLOv5m6	1280	51.3	69.3	887	11.1	6.8	35.7	50.0
YOLOv5l6	1280	53.7	71.3	1784	15.8	10.5	76.8	111.4
YOLOv5x6 + TTA	1280 1536	55.0 **55.8**	72.7 **72.7**	3136 -	26.2 -	19.4 -	140.7 -	209.8 -

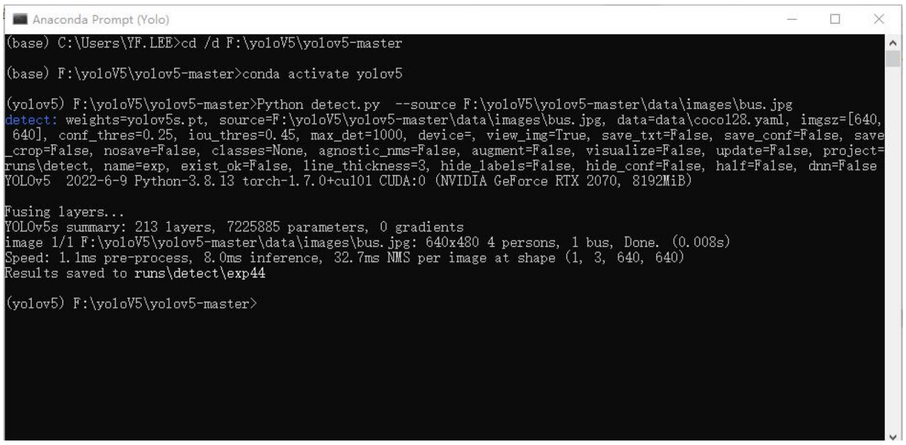
Figure 6.11 YOLOv5 pre-trained model

At last we are going to install yolov5 dependencies. Because Python lacks a range of packages such as Numpy, Matplotlib, Scipy, Scikit-learn, etc., we need pip install to import these packages to perform the appropriate operations (input: pip install -r requirements.txt to install requirements.txt). The next step is to install Cuda and Cudnn. Each version of Cuda corresponds to a different version of Cudnn. Our project used Cuda version 10.02 and Cudnn versions 1.60/1.50. Finally, we tested and successfully ran the detect.py file in the yolov5 folder in cmd.

Testing

Before each detection, we need to input: cd /d F:\yoloV5\yolov5-master, find the program's location and then activate it: conda activate yolov5. Finally, to identify it, input:

Python detect.py --source F:\yoloV5\yolov5-master\data\images\bus.jpg, where --source is followed by the path of the target image or video to be detected, Figure 6.12.



```
(base) C:\Users\YF.LEE>cd /d F:\yoloV5\yolov5-master

(base) F:\yoloV5\yolov5-master>conda activate yolov5

(yolov5) F:\yoloV5\yolov5-master>Python detect.py  --source F:\yoloV5\yolov5-master\data\images\bus.jpg
detect: weights=yolov5s.pt, source=F:\yoloV5\yolov5-master\data\images\bus.jpg, data=data\coco128.yaml, imgsz=[640,
 640], conf_thres=0.25, iou_thres=0.45, max_det=1000, device=, view_img=True, save_txt=False, save_conf=False, save
_crop=False, nosave=False, classes=None, agnostic_nms=False, augment=False, visualize=False, update=False, project=
runs\detect, name=exp, exist_ok=False, line_thickness=3, hide_labels=False, hide_conf=False, half=False, dnn=False
YOLOv5  2022-6-9 Python-3.8.13 torch-1.7.0+cu101 CUDA:0 (NVIDIA GeForce RTX 2070, 8192MiB)

Fusing layers...
YOLOv5s summary: 213 layers, 7225885 parameters, 0 gradients
image 1/1 F:\yoloV5\yolov5-master\data\images\bus.jpg: 640x480 4 persons, 1 bus, Done. (0.008s)
Speed: 1.1ms pre-process, 8.0ms inference, 32.7ms NMS per image at shape (1, 3, 640, 640)
Results saved to runs\detect\exp44

(yolov5) F:\yoloV5\yolov5-master>
```

Figure 6.12 YOLOv5 detecting

We then compare the results of YOLOv3 with those of YOLOv5, and it is clear that YOLOv5 is much better at detecting small objects. Figure 6.13

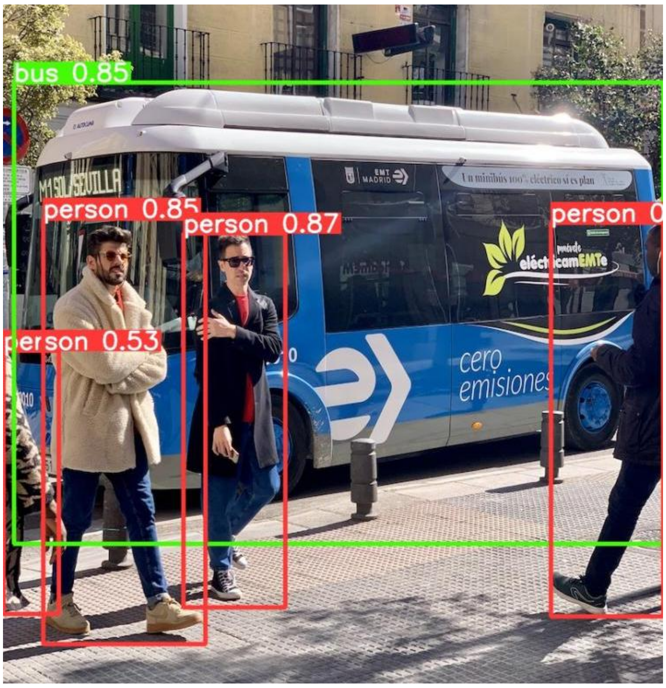


Figure 6.13 Test image

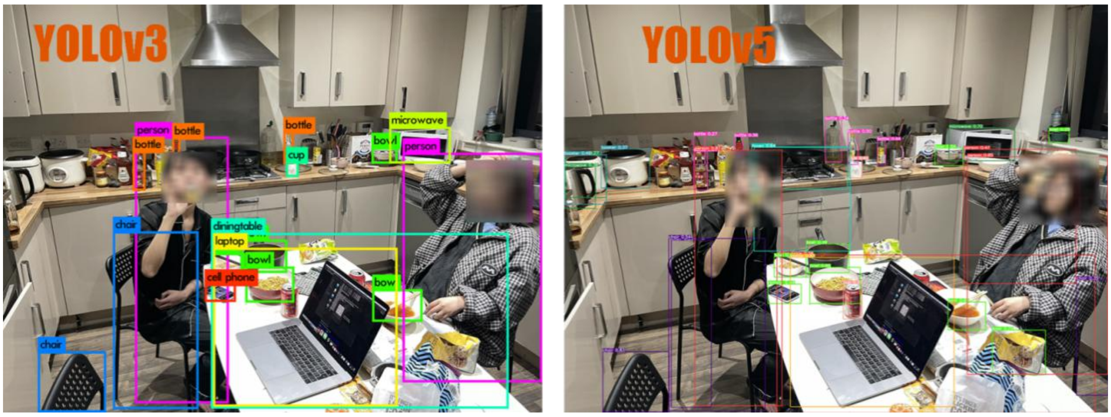
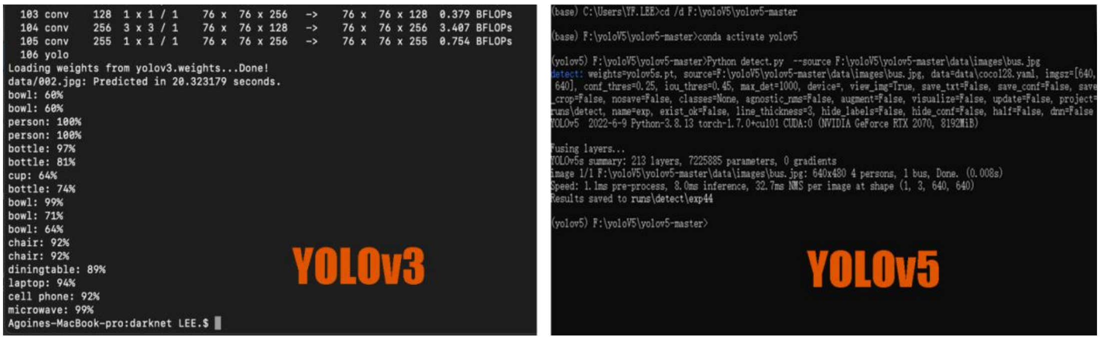


Figure 6.14 The different results between YOLOv3 and YOLOv5

From the comparison results, we can quickly tell that YOLOv5 can detect more objects and is more accurate. However, the detection box in Window's YOLOv5 is less evident than in Mac Os, probably due to the system separation rate.

Next is the speed comparison. Figure 6.15 shows the detection of YOLOv3 and YOLOv5 in the same image. A visual comparison of the images shows a considerable difference between the two versions of the YOLO algorithm, with YOLOv3 detecting life photos in around 20.3 seconds, while YOLOv5 is incredibly fast at less than 1 second. However, there are computer performance and system differences in our comparison results, but it is enough to prove that YOLOv5 is much faster at target detection than YOLOv3



```
  103 conv    128  1 x 1 / 1    76 x  76 x 256   ->    76 x  76 x 128  0.379 BFLOPs
  104 conv    256  3 x 3 / 1    76 x  76 x 128   ->    76 x  76 x 256  3.407 BFLOPs
  105 conv    255  1 x 1 / 1    76 x  76 x 256   ->    76 x  76 x 255  0.754 BFLOPs
  106 yolo
Loading weights from yolov3.weights...Done!
data/002.jpg: Predicted in 20.323179 seconds.
bowl: 60%
bowl: 60%
person: 100%
person: 100%
bottle: 97%
bottle: 81%
cup: 64%
bottle: 74%
bowl: 99%
bowl: 71%
bowl: 64%
chair: 92%
chair: 92%
diningtable: 89%
laptop: 94%
cell phone: 92%
microwave: 99%
Agoines-MacBook-pro:darknet LEE.$
```

```
(base) C:\Users\YF.LEE>cd /d F:\yoloV5\yolov5-master

(base) F:\yoloV5\yolov5-master>conda activate yolov5

(yolov5) F:\yoloV5\yolov5-master>Python detect.py  --source F:\yoloV5\yolov5-master\data\images\bus.jpg
detect: weights=yolov5s.pt, source=F:\yoloV5\yolov5-master\data\images\bus.jpg, data=data\coco128.yaml, imgsz=[640,
 640], conf_thres=0.25, iou_thres=0.45, max_det=1000, device=, view_img=True, save_txt=False, save_conf=False, save
_crop=False, nosave=False, classes=None, agnostic_nms=False, augment=False, visualize=False, update=False, project=
runs\detect, name=exp, exist_ok=False, line_thickness=3, hide_labels=False, hide_conf=False, half=False, dnn=False
YOLOv5  2022-6-9 Python-3.8.13 torch-1.7.0+cu101 CUDA:0 (NVIDIA GeForce RTX 2070, 8192MiB)

Fusing layers...
YOLOv5s summary: 213 layers, 7225885 parameters, 0 gradients
image 1/1 F:\yoloV5\yolov5-master\data\images\bus.jpg: 640x480 4 persons, 1 bus, Done. (0.008s)
Speed: 1.1ms pre-process, 8.0ms inference, 32.7ms NMS per image at shape (1, 3, 640, 640)
Results saved to runs\detect\exp44

(yolov5) F:\yoloV5\yolov5-master>
```

Figure 6.15 Speed comparison

At last, we performed video detection. Below is a screenshot of our work in video recognition. Figure 6.16 is the first attempt at video detection, and figure 6.17 is the video detection of a simulated traffic accident scene in a virtual city.

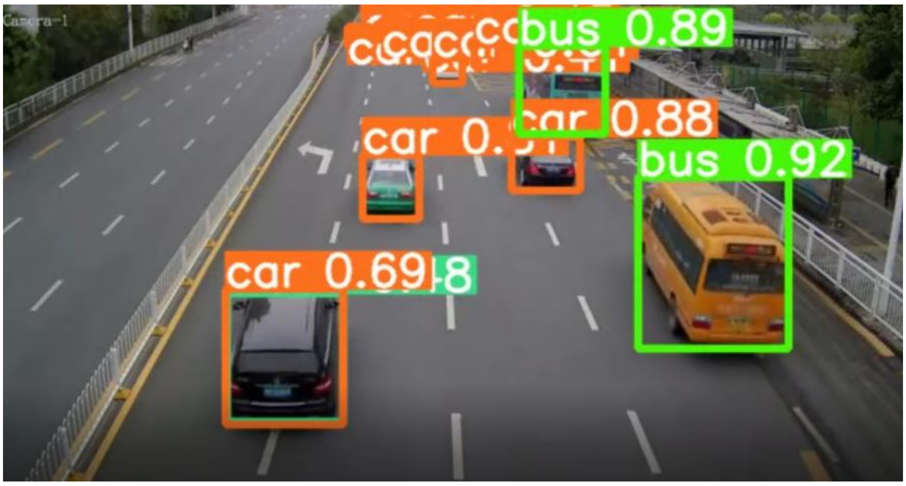


Figure 6.16 Video test-1

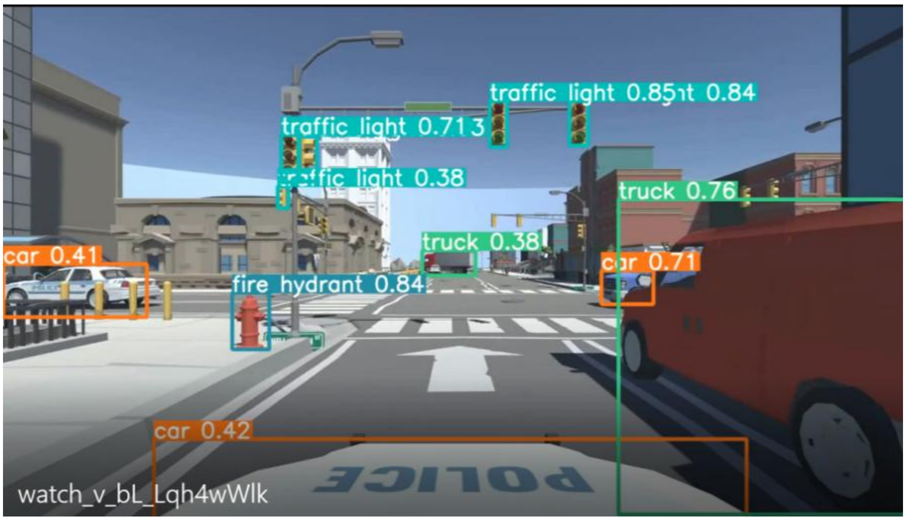


Figure 6.17 Simulated traffic accident scenedetection

6.3.3 Model training

YOLOv5 is a type of object detection algorithm that is designed to identify and locate objects in images (Karthi, Muthulakshmi, Priscilla, Praveen, and Vanisri, 2021). For YOLOv5 to be able to do this effectively, it needs to be trained on a dataset of images annotated with labels indicating the locations and classes of the objects in the images. During training, the YOLOv5 model learns to identify input image patterns corresponding to different object classes and locations. It does this by adjusting the weights of its neural network based on the errors it makes while predicting the labels of the training images (Redmon, Divvala, Girshick and Farhadi, 2016.). By doing this repeatedly over many iterations, the model gradually improves its accuracy and ability to detect objects in new images.

Training increases the accuracy of YOLOv5's detection of targets and the variety of detections. It would not have learned to identify the patterns that correspond to different object classes and locations, and its predictions would need to be more accurate and complete. Therefore, model training is a crucial step in using YOLOv5 for object detection.

In the introduction to this section, we only briefly discussed model training, as YOLOv5 comes with a database of a specific size so that we can perform preliminary object detection directly. Model training is a single and lengthy process, and the LabelImg tool (Tzutalin - Git code 2015) is the essential model annotation tool used for training.

6.4 Summary

This section focuses on an in-depth discussion and comparison of the YOLOv3 and YOLOv5 algorithms. First, in section 6.2, we summarise the changes between the YOLOv3 version and the first version of YOLO, introduce the changes and optimisations, and finally complete the image detection using YOLOv3. In the next section, 6.3, we analyse the upgrades to YOLOv5, summarise the differences with YOLOv3 and introduce the environment build. At last, the traffic accident simulation in the virtual city was successfully detected using YOLOv5.

In this chapter, we introduce two versions of YOLO, and this is because each version of the YOLO algorithm is updated too quickly, and the efficiency and accuracy of object recognition are substantially improved. In order to get more accurate experimental data, we choose to use the current latest version of YOLO for experimental detection.

The field of computer vision is rapidly evolving, with new techniques and ideas emerging all the time. As a result, YOLO and other object detection algorithms need to be updated frequently to incorporate the latest research and advancements. As the use cases for object detection expand and become more complex, there is a growing demand for faster and more accurate algorithms that can handle larger datasets and more diverse object categories. This demand drives the rapid updates to YOLO versions.

Chapter7 Conclusion and Future Work

7.1 Conclusion

Motor traffic is an integral part of contemporary society, and its problems are magnified as society develops. Increasingly, car accidents have become a significant problem affecting people's daily travel. One of the significant factors contributing to traffic accidents is the driver's blind spots. According to a report by the UK Department for Transport, over 13,000 accidents occurred in the UK in 2019 as a result of drivers failing to look properly, which includes accidents caused by blind spots. When a driver changes lanes, turns or overtakes, it is difficult to detect the presence of cars, pedestrians or other objects in the driver's blind spot, which can lead to car collisions and even injuries. As technology advances, we try to experiment with more scientific solutions to the problems caused by drivers' blind spots. Combining object detection technology with driver blind spots detection is a promising area of research in automotive safety. Blind spots are areas around a vehicle that are not visible to the driver, which can lead to accidents if not properly monitored. Object detection technology can be used to detect the presence of objects in blind spots and alert the driver to their presence.

Experimenting with the driver's blind spot on real roads is a dangerous project category. Now the state of the art – no ways to safely train AI of accidents without damage to property and harm to persons the solution, using virtual world to train AI of accidents without damage to property and harm to persons the results – recognition of incidents that we trained the AI. To address this issue, we experimented with and successfully tested the driver's blind spot in a virtual environment by constructing virtual accidents without damage to property and human well-being. Firstly, we used Unity3D to build the virtual scene, which was completed with a combination of several building models and the layout of the roads to create the virtual city. We then measured and recorded the extent and angle of the driver's blind spot in real life at rest and reproduced it in the virtual scene. Next, we count the scenarios that cause the most traffic accidents. At last, the dangerous situation due to the driver's blind spot was simulated using a 3D environment.

Ultimately, we settled on YOLO (You only look once) as our detection algorithm. We describe in detail the architecture of the YOLO algorithm and the processing of the target. Two versions of YOLOv3 and YOLOv5 were tested and compared for image recognition. We used YOLOv5 for the final video recognition. To our surprise, YOLOv5 was fast and accurate in detecting the experimental videos, as we performed a simple database training on YOLOv5 during the video recognition phase due to time constraints. We will continue refining our database to improve the speed and accuracy of YOLOv5's detection.

In this project, we combined car blindness and recognition algorithms for object recognition in the driver's blind spot. At the same time, we built a virtual environment of an urban road to test our proposal. We used the YOLO algorithm for the recorded video of the virtual scene to verify the feasibility of the YOLOv5 algorithm for car blind spot detection. The complete urban virtual environment will allow us to perform more tests in the future.

7.2 Future works

For time reasons, we only spent a small quantity of time on additional database training after we got the full test results. We will continue to enhance our database training to improve the speed and accuracy of YOLOv5. YOLO has been updated very quickly, and as of March 2023, YOLOv7 has been announced and is gradually becoming more widely used. We need to keep updating the version used for testing to ensure a faster identification process and more accurate results.

For the development of the project, our next unfinished goal is the detection of distance, where the distance to the target object is calculated from the coordinates, and an alarm or corresponding action is generated if the vehicle's current speed is dangerous concerning the distance between the object.

YOLO is a real-time object detection system that can be used in conjunction with self-driving vehicles to detect and identify objects in real-time. Self-driving vehicles rely heavily on object detection to navigate and avoid obstacles. YOLO is an effective object detection system that can detect objects such as vehicles, pedestrians, traffic signs, and more, in real-time. By integrating YOLO with a self-driving vehicle system, the vehicle can quickly and accurately identify objects in its environment, allowing it to make informed decisions about its movements and adjust its behavior accordingly. This can significantly improve the safety and reliability of self-driving vehicles.

However, it is important to note that object detection is just one part of the self-driving vehicle system. Other components such as mapping, localization, and path planning are also essential for safe and efficient operation of the vehicle.

Bibliography

[1] Improving Road Safety in London, Isabel Dedring, Katherine McKinlay November 19, 2014. https://newcities.org/using-innovative-technology-improve-road-safety-london/

[2] Innovation and safety: creating a safer road transport ecosystem24 October 2019 | By Matthias Maedge - IRU

[3] Reinier J. Jansen, Silvia F. Varotto, Caught in the blind spot of a truck: A choice model on driver glance behavior towards cyclists at intersections, Accident Analysis & Prevention, Volume 174, 2022, 106759, ISSN 0001-4575, https://doi.org/10.1016/j.aap.2022.106759

[4] Mohammed Said Obeidat, Majd M. Rababa, Wa’il R. Tyfour. (2022) Effects of vehicle’s human machine interface devices on driving distractions. Theoretical Issues in Ergonomics Science 23:4, pages 414-434.

[5] Y. L. Chen, B.F. Wu, H.Y. Huang and C. J. Fan, "A real-time vision system for nighttime vehicle detection and traffic surveillance", IEEE Trans Ind Electron, vol. 58, no. 5, pp. 2030-2044, 2011.

[6] J. Verhaevert, "Detection of vulnerable road users in blind spots through Bluetooth Low Energy," 2017 Progress In Electromagnetics Research Symposium - Spring (PIERS), 2017, pp. 227-231, doi: 10.1109/PIERS.2017.8261738.

[7] TWM365288 (U)-Imaging device for car-side blind spot, TW20090204089U 20090316, Classification: - international: B60R1/08; G02B27/01

[8] Russell, D. (2009) Left-hand Drive HGVs and Foreign Truck Drivers in OTS. rep.

[9] Traffic accident simulation impact experiment table, J. JINGWEN. , 2021.

[10] Leudet, J., Mikkonen, T., Christophe, F., Männistö, T. (2018). Virtual Environment for Training Autonomous Vehicles. In: Giuliani, M., Assaf, T., Giannaccini, M. (eds) Towards Autonomous Robotic Systems. TAROS 2018. Lecture Notes in Computer Science(), vol 10965. Springer, Cham. https://doi.org/10.1007/978-3-319-96728-8_14

[11] Jha, N., Srinivasa, D.K., Roy, G., Jagdish, S. and Minocha, R.K., 2004. Epidemiological study of road traffic accident cases: A study from South India. *Indian J Community Med*, *29*(1), pp.20-4.

[12] Ryder, B., Gahr, B., Egolf, P., Dahlinger, A. & Wortmann, F. 2017, "Preventing traffic accidents with in-vehicle decision support systems - The impact of accident

hotspot warnings on driver behaviour", Decision Support Systems, vol. 99, pp. 64-74.

[13] Isaksson-Hellman, I. & Lindman, M. 2018, "An evaluation of the real-world safety effect of a lane change driver support system and characteristics of lane change crashes based on insurance claims data", Traffic injury prevention, vol. 19, no. sup1, pp. S104-S111.

[14] Leading Causes of Car Accidents UK 2022 UPDATED MAY 5, 2022 - ERIN YURDAY, CO-FOUNDER

[15] Transport for London – Casualties in Greater London during 2021. https://www.gov.uk/ Data Release

[16] Social Determinants of Health EDITORS 2018 World Health Organization NUMBER OF PAGES 403 REFERENCE NUMBERS ISBN: 9789241565684

[17] A review of the traffic accidents and related practices worldwide 2022. https://www.researchgate.net/publication/334606486_A_Review_of_the_Traffic_Accidents_and_Related_Practices_Worldwide (Accessed: December 16, 2022).

[18] Transport, D.for (2022) Road accidents and safety statistics, GOV.UK. GOV.UK. Available at: https://www.gov.uk/government/collections/road-accidents-and-safety-statistics

[19] Singh, J., Sahni, M.K., Bilquees, S., Khan, S.S. and Haq, I., 2016. Reasons for road traffic accidents-victims' perspective. International Journal of Medical Science and Public Health, 5(04), p.814.

[20] Tanaboriboon, Y. and Satiennam, T., 2005. Traffic accidents in Thailand. IATSS research, 29(1), pp.88-100.

[21] China's traffic fatality rate is higher than the global average (2014) University of Michigan News. Available at: https://news.umich.edu/zh-hans/%E4%B8%AD%E5% 9B%BD%E4%BA%A4

[22] Road safety questions and answers (2021) RAC Foundation RSS2. Available at: https://www.racfoundation.org/motoring-faqs/safety (Accessed: December 17, 2022).

[23] Levin, G. (2022) 10 most common causes of car accidents, The Levin Firm. Gabriel Levin. Available at: https://www.levininjuryfirm.com/what-is-common-cause-car-accidents/ (Accessed: December 17, 2022).

[24] Hammad, H.M., Ashraf, M., Abbas, F., Bakhat, H.F., Qaisrani, S.A., Mubeen, M., Fahad, S. and Awais, M., 2019. Environmental factors affecting the frequency of road traffic accidents: a case study of sub-urban area of Pakistan. *Environmental*

Science and Pollution Research, *26*(12), pp.11674-11685.

[25] Lavalliere, M., Laurendeau, D., Simoneau, M. and Teasdale, N., 2011. Changing lanes in a simulator: effects of aging on the control of the vehicle and visual inspection of mirrors and blind spot. *Traffic injury prevention*, *12*(2), pp.191-200.

[26] Jonathan (2020) 5 major causes of UK road traffic accidents and how to avoid them, Serious Injury law. Available at: https://www.seriousinjurylaw.co.uk/ resources/blog/5-major-causes-of-uk-road-traffic-accidents-and-how-to-avoid-them/

[27] Kaplan, S., Guvensan, M.A., Yavuz, A.G. & Karalurt, Y. 2015, "Driver Behavior Analysis for Safe Driving: A Survey", IEEE transactions on intelligent transportation systems, vol. 16, no. 6, pp. 3017-3032.

[28] Pecherková, P. and Nagy, I., 2017, May. Analysis of discrete data from traffic accidents. In *2017 Smart City Symposium Prague (SCSP)* (pp. 1-4). IEEE.

[29] Krishnan, P.V., Sheel, V.C., Viswanadh, M.V.S., Shetty, C. and Seema, S., 2018, December. Data analysis of road traffic accidents to minimize the rate of accidents. In *2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS)* (pp. 247-253). IEEE.

[30] Kumar, S., Toshniwal, D. and Parida, M., 2017. A comparative analysis of heterogeneity in road accident data using data mining techniques. *Evolving systems*, *8*(2), pp.147-155.

[31] Curry, A.E., Hafetz, J., Kallan, M.J., Winston, F.K. and Durbin, D.R., 2011. Prevalence of teen driver errors leading to serious motor vehicle cr

[32] ashes. *Accident Analysis & Prevention*, *43*(4), pp.1285-1290.

[33] Thomas, P., Morris, A., Talbot, R. and Fagerlind, H., 2013. Identifying the causes of road crashes in Europe. *Annals of advances in automotive medicine*, *57*, p.13.

[34] How can blind spots hide deadly hazards (2021) Hughey Law Firm. Available at: https://www.hugheylawfirm.com/blind-spots-hide-deadly-hazards/#:~:text=Most%20obviously%2C%20blind%20spots%20put,potentially%20invisible%20to%20other%20drivers. (Accessed: December 20, 2022).

[35] Saito, Y., Sugaya, F., Inoue, S., Raksincharoensak, P. and Inoue, H., 2021. A context-aware driver model for determining recommended speed in blind intersection situations. *Accident Analysis & Prevention*, *163*, p.106447.

[36] Wang, Z., Jin, Q. and Wu, B., 2022, September. Design of a Vision Blind Spot Detection System Based on Depth Camera. In *2022 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive*

Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/ CyberSciTech) (pp. 1-5). IEEE.

[37] The AA Sep 2021 AA. Available at: https://www.theaa.com/driving-school/driving-lessons/advice/blind-spots#:~:text=A%20blind%20spot%20is%20the,or%20pedestrian%20from%20your%20view. (Accessed: December 20, 2022).

[38] Sui, S., Li, T. and Chen, S., 2022, August. A-pillar Blind Spot Display Algorithm Based on Line of Sight. In *2022 IEEE 5th International Conference on Computer and Communication Engineering Technology (CCET)* (pp. 100-105). IEEE.

[39] Ekroll, V., Svalebjørg, M., Pirrone, A., Böhm, G., Jentschke, S., van Lier, R., Wagemans, J. and Høye, A., 2021. The illusion of absence: how a common feature of magic shows can explain a class of road accidents. *Cognitive research: principles and implications*, *6*(1), pp.1-16.

[40] Trucker's blind spots are different (and bigger) than drivers' (2020) Harmon Parker, P.A. Available at: https://www.harmonparkerlaw.com/news/2020/january/truckers-blind-spots-are-different-and-bigger-th/ (Accessed: December 20, 2022).

[41] Truck drivers and blind spots: Bohn & Fletcher (2022) Bohn & Fletcher, LLP. Available at: https://www.bohnlaw.com/2022/03/22/truck-drivers-blind-spots/ (Accessed: December 21, 2022).

[42] Ghosh, C.P. (ed.) (2022) Why are blind spot mirrors essential for your car Carorbis.com. Available at: https://carorbis.com/blog/why-are-blind-spot-mirrors-essential-for-your-car/ (Accessed: December 21, 2022).

[43] Hester, T.S. (2021) Blind-Spot Mirror test, Auto Express. Available at: https://www.autoexpress.co.uk/accessories-tyres/36004/blind-spot-mirror-test (Accessed: December 21, 2022).

[44] Michael Parker, Chapter 20 - Automotive Radar With contributions by Ben Esposito., Editor(s): Michael Parker, Digital Signal Processing 101 (Second Edition), Newnes, 2017,

[45] Vazquez, M.A.R.T.A.M.A.R.T.I.N.E.Z., 2022, Radar for automotive: Why do we need radar? - semicon ductor engineering. Available at:https://semiengineering.com/radar-for-automotive-why-do-we-need-radar/, Accessed: December 22, ., 2022

[46] Shawn CARPENTER. Autonomous vehicle radar: Improving radar performance with simulation. *ANSYS [online]. Canosburg: ANSYS [cit. 2022-12-20]. Dostupne´ z: https://www. ansys. com/about-ansys/advantage-magazine*, 12,

2018.

[47] Martin Schneider. Automotive radar- status and trends. In *German microwave conference*, pages 144– 147, 2005.

[48] Shawn CARPENTER. Autonomous vehicle radar: Improving radar performance with simulation. *ANSYS [online]. Canosburg: ANSYS [cit. 2022-12-21]. Dostupne´ z: https://www. ansys. com/about-ansys/advantage-magazine*, 12, 2018.

[49] Kevin Lim, Paul Treitz, Michael Wulder, Beno ̂ ıt St-Onge, and Martin Flood. LiDAR remote sensing of forest structure. Progress in Physical Geography: Earth and Environment, 27(1):88– 106, mar 2003. doi:10.1191/0309133303pp360ra.

[50] Grimes, D.M. and Jones, T.O., 1974. Automotive radar: A brief review. *Proceedings of the IEEE*, *62*(6), pp.804-822.

[51] Khader, M. and Cherian, S., 2020. An introduction to automotive lidar. *Texas Instruments*.

[52] Neal, A. (2018) Lidar vs. Radar, Fierce Electronics. Available at: https://www.fierceelectronics.com/components/lidar-vs-radar#:~:text=The%20RADAR%20system%20works%20in,light%20waves%20when%20contacting%20objects. (Accessed: December 22, 2022).

[53] Roriz, R., Cabral, J. and Gomes, T., 2021. Automotive LiDAR technology: A survey. *IEEE Transactions on Intelligent Transportation Systems*.

[54] Li, Y. and Ibanez-Guzman, J., 2020. Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems. *IEEE Signal Processing Magazine*, *37*(4), pp.50-61.

[55] Steinbaeck, J., Steger, C., Holweg, G. and Druml, N., 2017, October. Next generation radar sensors in automotive sensor fusion systems. In *2017 Sensor Data Fusion: Trends, Solutions, Applications (SDF)* (pp. 1-6). IEEE.

[56] Rasshofer, R.H. and Gresser, K., 2005. Automotive radar and lidar systems for next generation driver assistance functions. *Advances in Radio Science*, *3*(B. 4), pp.205-209.

[57] John.C.I (2019) *Lidar and Radar Information*, – *LiDAR and RADAR Information*. Available at: https://lidarradar.com/info/advantages-and-disadvantages-of-radar-systems (Accessed: December 24, 2022).

[58] Warren, M.E., 2019, June. Automotive LIDAR technology. In *2019 Symposium on VLSI Circuits* (pp. C254-C255). IEEE.

[59] Karpathy, A. (2021) *Why lidar is doomed*, *Volt Equity*. Available at: https://www.

voltequity.com/article/why-lidar-is-doomed (Accessed: December 25, 2022).

[60] Guerrero-Ibáñez, J., Zeadally, S. and Contreras-Castillo, J., 2018. Sensor technologies for intelligent transportation systems. *Sensors*, *18*(4), p.1212.

[61] Patole, S.M., Torlak, M., Wang, D. and Ali, M., 2017. Automotive radars: A review of signal processing techniques. *IEEE Signal Processing Magazine*, *34*(2), pp.22-35.

[62] SheldonCoffee, C. (2020) How science fiction movies have influenced technology, The Film Magazine. Available at: https://www.thefilmagazine.com/how-science-fiction-movies-have-influenced-technology/ (Accessed: December 29, 2022).

[63] Tonnis, M., Sandor, C., Klinker, G., Lange, C. and Bubb, H., 2005, October. Experimental evaluation of an augmented reality visualization for directing a car driver's attention. In *Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'05)* (pp. 56-59). IEEE.

[64] A Tunisia-El Mannar University Tunis National School of Engineering b Tunisia Sousse University National Sousse National School of Engineering Procedia Computer Science 73 (2015) 242 – 249

[65] Patel, A. (2020) What is object detection?, Medium. ML Research Lab. Available at: https://medium.com/ml-research-lab/what-is-object-detection-51f9d872ece7 (Accessed: January 2, 2023).

[66] Zou, Z., Shi, Z., Guo, Y. and Ye, J., 2019. Object detection in 20 years: A survey. *arXiv preprint arXiv:1905.05055*.

[67] Choudhury, A. (2022) Top 8 algorithms for object detection, Analytics India Magazine. Available at: https://analyticsindiamag.com/top-8-algorithms-for-object-detection/ (Accessed: January 2, 2023).

[68] Li, Y. and Ibanez-Guzman, J., 2020. Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems. *IEEE Signal Processing Magazine*, *37*(4), pp.50-61.

[69] Ouyang (2022) Blind Spot Monitoring System Advantages and disadvantages, blindspotmonitor. Available at: https://www.blindspotmonitor.com/blind-spot-monitoring-system-advantages/ (Accessed: January 3, 2023).

[70] LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature 521(7553), 436 (2015)

[71] Hong, Z.: A preliminary study on artificial neural network. In: 2011 6th IEEE Joint International Information Technology and Artificial Intelligence Conference, vol. 2, pp. 336–338 (2011)

[72] Dhillon, A. and Verma, G.K., 2020. Convolutional neural network: a review of

models, methodologies and applications to object detection. *Progress in Artificial Intelligence*, *9*(2), pp.85-112.

[73] Wang, X.J., Zhao, L.L., Wang, S.: A novel SVM video object extraction technology. In: 2012 8th International Conference on Natural Computation, pp. 44–48. IEEE (2012)

[74] Rish, I.: An empirical study of the naive Bayes classifier. In: IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, vol. 3, no. 22, pp. 41–46 (2001)

[75] Fukushima, K.: Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biol. Cybern. 36, 193–202 (1998)

[76] Papakostas, M., Giannakopoulos, T., Makedon, F., Karkaletsis, V.: Short-term recognition of human activities using convolutional neural networks. In: 2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), pp. 302–307. IEEE (2016)

[77] Yudistira, N., Kurita, T.: Gated spatio and temporal convolutional neural network for activity recognition: towards gated multimodal deep learning. EURASIP J. Image Video Process. 2017, 85 (2017)

[78] Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2011)

[79] Zhou, X., Gong, W., Fu, W., Du, F.: Application of deep learning in object detection. In: 2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS), pp. 631–634. IEEE (2017)

[80] Ranjan, R., Sankaranarayanan, S., Bansal, A., Bodla, N., Chen, J.-C., Patel, V.M., Castillo, C.D., Chellappa, R.: Deep learning for understanding faces: machines may be just as good, or better, than humans. IEEE Signal Process. Mag. 35(1), 66–83 (2018)

[81] Druzhkov, P.N., Kustikova, V.D.: A survey of deep learning methods and software tools for image classification and object detection. Pattern Recognit. Image Anal. 26(1), 9–15 (2016)

[82] Milyaev, S., Laptev, I.: Towards reliable object detection in noisy images. Pattern Recognit. Image Anal. 27(4), 713–722 (2017)

[83] Zhao, Zhong-Qiu, Peng Zheng, Shou-tao Xu and Xindong Wu. “Object Detection With Deep Learning: A Review.” IEEE Transactions on Neural Networks and Learning Systems 30 (2018): 3212-3232.

[84] Hubel DH, Wiesel TN (1968) Receptive fields and functional architecture of

monkey striate cortex. J Physiol 195:215–243

[85] Yamashita, R., Nishio, M., Do, R.K.G. and Togashi, K., 2018. Convolutional neural networks: an overview and application in radiology. *Insights into imaging*, *9*(4), pp.611-629.

[86] Du, X., Cai, Y., Wang, S. and Zhang, L., 2016, November. Overview of deep learning. In *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)* (pp. 159-164). IEEE.

[87] A. Bordes, X. Glorot, J. Weston, et al. Joint learning of words and meaning representations for open-text semantic parsing, in: Proceedings of the AISTATS, 2012.

[88] T. Mikolov, I. Sutskever, K. Chen, et al., Distributed representations of words and phrases and their compositionality, in: Proceedings of the NIPS, 2013.

[89] D. Ciresan, U. Meier, J. Schmidhuber, Multi-column deep neural networks for image classification, in: Proceedings of the CVPR, 2012.

[90] L. Deng A tutorial survey of architectures, algorithms, and applications for deep learning APSIPA Trans. Signal Inf. Process., 3 (2014), p. e2

[91] Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S. and Lew, M.S., 2016. Deep learning for visual understanding: A review. *Neurocomputing*, *187*, pp.27-48.

[92] Y. LeCun, L. Bottou, Y. Bengio, et al. Gradient-based learning applied to document recognition Proc. IEEE, 86 (11) (1998), pp. 2278-2324

[93] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proceedings of the NIPS, 2012.

[94] M. Zeiler Hierarchical Convolutional Deep Learning in Computer Vision (Ph.D. thesis) (New York University, 2014)

[95] Jeong, W.K., Pfister, H. and Fatica, M., 2011. Medical image processing using GPU-accelerated ITK image filters. In *GPU Computing Gems Emerald Edition* (pp. 737-749). Morgan Kaufmann.

[96] Ñanculef, R., Radeva, P. and Balocco, S., 2020. Training Convolutional Nets to Detect Calcified Plaque in IVUS Sequences. In *Intravascular Ultrasound* (pp. 141-158). Elsevier.

[97] Close C. Szegedy, W. Liu, Y. Jia, et al., Going deeper with convolutions, in: Proceedings of the CVPR, 2015.

[98] M.D. Zeiler, R. Fergus, Stochastic pooling for regularization of deep convolutional neural networks, in: Proceedings of the ICLR, 2013.

[99] M. Zeiler Hierarchical Convolutional Deep Learning in Computer Vision (Ph.D.

thesis) (New York University, 2014)

[100] K. He, X. Zhang, S. Ren, et al., Spatial pyramid pooling in deep convolutional networks for visual recognition, in: Proceedings of the ECCV, 2014.

[101] SChapter 17 - Residential Security System Example Using the Object-Oriented Systems Engineering Method,Sanford Friedenthal, Alan Moore, Rick Steiner,In The MK/OMGPress,A Practical Guide to SysML (Third Edition),Morgan Kaufmann,2015

[102] W. Ouyang, P. Luo, X. Zeng, et al., DeepID-Net: multi-stage and deformable deep convolutional neural networks for object detection, in: Proceedings of the CVPR, 2015.

[103] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proceedings of the NIPS, 2012.

[104] R. Girshick, J. Donahue, T. Darrell, et al., Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the CVPR, 2014.

[105] M. Oquab, L. Bottou, I. Laptev, et al., Learning and transferring mid-level image representations using convolutional neural networks, in: Proceedings of the CVPR, 2014.

[106] W. Hu, T. Tan, L. Wang and S. Maybank, "A survey on visual surveillance of object motion and behaviors", IEEE Transactions on Systems Man and Cybernetics Part C: Applications and Reviews, vol. 34, no. 3, pp. 334-352, Aug 2004.

[107] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features", IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 511-518, Dec 2001.

[108] R. Padilla, C. Costa Filho and M. Costa, "Evaluation of haar cascade classifiers designed for face detection", World Academy of Science Engineering and Technology, vol. 64, pp. 362-365, 2012.

[109] E. Ohn-Bar and M. M. Trivedi, "To boost or not to boost? on the limits of boosted trees for object detection", IEEE International Conference on Pattern Recognition, pp. 3350-3355, Dec 2016.

[110] Z. Sun, G. Bebis and R. Miller, "On-road vehicle detection: A review", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 5, pp. 694-711, May 2006.

[111] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image

recognition", IEEE Conference on Computer Vision and Pattern Recognition, pp. 770-778, Jun 2016.

[112] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation", IEEE Conference on Computer Vision and Pattern Recognition, Jun 2014.

[113] R. Girshick, "Fast r-cnn", IEEE International Conference on Computer Vision, Dec 2015.

[114] S. Ren, K. He, R. Girshick and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks", Advances in Neural Information Processing Systems 28, pp. 91-99, 2015.

[115] J. Dai, Y. Li, K. He and J. Sun, "R-FCN: object detection via region-based fully convolutional networks", CoRR, 2016.

[116] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, et al., "SSD: single shot multibox detector", CoRR, 2015.

[117] J. Redmon and A. Farhadi, "Yolo9000: Better faster stronger", IEEE Conference on Computer Vision and Pattern Recognition, vol. 2017, pp. 7263-7271

[118] R. Padilla, S. L. Netto and E. A. B. da Silva, "A Survey on Performance Metrics for Object-Detection Algorithms," 2020 International Conference on Systems, Signals and Image Processing (IWSSIP), 2020, pp. 237-242, doi: 10.1109/IWSSIP48289.2020.9145130.

[119] He, K., Gkioxari, G., Dollár, P. and Girshick, R., 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 2961-2969).

[120] Bharati, P. and Pramanik, A., 2020. Deep learning techniques—R-CNN to mask R-CNN: a survey. *Computational Intelligence in Pattern Recognition*, pp.657-668.

[121] Theckedath, D. and Sedamkar, R.R., 2020. Detecting affect states using VGG16, ResNet50 and SE-ResNet50 networks. *SN Computer Science*, *1*(2), pp.1-7.

[122] Garipov, T., Podoprikhin, D., Novikov, A. and Vetrov, D., 2016. Ultimate tensorization: compressing convolutional and fc layers alike. *arXiv preprint arXiv:1611.03214*.

[123] Chityala, R.N., Hoffmann, K.R., Bednarek, D.R. and Rudin, S., 2004, May. Region of interest (ROI) computed tomography. In *Medical Imaging 2004: Physics of Medical Imaging* (Vol. 5368, pp. 534-541). SPIE.

[124] Hu, R., Tian, B., Yin, S. and Wei, S., 2018, November. Efficient hardware architecture of softmax layer in deep neural network. In *2018 IEEE 23rd International Conference on Digital Signal Processing (DSP)* (pp. 1-5). IEEE.

[125] Rajeshwari, P., Abhishek, P., Srikanth, P. and Vinod, T., 2019. Object detection: an overview. *Int. J. Trend Sci. Res. Dev.(IJTSRD)*, *3*(1), pp.1663-1665.

[126] Ren, S., He, K., Girshick, R. and Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, *28*.

[127] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).

[128] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[129] Jiang, P., Ergu, D., Liu, F., Cai, Y. and Ma, B., 2022. A Review of Yolo algorithm developments. *Procedia Computer Science*, *199*, pp.1066-1073.

[130] Singla, A., Yuan, L. and Ebrahimi, T., 2016, October. Food/non-food image classification and food categorization using pre-trained googlenet model. In *Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management* (pp. 3-11).

[131] *Mohan-, B.S. et al. (2021) 6 different types of object detection algorithms in Nutshell, MLK - Machine Learning Knowledge. Available at: https://machinelearningknowledge.ai/different-types-of-object-detection-algorithms/ (Accessed: January 9, 2023).*

[132] Arya, M.C. and Rawat, A., 2020. A review on YOLO (You Look Only One)-an algorithm for real time object detection. *J Eng Sci*, *11*, pp.554-7.

[133] Makransky, G., Borre‐Gude, S. and Mayer, R.E., 2019. Motivational and cognitive benefits of training in immersive virtual reality based on multiple assessments. *Journal of Computer Assisted Learning*, *35*(6), pp.691-707.

[134] Jenny Lin, Xingwen Guo, Jingyu Shao, Chenfanfu Jiang, Yixin Zhu, and Song-Chun Zhu. 2016. A virtual reality platform for dynamic human-scene interaction. In SIGGRAPH ASIA 2016 Virtual Reality meets Physical Reality: Modelling and Simulating Virtual Humans and Environments (SA '16). Association for Computing Machinery, New York, NY, USA, Article 11, 1–4. https://doi.org/10.1145/2992138.2992144

[135] Zhu Z, Huang T, Shi B, et al. Progressive pose attention transfer for person image generation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019. 2347–2356

[136] Varol G, Romero J, Martin X, et al. Learning from synthetic humans. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017. 109–117

[137] Ros G, Sellart L, Materzynska J, et al. The synthia dataset: a large collection of synthetic images for semantic segmentation of urban scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016. 3234–3243

[138] Hinterstoisser S, Pauly O, Heibel H, et al. An annotation saved is an annotation earned: using fully synthetic training for object instance detection. 2019. ArXiv: 1902.09967

[139] Liao, M., Song, B., Long, S., He, M., Yao, C. and Bai, X., 2020. SynthText3D: synthesizing scene text images from 3D virtual worlds. *Science China Information Sciences*, *63*(2), pp.1-14.

[140] Mérac, B.R.du (2021) Avsimulation fait monter UN Passager Minoritaire, CFNEWS. CFNEWS .Available at: https://www.cfnews.net/L-actualite/M-A-Corporate/Operations/Augmentation-de-capital/AVSimulation-fait-monter-un-passager-minoritaire-353283 (Accessed: January 19, 2023).

[141] Messaoudi, F., Simon, G. and Ksentini, A., 2015, December. Dissecting games engines: The case of Unity3D. In *2015 international workshop on network and systems support for games (NetGames)* (pp. 1-6). IEEE.

[142] Foffa, A., 2022. Arch-Viz: the emergence of the architectural visualisation industry in the United Kingdom and the business of images. *Entreprises et histoire*, *108*(3), pp.94-111.

[143] Bartneck, C., Soucy, M., Fleuret, K. and Sandoval, E.B., 2015, August. The robot engine—Making the unity 3D game engine work for HRI. In *2015 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)* (pp. 431-437). IEEE.

[144] Derakhshani, D., 2012. *Introducing Autodesk Maya 2013*. John Wiley & Sons.

[145] Liu, J. and Li, C.M., 2011. The application of Maya in film 3D animation design. In *Key Engineering Materials* (Vol. 480, pp. 998-1002). Trans Tech Publications Ltd.

[146] Brenner, C., Haala, N. and Fritsch, D., 2001. Towards fully automated 3d city

model generation. In: E. Baltsavias, A. Grün and L. van Gool (eds), Proc. 3rd Int. Workshop on Automatic Extraction of Man-Made Objects from Aerial and Space Images.

[147] Hongli Liu and Linlin Wang, "Application of virtual reality in ancient buildings-take the summer palace for example", Proceedings of First International Conference on Modelling and Simulation, pp. 59-64, 2008.

[148] Taylor, M.C., Baruya, A. and Kennedy, J.V., 2002. *The relationship between speed and accidents on rural single-carriageway roads* (Vol. 511). Crowthorne: TRL.

[149] Huang, T.C. and Lin, C.Y., 2017. From 3D modeling to 3D printing: Development of a differentiated spatial ability teaching model. *Telematics and Informatics*, *34*(2), pp.604-613.

[150] Rohil, M.K. and Ashok, Y., 2022. Visualization of urban development 3D layout plans with augmented reality. *Results in Engineering*, *14*, p.100447.

[151] *S. Thrun, W. Burgard, and D. Fox. Probabilistic Robotics (Intelligent Robotics and Autonomous Agents). The MIT Press, September 2005. 21946, 1947*

[152] Cipriani, E., Gemma, A. and Nigro, M., 2014. A Road Network Design Model for Large-Scale Urban Network. *Computer-based Modelling and Optimization in Transportation*, pp.139-149.

[153] Cantarella, G.E., Pavone, G. and Vitetta, A., 2006. Heuristics for urban road network design: lane layout and signal settings. *European Journal of Operational Research*, *175*(3), pp.1682-1695.

[154] Kumar, A. and Kumar, A., 2021. Tools for Architectural Visualization. *Immersive 3D Design Visualization: With Autodesk Maya and Unreal Engine 4*, pp.7-15.

[155] Hadipriono, F.C., Duane, J.W., Nemeth, Z.A. and Won, S., 2003. Implementation of a virtual environment for traffic accident simulation. *Journal of Intelligent & Fuzzy Systems*, *14*(4), pp.191-202.

[156] Thakurdesai, H.M. and Aghav, J.V., 2021. Autonomous cars: technical challenges and a solution to blind spot. In *Advances in Computational Intelligence and Communication Technology: Proceedings of CICT 2019* (pp. 533-547). Springer Singapore.

[157] Kiefer, R.J. and Hankey, J.M., 2008. Lane change behavior with a side blind zone alert system. *Accident Analysis & Prevention*, *40*(2), pp.683-690.

[158] National Safety Council (2019) Blind Spot Warning & more, My Car Does What. Available at:https://mycardoeswhat.org/deeper-learning/blind-spot-warning/#:~:

text=The%20detection%20area%20covers%20approximately,side%2C%20rear%2C%20and%20front. (Accessed: February 9, 2023).

[159] Hassan, M.Z. and Zainal Ariffin, H.I., 2013. Development of vehicle blind spot system for passenger car. In *Applied Mechanics and Materials* (Vol. 393, pp. 350-353). Trans Tech Publications Ltd.

[160] England, H., 2021. Design Manual for Roads and Bridges, CD 127-Cross-Sections and Headrooms.

[161] *Response to Milton Rd and Histon Rd consultations* (no date) *Cite This For Me, a Chegg service*. Smarter Cambridge Transport organisation. Available at: https://www.citethisforme.com/cite/website (Accessed: February 16, 2023).

[162] Williamson, A., 2021. Why do we make safe behaviour so hard for drivers?. *Journal of road safety*, *32*(1), pp.24-36.

[163] Summerskill, S., Marshall, R., Cook, S., Lenard, J. and Richardson, J., 2016. The use of volumetric projections in Digital Human Modelling software for the identification of large goods vehicle blind spots. *Applied ergonomics*, *53*, pp.267-280.

[164] Alosco, M.L., Spitznagel, M.B., Fischer, K.H., Miller, L.A., Pillai, V., Hughes, J. and Gunstad, J., 2012. Both texting and eating are associated with impaired simulated driving performance. *Traffic injury prevention*, *13*(5), pp.468-475.

[165] Beresnev, P., Zarubin, D., Tyugin, D. and Pinchin, A., 2022, October. The development of a blind spot monitoring system for commercial vehicles. In *AIP Conference Proceedings* (Vol. 2503, No. 1, p. 080018). AIP Publishing LLC.

[166] *Obeidat, M.S., Altheeb, N.F., Momani, A. and Al Theeb, N., 2022. Analyzing the invisibility angles formed by vehicle blind spots to increase driver's field of view and traffic safety. International journal of occupational safety and ergonomics, 28(1), pp.129-138.*

[167] Fang, W., Love, P.E., Luo, H. and Ding, L., 2020. Computer vision for behaviour-based safety in construction: A review and future directions. Advanced Engineering Informatics, 43, p.100980.

[168] *Zhao, Z.Q., Zheng, P., Xu, S.T. and Wu, X., 2019. Object detection with deep learning: A review. IEEE transactions on neural networks and learning systems, 30(11), pp.3212-3232.*

[169] *P. F. Felzenszwalb et al., "Object detection with discriminatively trained part-based models", IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 9, pp. 1627-1645, Sep. 2010.*

[170] *Pathak, A.R., Pandey, M. and Rautaray, S., 2018. Application of deep learning for object detection. Procedia computer science, 132, pp.1706-1717.*

[171] *Du, J., 2018, April. Understanding of object detection based on CNN family and YOLO. In Journal of Physics: Conference Series (Vol. 1004, p. 012029). IOP Publishing.*

[172] *Morbekar, A., Parihar, A. and Jadhav, R., 2020, June. Crop disease detection using YOLO. In 2020 international conference for emerging technology (INCET) (pp. 1-5). IEEE.*

[173] *Hui, J. https://jhui.github.io/2017/03/15/Fast-R-CNN-and-Faster-R-CNN/, Fast R-CNN and Faster R-CNN."*

[174] *Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y. and Berg, A.C., 2016. Ssd: Single shot multibox detector. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14 (pp. 21-37). Springer International Publishing.*

[175] *Valiati, G.R. and Menotti, D., 2018. A preliminary evaluation of pedestrian detection on real-world video surveillance. In Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV) (pp. 3-9). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).*

[176] *Bharati, P. and Pramanik, A., 2020. Deep learning techniques—R-CNN to mask R-CNN: a survey. Computational Intelligence in Pattern Recognition: Proceedings of CIPR 2019, pp.657-668.*

[177] *Chen, G. and Qin, H., 2022. Class-discriminative focal loss for extreme imbalanced multiclass object detection towards autonomous driving. The Visual Computer, 38(3), pp.1051-1063.*

[178] *Wang, Yuanyuan, Chao Wang, Hong Zhang, Yingbo Dong, and Sisi Wei. 2019. "Automatic Ship Detection Based on RetinaNet Using Multi-Resolution Gaofen-3 Imagery" Remote Sensing 11, no. 5: 531. https://doi.org/10.3390/rs11050531*

[179] *Padilla, R., Netto, S.L. and Da Silva, E.A., 2020, July. A survey on performance metrics for object-detection algorithms. In 2020 international conference on systems, signals and image processing (IWSSIP) (pp. 237-242). IEEE.*

[180] Paneru, S. and Jeelani, I., 2021. Computer vision applications in construction: Current state, opportunities & challenges. *Automation in Construction*, *132*, p.103940.

[181] Xiao, Y., Wang, X., Zhang, P., Meng, F. and Shao, F., 2020. Object detection based on faster R-CNN algorithm with skip pooling and fusion of contextual information. *Sensors*, *20*(19), p.5490.

[182] Thuan, D., 2021. Evolution of Yolo algorithm and Yolov5: The State-of-the-Art object detention algorithm.

[183] Kim, J. and Cho, J., 2021. A set of single YOLO modalities to detect occluded entities via viewpoint conversion. *Applied Sciences*, *11*(13), p.6016.

[184] Kharchenko, V. and Chyrka, I., 2018, July. Detection of airplanes on the ground using YOLO neural network. In *2018 IEEE 17th international conference on mathematical methods in electromagnetic theory (MMET)* (pp. 294-297). IEEE.

[185] Ivašić-Kos, M., Krišto, M. and Pobar, M., 2019, April. Human detection in thermal imaging using YOLO. In *Proceedings of the 2019 5th International Conference on Computer and Technology Applications* (pp. 20-24).

[186] Pham, M.T., Courtrai, L., Friguet, C., Lefèvre, S. and Baussard, A., 2020. YOLO-Fine: One-stage detector of small objects under various backgrounds in remote sensing images. *Remote Sensing*, *12*(15), p.2501.

[187] Kosuge, A., Suehiro, S., Hamada, M. and Kuroda, T., 2022. mmWave-YOLO: A mmWave Imaging Radar-Based Real-Time Multiclass Object Recognition System for ADAS Applications. *IEEE Transactions on Instrumentation and Measurement*, *71*, pp.1-10.

[188] Huang, R., Pedoeem, J. and Chen, C., 2018, December. YOLO-LITE: a real-time object detection algorithm optimized for non-GPU computers. In *2018 IEEE international conference on big data (big data)* (pp. 2503-2510). IEEE.

[189] Jiang, P., Ergu, D., Liu, F., Cai, Y. and Ma, B., 2022. A Review of Yolo algorithm developments. *Procedia Computer Science*, *199*, pp.1066-1073.

[190] Benedict, S., 2022. Object detection techniques and algorithms. In *Deep Learning Technologies for Social Impact*. IOP Publishing.

[191] Jeong, H.J., Park, K.S. and Ha, Y.G., 2018, January. Image preprocessing for efficient training of YOLO deep learning networks. In *2018 IEEE International Conference on Big Data and Smart Computing (BigComp)* (pp. 635-637). IEEE.

[192] Benali Amjoud, A. and Amrouch, M., 2020. Convolutional neural networks backbones for object detection. In *Image and Signal Processing: 9th International Conference, ICISP 2020, Marrakesh, Morocco, June 4–6, 2020, Proceedings 9* (pp. 282-289). Springer International Publishing.

[193] Cha, Y.J., Choi, W. and Büyüköztürk, O., 2017. Deep learning - based crack

damage detection using convolutional neural networks. *Computer‐Aided Civil and Infrastructure Engineering*, *32*(5), pp.361-378.

[194] Zhu, J., Fang, L. and Ghamisi, P., 2018. Deformable convolutional neural networks for hyperspectral image classification. *IEEE Geoscience and Remote Sensing Letters*, *15*(8), pp.1254-1258.

[195] Chen, Y., Jiang, H., Li, C., Jia, X. and Ghamisi, P., 2016. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, *54*(10), pp.6232-6251.

[196] Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2017. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), pp.84-90.

[197] Zhu, X., Su, W., Lu, L., Li, B., Wang, X. and Dai, J., 2020. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*.

[198] Truong, N.Q., Lee, Y.W., Owais, M., Nguyen, D.T., Batchuluun, G., Pham, T.D. and Park, K.R., 2020. SlimDeblurGAN-based motion deblurring and marker detection for autonomous drone landing. *Sensors*, *20*(14), p.3918.

[199] Ghimire, S.U.M.A.N. and Horanont, T., 2017. Monitoring crop health growth and its stand count attributes using uav based precision agriculture: a study in tropical farmland of Thailand. *Doctoral Dissertation*.

[200] Maity, M., Banerjee, S. and Chaudhuri, S.S., 2021, April. Faster r-cnn and yolo based vehicle detection: A survey. In *2021 5th international conference on computing methodologies and communication (ICCMC)* (pp. 1442-1447). IEEE.

[201] Science, O.D.S.C.- O.D. (2018) Overview of the yolo object detection algorithm, Medium. Medium. Available at: https://odsc.medium.com/overview-of-the-yolo-object-detection-algorithm-7b52a745d3e0 (Accessed: March 6, 2023).

[202] Sachdeva, A. (2017) Yolo - 'you only look once' for object detection explained, Medium. diaryofawannapreneur. Available at: https://medium.com/diaryofawannapreneur/yolo-you-only-look-once-for-object-detection-explained-6f80ea7aaa1e (Accessed: March 6, 2023).

[203] Wang, F., Yang, X., Zhang, Y. and Yuan, J., 2020, September. Ship target detection algorithm based on improved YOLOv3. In *Proceedings of the 3rd International Conference on Big Data Technologies* (pp. 162-166).

[204] Ahmad, T., Ma, Y., Yahya, M., Ahmad, B., Nazir, S. and Haq, A.U., 2020. Object

detection through modified YOLO neural network. *Scientific Programming*, *2020*, pp.1-10.

[205] Hosang, J., Benenson, R. and Schiele, B., 2017. Learning non-maximum suppression. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4507-4515).

[206] Nie Y, Sommella P, O'Nils M, Liguori C, Lundgren J. Automatic detection of melanoma with yolo deep convolutional neural networks. In2019 E-Health and Bioengineering Conference (EHB) 2019 Nov 21 (pp. 1-4). IEEE.

[207] Mujahid, A., Awan, M.J., Yasin, A., Mohammed, M.A., Damaševičius, R., Maskeliūnas, R. and Abdulkareem, K.H., 2021. Real-time hand gesture recognition based on deep learning YOLOv3 model. Applied Sciences, 11(9), p.4164.

[208] Won, J.H., Lee, D.H., Lee, K.M. and Lin, C.H., 2019, June. An improved YOLOv3-based neural network for de-identification technology. In 2019 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC) (pp. 1-2). IEEE.

[209] Targ, S., Almeida, D. and Lyman, K., 2016. Resnet in resnet: Generalizing residual architectures. *arXiv preprint arXiv:1603.08029*.

[210] Mao, Q.C., Sun, H.M., Liu, Y.B. and Jia, R.S., 2019. Mini-YOLOv3: real-time object detector for embedded applications. *Ieee Access*, *7*, pp.133529-133538.

[211] Dhyanjith, G., Manohar, N. and Raj, A.V., 2021, July. Helmet Detection Using YOLO V3 And Single Shot Detector. In *2021 6th International Conference on Communication and Electronics Systems (ICCES)* (pp. 1844-1848). IEEE.

[212] Chen, Z., Cao, L. and Wang, Q., 2022. Yolov5-based vehicle detection method for high-resolution UAV images. *Mobile Information Systems*, *2022*..

[213] Lawal, O.M., 2023. YOLOv5-LiNet: A lightweight network for fruits instance segmentation. *Plos one*, *18*(3), p.e0282297.

[214] Ma, X., Dai, Z., He, Z., Ma, J., Wang, Y. and Wang, Y., 2017. Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction. *Sensors*, *17*(4), p.818.

[215] Xiaoping, Z., Jiahui, J., Li, W., Zhonghe, H. and Shida, L., 2021. People's fast moving detection method in buses based on YOLOv5. *Int. J. Sens. Sens. Netw*, *9*, p.30.

[216] Nguyen, Q.T., 2022. Detrimental Starfish Detection on Embedded System: A Case Study of YOLOv5 Deep Learning Algorithm and TensorFlow Lite

framework. *Journal of Computer Sciences Institute*, *23*, pp.105-111.

[217] Thuan, D., 2021. Evolution of Yolo algorithm and Yolov5: The State-of-the-Art object detention algorithm.

[218] Glenn-jocher,2021 Yolov5/torch_utils.py at 17B0F71538D3E9990E0E6C4B5C 7C48375956EFA3 · ultralytics/yolov5, GitHub. Available at: https://github.com/ ultralytics/yolov5/blob/17b0f71538d3e9990e0e6c4b5c7c48375956efa3/utils/torch_utils.py#L96-L102 (Accessed: March 15, 2023).

[219] Nepal, U. and Eslamiat, H., 2022. Comparing YOLOv3, YOLOv4 and YOLOv5 for autonomous landing spot detection in faulty UAVs. *Sensors*, *22*(2), p.464.

[220] Shetty, A.K., Saha, I., Sanghvi, R.M., Save, S.A. and Patel, Y.J., 2021, April. A review: Object detection models. In *2021 6th International Conference for Convergence in Technology (I2CT)* (pp. 1-8). IEEE.

[221] Huang, T., Cheng, M., Yang, Y., Lv, X. and Xu, J., 2022, January. Tiny Object Detection based on YOLOv5. In *2022 the 5th International Conference on Image and Graphics Processing (ICIGP)* (pp. 45-50).

[222] Wu, T.H., Wang, T.W. and Liu, Y.Q., 2021, June. Real-time vehicle and distance detection based on improved yolo v5 network. In *2021 3rd World Symposium on Artificial Intelligence (WSAI)* (pp. 24-28). IEEE.

[223] Su, F., Zhao, Y., Wang, G., Liu, P., Yan, Y. and Zu, L., 2022. Tomato Maturity Classification Based on SE-YOLOv3-MobileNetV1 Network under Nature Greenhouse Environment. *Agronomy*, *12*(7), p.1638.

[224] Rolon-Mérette, D., Ross, M., Rolon-Mérette, T. and Church, K., 2016. Introduction to Anaconda and Python: Installation and setup. *Quant. Methods Psychol*, *16*(5), pp.S3-S11.

[225] Karthi, M., Muthulakshmi, V., Priscilla, R., Praveen, P. and Vanisri, K., 2021, September. Evolution of yolo-v5 algorithm for object detection: automated detection of library books and performace validation of dataset. In *2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)* (pp. 1-6). IEEE.

[226] Labelimg tool. Tzutalin - Git code (2015): https://github.com/ tzutalin/labelImg.

Appendix

YOLOv5 Full Code and Comments on Different Parts of The Code

For the interpretation of the train.py file.

```
  # Directories
  wdir = save_dir / 'weights'
  wdir.mkdir(parents=True, exist_ok=True) # make dir
  last = wdir / 'last.pt'
  best = wdir / 'best.pt'
      results_file = save_dir / 'results.txt'
        # Configure
        plots = not opt.evolve # create plots
        cuda = device.type != 'cpu'
        init_seeds(2 + rank)
        with open(opt.data) as f:
          data_dict = yaml.load(f, Loader=yaml.FullLoader) # data dict
        with torch_distributed_zero_first(rank):
          check_dataset(data_dict) # check
        train_path = data_dict['train']
        test_path = data_dict['val']
        nc = 1 if opt.single_cls else int(data_dict['nc']) # number of classes
        names = ['item'] if opt.single_cls and len(data_dict['names']) != 1 else
data_dict['names'] # class names
        assert len(names) == nc, '%g names found for nc=%g dataset in %s' %
(len(names), nc, opt.data) # check assert True (Continue),False (Terminate
program operation)
        # Model
```

Loading models:

```
if pretrained:
  with torch_distributed_zero_first(rank):
    pass
    # attempt_download(weights) # download if not found locally
  ckpt = torch.load(weights, map_location=device) # load checkpoint
  # print('ckpt1', ckpt)
  if hyp.get('anchors'):
   ckpt['model'].yaml['anchors'] = round(hyp['anchors']) # force autoanchor
  model = Model(opt.cfg or ckpt['model'].yaml, ch=3, nc=nc).to(device) # create ch=3 (Number of input channels)
  exclude = ['anchor'] if opt.cfg or hyp.get('anchors') else [] # exclude keys
  state_dict = ckpt['model'].float().state_dict() # to FP32
  state_dict = intersect_dicts(state_dict, model.state_dict(), exclude=exclude) # intersect
  model.load_state_dict(state_dict, strict=False) # load
  logger.info('Transferred %g/%g items from %s' % (len(state_dict), len(model.state_dict()), weights)) # report
 else:
  model = Model(opt.cfg, ch=3, nc=nc).to(device) # create
```

(Load the pre-trained model if it is available and get the anchors if there are any in hyp.yaml)

```
 # Freeze
 freeze = [] # parameter names to freeze (full or partial)
 for k, v in model.named_parameters():
  v.requires_grad = True # train all layers
  if any(x in k for x in freeze):
   print('freezing %s' % k)
   v.requires_grad = False
```

Set the layers to be frozen, i.e. some of the weights of the model are frozen and will not change during the training of the model, only the weight parameters thought to be in the frozen layers will be trained.

```
 # Optimizer
 nbs = 64 # nominal batch size ;
 accumulate = max(round(nbs / total_batch_size), 1) # accumulate loss before optimizing
 hyp['weight_decay'] *= total_batch_size * accumulate / nbs # scale weight_decay
 logger.info(f"Scaled weight_decay = {hyp['weight_decay']}")

 pg0, pg1, pg2 = [], [], [] # optimizer parameter groups
 for k, v in model.named_modules():
  if hasattr(v, 'bias') and isinstance(v.bias, nn.Parameter):
   pg2.append(v.bias) # biases
  if isinstance(v, nn.BatchNorm2d):
   pg0.append(v.weight) # no decay
  elif hasattr(v, 'weight') and isinstance(v.weight, nn.Parameter):
   pg1.append(v.weight) # apply decay
   # print('pg1', pg1)
```

Optimizer:

```
 if opt.adam:
  optimizer = optim.Adam(pg0, lr=hyp['lr0'], betas=(hyp['momentum'], 0.999)) # adjust beta1 to momentum
 else:
  optimizer = optim.SGD(pg0, lr=hyp['lr0'], momentum=hyp['momentum'], nesterov=True)

 optimizer.add_param_group({'params': pg1, 'weight_decay': hyp['weight_decay']}) # add pg1 with weight_decay
 optimizer.add_param_group({'params': pg2}) # add pg2 (biases)
 logger.info('Optimizer groups: %g .bias, %g conv.weight, %g other' % (len(pg2), len(pg1), len(pg0)))
 del pg0, pg1, pg2

 lf = one_cycle(1, hyp['lrf'], epochs) # cosine 1->hyp['lrf']
    scheduler = lr_scheduler.LambdaLR(optimizer, lr_lambda=lf)
     start_epoch, best_fitness = 0, 0.0
     if pretrained:
      # Optimizer
      # print('ckpt2', ckpt)
      if ckpt['optimizer'] is not None:
       optimizer.load_state_dict(ckpt['optimizer'])
       best_fitness = ckpt['best_fitness']

      # Results
      if ckpt.get('training_results') is not None:
       with open(results_file, 'w') as file:
        file.write(ckpt['training_results']) # write results.txt

      # Epochs
      start_epoch = ckpt['epoch'] + 1 # ckpt['epoch'] = -1
      if opt.resume:
        assert start_epoch > 0, '%s training to %g epochs is finished, nothing to
resume.' % (weights, epochs)
      if epochs < start_epoch:
        logger.info('%s has been trained for %g epochs. Fine-tuning for %g additional
epochs.' %
            (weights, ckpt['epoch'], epochs))
        epochs += ckpt['epoch'] # finetune additional epochs

      del ckpt, state_dict
```

Load the saved best_fitness, epoch from the pre-trained model and use it as the starting best_fitness and epoch for the training model (default is -1 in the source code)

```
    gs = int(model.stride.max()) # grid size (max stride)
    nl = model.model[-1].nl # number of detection layers (used for scaling hyp['obj']);
    imgsz, imgsz_test = [check_img_size(x, gs) for x in opt.img_size] # verify imgsz
are gs-multiples imgsz
```

Data generators:

```
    dataloader, dataset = create_dataloader(train_path, imgsz, batch_size, gs, opt,
                    hyp=hyp, augment=True, cache=opt.cache_images, rect=opt.rect,
rank=rank,
                    world_size=opt.world_size, workers=opt.workers,
                    image_weights=opt.image_weights, quad=opt.quad)

    mlc = np.concatenate(dataset.labels, 0)[:, 0].max() # max label class;
    nb = len(dataloader) # number of batches
    assert mlc < nc, 'Label class %g exceeds nc=%g in %s. Possible class labels
are 0-%g' % (mlc, nc, opt.data, nc - 1)
    def cache_labels(self, path=Path('./labels.cache')):
      # Cache dataset labels, check images and read shapes
      x = {} # dict
    nm, nf, ne, nc = 0, 0, 0, 0 # number missing, found, empty, duplicate
    pbar = tqdm(zip(self.img_files, self.label_files), desc='Scanning images', total=len(self.img_files))
    for i, (im_file, lb_file) in enumerate(pbar): ## im_file is jpg,lb_file is txt with labels and coordinates.
      try:
        # verify images
        im = Image.open(im_file)
        im.verify() # PIL verify (Verify that the image is not corrupted)
        shape = exif_size(im) # image size
        assert (shape[0] > 9) & (shape[1] > 9), 'image size <10 pixels'

        # verify labels
        if os.path.isfile(lb_file):
          nf += 1 # label found (Labels can be found for)        with open(lb_file, 'r') as f:
            l = np.array([x.split() for x in f.read().strip().splitlines()], dtype=np.float32) # labels
          if len(l):
            assert l.shape[1] == 5, 'labels require 5 columns each'
            assert (l >= 0).all(), 'negative labels' #
(# Negative labeling is wrong )
assert (l[:, 1:] <= 1).all(), 'non-normalized or out of bounds coordinate labels'
            assert np.unique(l, axis=0).shape[0] == l.shape[0], 'duplicate labels'
          else:
              ne += 1 # label empty with txt, but nothing in txt file, with xml file, but xml file is empty
            l = np.zeros((0, 5), dtype=np.float32) ## with xml file, but no tags, load tag as [], empty set
        else:
          #No txt's not loading
          nm += 1 # label missing No txt, can't find the tag
          l = np.zeros((0, 5), dtype=np.float32)

        x[im_file] = [l, shape] ##
except Exception as e:
        nc += 1
        print('WARNING: Ignoring corrupted image and/or label %s: %s' % (im_file, e))

      pbar.desc = f"Scanning '{path.parent / path.stem}' for images and labels... " \
            f"{nf} found, {nm} missing, {ne} empty, {nc} corrupted"

    if nf == 0:
      print(f'WARNING: No labels found in {path}. See {help_url}')

    x['hash'] = get_hash(self.label_files + self.img_files)
    x['results'] = [nf, nm, ne, nc, i + 1]
    torch.save(x, path) # save for next time
    logging.info(f"New cache created: {path}")
    return x
```

Parameters and category weights:

```
          # Model parameters
          hyp['cls'] *= nc / 80. # scale hyp['cls'] to class count;
          hyp['obj'] *= imgsz ** 2 / 640. ** 2 * 3. / nl # scale hyp['obj'] to image size and output layers
          model.nc = nc # attach number of classes to model
          model.hyp = hyp # attach hyperparameters to model
          model.gr = 1.0 # iou loss ratio (obj_loss = 1.0 or iou)
          model.class_weights = labels_to_class_weights(dataset.labels, nc).to(device) * nc # attach class weights
          model.names = names
```

labels_to_class_weights ;

```
def labels_to_class_weights(labels, nc=80):
  # Get class weights (inverse frequency) from training labels
  if labels[0] is None: # no labels loaded
    return torch.Tensor()
  labels = np.concatenate(labels, 0) # labels.shape = (866643, 5) for COCO
  classes = labels[:, 0].astype(np.int) # labels = [class xywh]
  weights = np.bincount(classes, minlength=nc) # occurrences per class; number of each category; e.g. label [1,2,4,5,1] weights as [0,2,1,0,1,1]

  weights[weights == 0] = 1 # replace empty bins with 1; convert category number 0 to 1
  weights = 1 / weights # number of targets per class; use the reciprocal of the number of each class as its weight value
  weights /= weights.sum() # normalize; normalize the weights of each class
      return torch.from_numpy(weights)

          # Update image weights (optional)
          if opt.image_weights:
            # Generate indices
            if rank in [-1, 0]:
              cw = model.class_weights.cpu().numpy() * (1 - maps) ** 2 / nc # class weights; The map for each category, which is updated at test time, is given a small weight for the next training for large maps and a large weight for small maps when test.test is performed.
              iw = labels_to_image_weights(dataset.labels, nc=nc, class_weights=cw) # image weights;; iw weight of each image selected, probability
              dataset.indices = random.choices(range(dataset.n), weights=iw, k=dataset.n) # rand weighted idx; from all training sets (based on the index), select all images in the training set, those with large category weights are selected multiple times, those with small weights are selected less often, or not even once
            # Broadcast if DDP
            if rank != -1:
              indices = (torch.tensor(dataset.indices) if rank == 0 else torch.zeros(dataset.n)).int()
              dist.broadcast(indices, 0)
              if rank != 0:
                dataset.indices = indices.cpu().numpy()
```

Warmup and forward propagation:

```
          for i, (imgs, targets, paths, _) in pbar: # batch
```

```
------------------------------------------------------------
        ni = i + nb * epoch # number integrated batches (since train start); ni is the
total number of images
        imgs = imgs.to(device, non_blocking=True).float() / 255.0 # uint8 to float32,
0-255 to 0.0-1.0

          # Warmup; nw: total number of images for warmup.
        if ni <= nw:
          xi = [0, nw] # x interp
          # model.gr = np.interp(ni, xi, [0.0, 1.0]) # iou loss ratio (obj_loss = 1.0 or iou)
          accumulate = max(1, np.interp(ni, xi, [1, nbs / total_batch_size]).round())
          for j, x in enumerate(optimizer.param_groups):
            # bias lr falls from 0.1 to lr0, all other lrs rise from 0.0 to lr0
            x['lr'] = np.interp(ni, xi, [hyp['warmup_bias_lr'] if j == 2 else 0.0, x['initial_lr'] *
lf(epoch)])
            if 'momentum' in x:
              x['momentum'] = np.interp(ni, xi, [hyp['warmup_momentum'],
hyp['momentum']])

        # Forward
        with amp.autocast(enabled=cuda): ## (Semi-precision calculations for faster
training)
          pred = model(imgs) # forward
          loss, loss_items = compute_loss(pred, targets.to(device), model) # loss
scaled by batch_size
          # print('loss, loss_items', loss, loss_items)
          if rank != -1:
            loss *= opt.world_size # gradient averaged between devices in DDP mode
          if opt.quad:
            loss *= 4.

        # Backward
        scaler.scale(loss).backward()
```

Loss function calculation:

```
    def compute_loss(p, targets, model): # predictions, targets, model
    device = targets.device
    lcls, lbox, lobj = torch.zeros(1, device=device), torch.zeros(1, device=device), torch.zeros(1, device=device)
      tcls, tbox, indices, anchors = build_targets(p, targets, model) # targets tcls (3,808) denotes the class of the gt box corresponding to the 3 detection heads, tobox (3, ([808,4])) denotes the gt box corresponding to the 3 detection heads xywh; 808 means 808 boxes
      # anchors (3, ([802,2])), indicating the anchor corresponding to the 808 gt boxes for each detection header h = model.hyp # hyperparameters

    # Define criteria
    BCEcls = nn.BCEWithLogitsLoss(pos_weight=torch.tensor([h['cls_pw']], device=device)) # weight=model.class_weights) It combines nn.sigmoid() and nn.BCELoss() together
    BCEobj = nn.BCEWithLogitsLoss(pos_weight=torch.tensor([h['obj_pw']], device=device))

    # Class label smoothing https://arxiv.org/pdf/1902.04103.pdf eqn 3
    cp, cn = smooth_BCE(eps=0.0)

    # Focal loss
    g = h['fl_gamma'] # focal loss gamma
    if g > 0:
      BCEcls, BCEobj = FocalLoss(BCEcls, g), FocalLoss(BCEobj, g)
    # Losses
    nt = 0 # number of targets
    no = len(p) # number of outputs Final output of three layers no=3
    balance = [4.0, 1.0, 0.3, 0.1, 0.03] # P3-P7 Confidence loss [80 80] needs to be multiplied by 4.0
    for i, pi in enumerate(p): # layer index, layer predictions, Three levels of measurement, i only 0, 1, 2 Predicted head [80 80] [40 40] [20 20]
      b, a, gj, gi = indices[i] # image, anchor, gridy, gridx
      tobj = torch.zeros_like(pi[..., 0], device=device) # target obj
      n = b.shape[0] # number of targets targets[]
      if n:
        nt += n # cumulative targets
        ps = pi[b, a, gj, gi] # prediction subset corresponding to targets
        # Regression
        pxy = ps[:, :2].sigmoid() * 2. - 0.5
        pwh = (ps[:, 2:4].sigmoid() * 2) ** 2 * anchors[i]
        pbox = torch.cat((pxy, pwh), 1) # predicted box
        iou = bbox_iou(pbox.T, tbox[i], x1y1x2y2=False, CIoU=True) # iou(prediction, target) ## (Compare and contrast the differences between GIOU, CIOU and DIOU and their respective roles)
        lbox += (1.0 - iou).mean() # iou loss, (lbox += (1.0 - iou).mean() # iou loss, since it is a loss, it needs to be negative, or as small as possible)
        # Objectness
        tobj[b, a, gj, gi] = (1.0 - model.gr) + model.gr * iou.detach().clamp(0).type(tobj.dtype) # iou ratio Confidence prediction
        if model.nc > 1: # cls loss (only if multiple classes); ps[:, 5:] Corresponds to three categories, multiple categories are only required for execution
    t = torch.full_like(ps[:, 5:], cn, device=device) # targets

     t[range(n), tcls[i]] = cp

    lcls += BCEcls(ps[:, 5:], t) # BCE

     lobj += BCEobj(pi[..., 4], tobj) * balance[i] # obj loss
      # print("lobj", lobj)
     s = 3 / no # output count scaling
     lbox *= h['box'] * s
     lobj *= h['obj']
     lcls *= h['cls'] * s
     bs = tobj.shape[0] # batch size tobj [batch-size 3 80 80 ] [batch-size 3 40 40 ] [batch-size 3 20 20 ]
     loss = lbox + lobj + lcls
     ## Last needs to be multiplied by bs, i.e. number of batches
     return loss * bs, torch.cat((lbox, lobj, lcls, loss)).detach()
```

Accuracy and recall calculation:

```
# DDP process 0 or single-GPU
if rank in [-1, 0]:
  # mAP
  if ema:
ema.update_attr(model, include=['yaml', 'nc', 'hyp', 'gr', 'names', 'stride',
'class_weights'])
final_epoch = epoch + 1 == epochs
if not opt.notest or final_epoch: # Calculate mAP
  results, maps, times = test.test(opt.data,
                          batch_size=total_batch_size,
imgsz=imgsz_test,
model=ema.ema,
                          single_cls=opt.single_cls,
dataloader=testloader,
save_dir=save_dir, plots=plots and final_epoch,
                          log_imgs=opt.log_imgs if wandb else 0)
if len(stats) and stats[0].any():
  p, r, ap, f1, ap_class = ap_per_class(*stats, plot=plots, save_dir=save_dir,
names=names)
  p, r, ap50, ap = p[:, 0], r[:, 0], ap[:, 0], ap.mean(1) # [P, R, AP@0.5, AP@0.5:0.95]
        ## Calculated what the accuracy and recall were for IOUs from 0.5 to 0.95, and
averaged
          mp, mr, map50, map = p.mean(), r.mean(), ap50.mean(), ap.mean()
  nt = np.bincount(stats[3].astype(np.int64), minlength=nc) # number of targets per
class
else:
        nt = torch.zeros(1)
              # Update best mAP
              fi = fitness(np.array(results).reshape(1, -1)) # weighted combination of [P, R,
mAP@.5, mAP@.5-.95]
              if fi > best_fitness:
                best_fitness = fi
              # Save model
              save = (not opt.nosave) or (final_epoch and not opt.evolve)
              if save:
```

```
with open(results_file, 'r') as f: # create checkpoint
  ckpt = {'epoch': epoch,
      'best_fitness': best_fitness,
      'training_results': f.read(),
      'model': ema.ema,
      'optimizer': None if final_epoch else optimizer.state_dict(),
      'wandb_id': wandb_run.id if wandb else None}
# Save last, best and delete
torch.save(ckpt, last)
if best_fitness == fi:
  torch.save(ckpt, best)
del ckpt
```

Equations：

$$\mathrm{IOU}_{pred}^{truth}$$

Overlap of Real Presence Box Predictions.

$$P_{\gamma}\left(object\right) * \mathrm{IOU}_{pred}^{truth}$$

(Objectives present in the cell) x (Overlap of Real Presence Box Predictions)

$$P_{\gamma}\left(class_i \middle| object\right)$$

Each grid predicts C conditional class probabilities, i.e., the probability that the grid belongs to a class provided it contains an object

$$P_{\gamma}\left(class_i \middle| object\right) * P_{\gamma}\left(object\right) * \mathrm{IOU}_{pred}^{truth} = P_{\gamma}\left(class_i\right) * \mathrm{IOU}_{pred}^{truth}$$

Multiplying the conditional category probability of each raster with the confidence level of each bounding box, the result contains both information about the probability of the predicted category in the bounding box and reflects the accuracy of whether the bounding box contains objects and the coordinates of the bounding box.