



Reinforcement Learning-Based Adaptive Traffic Signal Control with Emergency Vehicle Pre-emption Using YOLOv8 and SUMO

by

Shahriar Khan

Student ID: 2507023

Supervisor: Dr. Seena Joseph

Project submitted as part of the requirements for the award of
MSc Computing – Data Science and Analytics

School of Applied Computing
University of Wales Trinity Saint David

07 September, 2026

Declaration

I declare that this dissertation is my own work. All sources used in this dissertation have been appropriately acknowledged and referenced. This work has not been submitted previously for any other degree or qualification.

Signature: Shahriar Khan

Date: 07 September 2026

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Dr. Seena Joseph, for continuous guidance, constructive feedback, encouragement, and valuable support throughout the duration of this research project. The advice and expertise provided during the planning, implementation, and writing stages of this dissertation were instrumental in the successful completion of this work.

I would also like to thank the School of Applied Computing and the University of Wales Trinity Saint David for providing the academic environment, resources, and facilities required to undertake this research. Access to relevant learning materials, software tools, and research support greatly contributed to the completion of the project.

Special thanks are extended to all academic staff and colleagues whose discussions, suggestions, and shared knowledge helped shape different aspects of this study.

Finally, I am deeply grateful to my family and friends for their patience, encouragement, and unwavering support throughout my academic journey. Their motivation and understanding provided the confidence and determination needed to overcome challenges and successfully complete this dissertation.

Abstract

Traffic congestion remains a significant challenge in urban transportation systems, leading to increased travel times, fuel consumption, environmental pollution, and reduced network efficiency. Conventional fixed-time traffic signal controllers are unable to respond effectively to rapidly changing traffic conditions, often resulting in unnecessary delays and congestion. In addition, emergency vehicles frequently experience interruptions at signalised intersections, affecting response effectiveness.

This research presents an intelligent traffic management framework that integrates YOLOv8-based vehicle detection, Deep Reinforcement Learning (DRL), emergency vehicle prioritisation, and the Simulation of Urban Mobility (SUMO) environment. The perception layer employs YOLOv8 to detect vehicles and generate traffic-state information, while a Deep Q-Network (DQN) controller adaptively adjusts traffic signal timings according to observed traffic conditions. An emergency vehicle prioritisation mechanism was incorporated to provide preferential treatment for ambulances, police vehicles, and fire engines.

The YOLOv8 model achieved a Precision of 42.59%, Recall of 33.05%, and mAP@50 of 34.50% on the testing dataset. Although detection performance indicates scope for further improvement through larger and more balanced datasets, the model produced sufficient traffic information for adaptive signal control. Experimental evaluation demonstrated that the proposed framework reduced average vehicle waiting time from 14.28 seconds under fixed-time control to 2.80 seconds, representing an improvement of 80.40%. Average queue length decreased from 6.14 vehicles to 5.32 vehicles, corresponding to a reduction of 13.36%. Emergency vehicle evaluation recorded an average travel time of 21.05 seconds and an average delay of 11.05 seconds.

The results demonstrate that combining computer vision, reinforcement learning, and emergency vehicle management within a unified framework can improve traffic signal performance and reduce congestion-related delays. The proposed system contributes to the advancement of Intelligent Transportation Systems by linking real-time traffic perception with adaptive signal optimisation and emergency vehicle support.

Contents

Declaration	1
Acknowledgements	2
Abstract	3
1 Introduction	10
1.1 Background	10
1.2 Research Problem	11
1.3 Aim and Objectives	12
1.4 Significance of the Study	13
1.5 Structure of the Report	13
2 Review of Literature	15
2.1 Traffic Signal Control Methods	15
2.1.1 Fixed-Time Traffic Signal Control	16
2.1.2 Sensor-Based Traffic Signal Control Methods	16
2.1.3 Reinforcement Learning-Based Traffic Signal Control Methods	18
2.2 Q-Learning and Deep Q-Network Approaches	20
2.2.1 Q-Learning	20
2.2.2 Deep Q-Network (DQN)	21
2.3 Justification for Selecting DQN	22
2.4 YOLO-Based Vehicle Detection and Computer Vision in Traffic Management	22
2.5 Emergency Vehicle Detection and Prioritisation Methods	24
2.6 SUMO Simulation Environment in Intelligent Transportation Research	25
2.7 Comparative Analysis of Existing Studies	25
2.8 Chapter Summary	28
3 Research Methodology	29
3.1 Research Philosophy	29
3.2 Research Approach	30
3.3 Research Strategy	31

3.4	Research Design	32
3.5	Data Collection Methods	35
3.5.1	Traffic Demand Data	36
3.6	Data Analysis	37
3.6.1	Vehicle Detection Analysis	37
3.6.2	Traffic Management Analysis	38
3.6.3	Comparative Analysis	38
3.7	Achievement of Research Objectives	39
3.8	Evaluation Framework	40
3.9	Chapter Summary	41
4	Design Process	42
4.1	Overall Architecture	42
4.2	Traffic Dataset Design	44
4.3	Vehicle Detection System Design	45
4.3.1	Overall YOLOv8 Architecture	45
4.3.2	Vehicle Detection Pipeline	46
4.4	Reinforcement Learning Controller Design	47
4.4.1	State Representation	48
4.4.2	Action Space	48
4.4.3	Reward Function Design	48
4.4.4	Deep Q-Network Architecture	49
4.4.5	DQN Learning Algorithm	51
4.5	Emergency Vehicle Prioritisation Design	52
4.5.1	Emergency Vehicle Detection	52
4.5.2	Direction Classification	52
4.5.3	Signal Pre-emption Strategy	53
4.5.4	Recovery Mechanism	53
4.6	Evaluation Design	54
5	Implementation	56
5.1	Dataset Annotation Tool Implementation	56
5.2	YOLOv8 Training Environment	57
5.3	Traffic State Acquisition	59
5.4	DQN Traffic Signal Controller	59
5.5	Reward Implementation	60
5.6	Emergency Vehicle Priority Implementation	60
5.7	SUMO–TraCI Integration	61
5.7.1	SUMO Simulation Configuration	62

5.8	Implementation Challenges	63
5.9	Chapter Summary	64
6	Evaluation of Results	65
6.1	Vehicle Detection Performance	65
6.2	Visual Evaluation of Detection Results	67
6.3	Reinforcement Learning Behaviour	71
6.4	Queue Length Analysis	72
6.5	Comparative Evaluation	74
6.6	Emergency Vehicle Evaluation	74
6.7	Achievement of Research Objectives	75
6.8	Chapter Summary	75
7	Conclusions and Recommendations	77
7.1	Research Summary	77
7.2	Key Findings	78
7.3	Research Contributions	78
7.4	Limitations	79
7.5	Recommendations for Future Work	80
7.6	Final Conclusion	80
8	Reflection	82
8.1	Project Management Reflection	82
8.2	Technical Skills Development	82
8.3	Personal Development	83
8.4	Reflection Summary	83
A	Source Code Samples	92
A.1	YOLOv8 Detection Implementation	92
A.2	Traffic State Extraction	92
A.3	DQN Action Selection	92
A.4	Emergency Vehicle Prioritisation	93

List of Figures

2.1	Conceptual architecture of a Q-Learning-based traffic signal controller. The agent interacts with the SUMO–TraCI environment through state observations and signal-control actions. Adapted from Ouyang et al. [1].	21
3.1	Research workflow adopted for the development and evaluation of the proposed intelligent traffic management framework.	35
4.1	Overall architecture of the proposed intelligent traffic management framework. The framework integrates YOLOv8-based traffic perception, traffic-state generation, Deep Q-Network traffic signal control, emergency vehicle prioritisation, SUMO–TraCI simulation, and performance evaluation within a unified system.	43
4.2	YOLOv8-based traffic perception architecture and traffic-state generation process used within the proposed framework.	46
4.3	Vehicle detection pipeline used to transform traffic images into traffic-state information for adaptive traffic signal control.	47
4.4	Deep Q-Network architecture used for adaptive traffic signal control. . . .	50
4.5	Conceptual architecture of the Deep Q-Network controller used for adaptive traffic signal control. Adapted from Ouyang et al. [1].	51
4.6	Emergency vehicle prioritisation workflow.	53
4.7	Emergency vehicle prioritisation process adopted in the proposed framework.	54
5.1	Custom image annotation application	57
5.2	Example annotated training image used during YOLOv8 model development.	58
5.3	Ground-truth validation annotations.	58
5.4	YOLOv8 prediction output on validation data.	58
5.5	Simplified DQN decision-making cycle used for traffic signal control. . . .	60
5.6	Emergency vehicle priority procedure	61
5.7	Communication architecture between SUMO, TraCI, and the external Python controller.	62

5.8	SUMO simulation configuration workflow showing the relationship between the road network file, route definitions, additional configuration files, Python controller, and SUMO configuration file.	63
6.1	Distribution of labelled vehicle instances across training categories. . . .	66
6.2	YOLOv8 training performance over 30 epochs.	67
6.3	Ground-truth annotations	68
6.4	YOLOv8 predictions	68
6.5	Comparison between validation annotations and YOLOv8 predictions for Validation Set 1.	68
6.6	Ground-truth annotations	69
6.7	YOLOv8 predictions	69
6.8	Comparison between validation annotations and YOLOv8 predictions for Validation Set 2.	69
6.9	YOLOv8 confusion matrix.	70
6.10	Precision–Recall curve for YOLOv8.	71
6.11	DQN reward behaviour during simulation.	72
6.12	Queue length variation under DQN traffic signal control.	73
6.13	Queue length variation under fixed-time traffic signal control.	73

List of Tables

2.1	Comparison of Existing Traffic Management Studies	27
3.1	Research Design and Methodology Overview	33
3.2	Summary of Data Sources Used in the Research	37
3.3	Relationship Between Research Objectives and Evaluation Metrics	40
3.4	Evaluation Framework	41
4.1	Design Components of the Proposed Framework	44
5.1	Vehicle Classes Used During Annotation for normal vehicles	57
5.2	Traffic-State Variables	59
5.3	Emergency Vehicle Priority Operations	61
5.4	Implementation Challenges and Solutions	64
6.1	YOLOv8 Vehicle Detection Performance	65
6.2	Comparison Between Fixed-Time Control and the Proposed DQN Frame- work	74
6.3	Emergency Vehicle Performance	74

Chapter 1

Introduction

1.1 Background

Traffic congestion has become one of the most significant challenges facing urban transportation systems across the world. Increasing urbanisation, population growth, and vehicle ownership have resulted in increased pressure on existing road infrastructure, leading to longer travel times, fuel wastage, environmental pollution, and reduced transportation efficiency [2], [3], [4], [5] .

Traffic signal control plays a vital role in managing urban traffic movement. Traditionally, most traffic intersections operate using fixed-time traffic signal control systems, where signal timings are predetermined based on historical traffic data [6] . While these systems are relatively simple and inexpensive to implement, they are unable to adapt to rapidly changing traffic conditions. Consequently, vehicles frequently experience unnecessary delays and congestion, particularly during peak traffic periods and unexpected traffic events [7], [8].

In response to these challenges, Intelligent Transportation Systems (ITS) have emerged as a promising solution for enhancing traffic management. Intelligent transportation systems integrate communication technologies, real-time monitoring, artificial intelligence, and data analytics to improve transportation efficiency [9], [10], [11], [12]. Among the various approaches proposed within ITS, Reinforcement Learning (RL) has gained considerable attention because of its ability to learn effective traffic signal control policies through interaction with the traffic environment [13]. At the same time, recent developments in computer vision have transformed traffic monitoring capabilities. Object detection algorithms such as YOLO (You Only Look Once) can analyse video streams and detect vehicles in real time with high accuracy and computational efficiency [14], [15], [16], [17], [18], [19], [20], [21], [22], [23].

Many existing studies focus either on traffic perception or traffic signal control independently. Furthermore, emergency vehicle prioritisation is frequently treated as a

separate problem despite its significant impact on public safety and transportation efficiency. Consequently, there is growing interest in developing integrated traffic management frameworks that combine real-time traffic perception, intelligent decision-making, and emergency vehicle prioritisation. This research proposes a unified framework that integrates YOLO-based vehicle detection, Reinforcement Learning-based traffic signal optimisation, and emergency vehicle pre-emption using the SUMO simulation environment.

1.2 Research Problem

Traffic congestion remains one of the most persistent challenges facing modern urban transportation systems. Increasing vehicle ownership, population growth, and urban expansion have placed considerable pressure on existing road infrastructure, resulting in longer travel times, increased fuel consumption, air pollution, and reduced transportation efficiency. Although numerous traffic management solutions have been developed over the last few decades, congestion at signalised intersections continues to negatively affect both road users and emergency response services [13], [24].

Conventional traffic signal control systems primarily rely on fixed-time schedules or sensor-based adaptive approaches. While these methods provide a basic level of traffic management, they often struggle to respond effectively to rapidly changing traffic conditions. Fixed-time systems operate using predefined signal timings based on historical traffic patterns, making them unsuitable for dynamic urban environments where traffic demand fluctuates throughout the day [24], [25], [26], [27], [28], [29]. Consequently, vehicles frequently experience unnecessary delays, increased queue lengths, and inefficient utilisation of available road capacity.

Recent advances in Reinforcement Learning (RL) have demonstrated significant potential for improving traffic signal optimisation. RL-based traffic controllers can adapt signal timings dynamically according to changing traffic conditions and have shown promising results in reducing vehicle delays and improving throughput. However, many existing RL-based systems depend heavily on simulation-generated traffic states rather than real-world perception data. As a result, the traffic information available to the learning agent may not accurately represent real urban traffic environments, limiting the practicality and scalability of these approaches in real-world deployments [13], [30].

Similarly, computer vision techniques, particularly YOLO-based object detection models, have achieved high levels of accuracy in vehicle detection, traffic monitoring, congestion analysis, and emergency vehicle recognition [15], [31]. Despite these advances, many YOLO-based traffic management systems are primarily used for traffic observation and data collection purposes. The outputs generated by these detection models are often not directly integrated into intelligent traffic signal decision-making processes, resulting in a disconnect between traffic perception and traffic control [31], [32].

Another important challenge concerns emergency vehicle prioritisation. Existing emergency vehicle management systems commonly utilise GPS devices, RFID technology, wireless communication networks, or predefined signal override mechanisms to grant priority to ambulances, fire engines, and police vehicles [24], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42]. Although these approaches can improve emergency response times, they frequently operate independently of overall traffic optimisation strategies. Consequently, emergency vehicle priority mechanisms may create additional congestion for surrounding traffic or fail to adapt effectively to complex traffic conditions [33], [31].

A review of current literature reveals three significant research gaps. First, many reinforcement learning-based traffic control systems lack integration with real-time traffic perception technologies. Second, YOLO-based vehicle detection frameworks are often used solely for traffic monitoring and are not directly incorporated into adaptive signal optimisation systems. Third, emergency vehicle prioritisation is commonly implemented as a standalone function rather than as an integrated component of intelligent traffic management frameworks [13], [33], [31].

Therefore, there is a need for an intelligent traffic management framework that combines real-time vehicle detection, adaptive traffic signal optimisation, and emergency vehicle prioritisation within a unified system. This research addresses this gap by integrating YOLO-based vehicle detection, Deep Reinforcement Learning, and emergency vehicle prioritisation using the SUMO simulation environment. The proposed framework aims to improve traffic flow efficiency, reduce congestion, minimise vehicle waiting times, and provide priority access for emergency vehicles while maintaining overall traffic performance.

1.3 Aim and Objectives

Aim

To develop and evaluate an intelligent traffic signal control system that integrates YOLO-based vehicle detection, Deep Reinforcement Learning, and emergency vehicle prioritisation for improving urban traffic efficiency.

Objectives

1. To critically review existing literature on adaptive traffic signal control, reinforcement learning, computer vision-based vehicle detection, and emergency vehicle prioritisation to identify current trends and research gaps.
2. To develop and simulate an intelligent traffic signal control framework using Deep Reinforcement Learning and YOLO-based emergency vehicle detection within the

SUMO environment to optimise traffic flow and prioritise emergency vehicles.

3. To evaluate the effectiveness of the proposed framework using performance metrics including waiting time, queue length, traffic throughput, and emergency vehicle delay, and compare the results with conventional fixed-time traffic signal control systems.

1.4 Significance of the Study

This research contributes to the field of Intelligent Transportation Systems by integrating computer vision, reinforcement learning, and emergency vehicle prioritisation within a single framework.

The proposed system aims to reduce waiting time, minimise congestion, improve traffic throughput, and reduce emergency vehicle delays. The findings may support the future development of smart city traffic management solutions.

1.5 Structure of the Report

Chapter 2: Review of Literature

This chapter critically examines existing research relating to Intelligent Transportation Systems (ITS), adaptive traffic signal control, reinforcement learning, computer vision, YOLO-based vehicle detection, traffic simulation, and emergency vehicle prioritisation. The review identifies current challenges, evaluates existing solutions, and highlights the research gaps addressed by the proposed framework.

Chapter 3: Research Methodology

This chapter presents the methodological framework adopted for the study. It discusses the research philosophy, research approach, experimental strategy, data collection procedures, validation methods, and evaluation techniques used to investigate the proposed intelligent traffic management framework.

Chapter 4: Design Process

This chapter describes the design decisions that guided the development of the system. It includes the overall framework architecture, dataset design, YOLOv8 detection architecture, reinforcement learning controller design, emergency vehicle prioritisation strategy, and experimental environment design.

Chapter 5: Implementation

This chapter explains the practical implementation of the proposed framework. It presents the development of the dataset annotation tool, YOLOv8 training environment, traffic-state acquisition module, Deep Q-Network controller, emergency vehicle priority mechanism, and SUMO–TraCI integration.

Chapter 6: Evaluation of Results

This chapter evaluates the performance of the implemented system using quantitative experiments. Vehicle detection accuracy, reinforcement learning behaviour, traffic congestion metrics, emergency vehicle prioritisation performance, and comparisons with conventional traffic signal control approaches are analysed and discussed.

Chapter 7: Conclusions and Recommendations

This chapter summarises the research findings and evaluates the extent to which the project objectives were achieved. It highlights the key contributions of the study, discusses limitations, and provides recommendations for future research and practical development.

Chapter 8: Reflection

This chapter provides a reflective discussion of the project development process. It examines project management, technical skills acquired, research skills developed, challenges encountered, lessons learned, and personal professional development resulting from the completion of the research.

Chapter 2

Review of Literature

2.1 Traffic Signal Control Methods

These systems were primarily developed using traffic engineering principles and mathematical optimisation techniques to minimise vehicle conflicts, improve intersection efficiency, and maintain orderly traffic movement [43], [7], [44]. Early traffic control strategies relied on predefined timing plans derived from historical traffic observations, assuming relatively stable traffic demand patterns over time. As a result, traditional TSC methods focused on controlling signal phases, cycle lengths, and green-time allocations to reduce delays and maximise intersection capacity [43], [45].

Conventional traffic management approaches are generally classified into two categories: fixed-time traffic control systems and traffic-responsive control systems. Fixed-time systems operate using predetermined signal timings that remain unchanged regardless of current traffic conditions. Although these systems are simple to implement and require minimal computational resources, they often perform poorly under fluctuating traffic demand, leading to increased vehicle delays and queue formation during peak periods [43], [46], [44]. To overcome these limitations, traffic-responsive methods were introduced, utilising sensor data such as vehicle counts, occupancy rates, and queue lengths to adjust signal timings dynamically. Examples include Webster-based optimisation, Green-Wave coordination, SCOOT, SCATS, and MaxPressure control, which aim to improve traffic progression and network throughput by adapting signal operations according to observed traffic conditions [7],[47],[48].

Despite their widespread adoption, traditional traffic signal control methods remain constrained by their dependence on predefined traffic models, handcrafted control logic, and limited adaptability to highly dynamic urban environments. These challenges have motivated the development of intelligent and data-driven traffic management approaches,

particularly Reinforcement Learning (RL) and Deep Reinforcement Learning (DRL), which are capable of learning adaptive signal control policies directly from traffic interactions [49],[50],[51],[52].

2.1.1 Fixed-Time Traffic Signal Control

Traditional Traffic Signal Control (TSC) approaches have long been used to manage traffic flow at signalised intersections through predetermined control strategies and traffic engineering models. Among these, Fixed-Time Control (FTC) remains one of the most widely deployed methods due to its simplicity, low implementation cost, and operational reliability. In FTC systems, signal phases and cycle lengths are calculated using historical traffic observations and remain unchanged during operation [43] [45]. Although this approach performs adequately under stable traffic conditions, its inability to respond to real-time traffic fluctuations often results in inefficient green-time allocation, increased queue lengths, and higher vehicle delays during peak-hour demand or unexpected traffic disruptions [46] [44],

Furthermore, fixed-time controllers lack the flexibility required for emergency vehicle prioritisation and adaptive traffic management, limiting their effectiveness in modern Intelligent Transportation Systems (ITS) [45]. To address these limitations, sensor-based traffic signal control systems were introduced to incorporate real-time traffic information into signal timing decisions. These systems utilise sensing technologies such as inductive loop detectors, Radio Frequency Identification (RFID), wireless sensor networks (WSNs), cameras, and roadside communication devices to monitor traffic conditions continuously and adjust signal operations accordingly [24], [46].

2.1.2 Sensor-Based Traffic Signal Control Methods

To overcome the limitations associated with fixed-time control systems, researchers introduced sensor-based traffic management approaches capable of monitoring real-time traffic conditions and adjusting signal timings accordingly [24]. These approaches utilise various sensing technologies to collect traffic information and support more responsive traffic control decisions.

Inductive Loop Detectors

Inductive loop detectors are among the earliest and most widely adopted traffic-monitoring technologies, providing reliable measurements of vehicle presence, traffic volume, and lane occupancy. However, their deployment requires road excavation and ongoing maintenance, resulting in significant infrastructure costs [24],[44].

Radio Frequency Identification (RFID)

RFID-based traffic management systems have also been explored, particularly for vehicle identification and emergency vehicle prioritisation. By enabling communication between RFID readers and tagged vehicles, these systems can facilitate faster identification of authorised vehicles and support priority-based signal control. Nevertheless, their effectiveness depends on the availability of specialised tags and extensive supporting infrastructure, which can restrict large-scale deployment [45].

Wireless Sensor Networks (WSNs)

Nellore and Hancke [24] reported that WSNs offer improved flexibility, easier deployment, and lower infrastructure requirements when compared with traditional loop detector systems. These characteristics make WSNs attractive for modern traffic monitoring applications. Despite enhancing traffic responsiveness, sensor-based control systems still exhibit several limitations. Their performance depends heavily on sensor accuracy, communication reliability, and infrastructure availability. In addition, conventional sensors primarily provide numerical traffic measurements and often lack the contextual information required to distinguish vehicle categories, analyse complex traffic behaviour, or identify emergency vehicles without additional technologies [46], [53].

Challenges of Sensor-Based Systems

Many sensing technologies require substantial infrastructure investment and ongoing maintenance. Hardware failures, environmental conditions, and communication issues may affect data quality and system reliability. Furthermore, traditional sensing systems typically provide limited contextual information and often cannot accurately distinguish between different vehicle categories, including emergency vehicles, without additional technologies [24]. As a result, their capability to support intelligent traffic management remains constrained.

Overall, traditional traffic signal control systems provide a stable foundation for traffic management but exhibit limited adaptability and perception capabilities. Consequently, while sensor-based systems improve upon fixed-time control, they remain constrained in highly dynamic urban environments where richer traffic perception and intelligent decision-making are required. These limitations have motivated increasing interest in Artificial Intelligence (AI), computer vision, and Reinforcement Learning (RL)-based traffic management approaches capable of learning traffic patterns automatically and adapting signal control policies in real time [50],[51],[52],[48].

2.1.3 Reinforcement Learning-Based Traffic Signal Control Methods

By combining reinforcement learning with deep neural networks, DRL methods can process complex traffic state representations and generate more sophisticated traffic control policies. Deep Q-Networks (DQN) have demonstrated strong performance in learning adaptive signal timing strategies that reduce congestion and improve traffic efficiency in simulated urban environments [47]. Similarly, Proximal Policy Optimisation (PPO) has gained attention due to its stable learning behaviour and ability to handle large-scale traffic control problems [54].

Recent studies have demonstrated the effectiveness of Deep Reinforcement Learning (DRL) for adaptive traffic signal control. For example, IntelliLight introduced a reinforcement learning framework capable of reducing vehicle waiting times and improving intersection efficiency through dynamic signal optimisation [50]. Although these results highlight the potential of DRL for congestion management, the framework relies on simulator-generated traffic states and assumes complete traffic observability, which may limit its applicability in real-world traffic environments where sensing uncertainties and perception errors are common [50, 44]. PressLight further improved traffic flow by incorporating the max-pressure control principle into deep reinforcement learning, enabling more effective coordination between signalised intersections [51]. Reinforcement learning-based traffic controllers within the SUMO simulation environment achieved better adaptability and overall traffic performance than conventional traffic control methods [30]. Despite these advances, several challenges remain. A significant limitation of many reinforcement learning-based traffic management systems is their reliance on simulator-generated traffic states. In most cases, traffic information is obtained directly from the simulation environment rather than from real-world sensing technologies, meaning that challenges associated with visual perception, sensor uncertainty, and incomplete observations are not adequately represented during controller training and evaluation [13] [30].

The integration of deep neural networks has further expanded these capabilities, leading to the development of Deep Reinforcement Learning (DRL) frameworks capable of modelling complex traffic dynamics and nonlinear decision relationships. Studies based on DQN, Double DQN, Dueling DQN, D3QN, DDPG, SAC, and Actor-Critic architectures have consistently demonstrated improvements in traffic throughput, delay reduction, queue minimisation, and waiting-time optimisation when compared with conventional traffic control approaches [50],[55],[51],[47],[52],[48]. Despite these advantages, RL-based traffic signal control systems face several significant challenges. One of the most frequently reported limitations is the high-dimensional state representation used to

describe traffic conditions.

Existing studies commonly employ vehicle counts, queue lengths, waiting times, occupancy rates, vehicle positions, lane densities, and signal phase information as state variables [50],[55],[47]. While detailed state representations can improve environmental awareness and learning performance, they also increase computational complexity, memory consumption, and training time. Moreover, increasing state dimensionality may introduce redundancy and noise, negatively affecting convergence speed and decision stability. Recent studies have therefore emphasised the need for compact yet informative traffic representations that balance learning efficiency and control performance [53],[48]. Another major challenge relates to scalability. While single-intersection control has produced encouraging results, real-world transportation systems consist of numerous interconnected intersections whose decisions influence one another. Consequently, researchers have proposed centralised learning frameworks and Multi-Agent Reinforcement Learning (MARL) approaches to coordinate neighbouring intersections [55],[51],[52].

Centralised architectures use a global critic or controller with visibility of the entire traffic network, whereas MARL distributes decision-making among local signal controllers. Cooperative learning frameworks such as IntelliLight, CoLight, Friend-DQN, and related MARL architectures have demonstrated superior scalability and coordination performance compared with isolated RL agents [50],[51],[52],[46]. However, they also introduce communication overhead, policy coordination complexity, and overgeneralisation issues that may reduce performance under heterogeneous traffic conditions [55],[48]. As a result, balancing local intersection optimisation with global network efficiency remains an active research challenge. The effectiveness of RL-based traffic signal control is also strongly influenced by reward design, state formulation, and action selection mechanisms. Existing reward functions are commonly formulated using weighted combinations of travel time, throughput, queue length, waiting time, delay, occupancy, and traffic pressure [50][47],[55]. Among these metrics, pressure-based rewards have received considerable attention because they directly measure traffic imbalances between incoming and outgoing approaches and have demonstrated strong performance in large-scale urban networks [51],[55].

Similarly, state definitions range from aggregated traffic statistics to detailed vehicle-level observations. However, there remains no universally accepted reward or state design, indicating that optimal formulations are highly dependent on network characteristics and traffic environments [53],[48],[44]. Recent research has further extended RL-based traffic signal control to specialised transportation scenarios, including emergency vehicle prioritisation, adaptive corridor management, uncertainty-aware decision-making,

and simulation-to-real transfer [45],[53],[48]. These studies demonstrate that hybrid approaches combining reinforcement learning, traffic-flow theory, signal pre-emption, and predictive traffic modelling can significantly improve traffic efficiency and emergency response performance. Nevertheless, their effectiveness often depends on accurate estimation of vehicle arrivals, traffic density, queue lengths, and congestion levels, highlighting an increasing dependence on reliable traffic sensing and perception technologies [45],[53],[48]. A recurring limitation across traditional, adaptive, and reinforcement learning-based traffic signal control studies is the assumption of complete and accurate traffic-state information. Most existing frameworks rely on simulation environments such as SUMO and directly obtain variables including queue lengths, waiting times, occupancy rates, vehicle locations, and traffic densities through virtual detectors and communication interfaces [46],[53].

The DQN architecture typically consists of an input layer representing the traffic state, one or more hidden layers responsible for feature extraction and function approximation, and an output layer representing the available traffic signal actions. The network estimates Q-values for each action and enables the controller to generalise across previously unseen traffic states. This ability makes DQN particularly suitable for traffic management problems characterised by large and continuously changing state spaces [1].

2.2 Q-Learning and Deep Q-Network Approaches

Reinforcement Learning (RL) has emerged as one of the most widely investigated approaches for adaptive traffic signal control because it enables an agent to learn traffic-management policies through interaction with the environment. Among RL methods, Q-Learning and Deep Q-Networks (DQN) are frequently applied to traffic-signal optimisation problems due to their ability to improve decision-making under dynamic traffic conditions [1].

2.2.1 Q-Learning

Q-Learning is a model-free reinforcement learning algorithm that estimates the quality of a state-action pair through a Q-value function. During training, the agent interacts with the environment, receives rewards, and updates Q-values using the Bellman equation:

$$Q(s, a) \leftarrow Q(s, a) + \alpha [R + \gamma \max_{a'} Q(s', a') - Q(s, a)] \quad (2.1)$$

where $Q(s, a)$ represents the value of taking action a in state s , α is the learning rate, γ is the discount factor, and R denotes the immediate reward. In traffic signal control

applications, Q-Learning agents typically use traffic conditions such as queue lengths, waiting times, occupancy rates, or traffic density as state representations. The agent then learns to select traffic-signal actions that maximise long-term rewards by reducing congestion and improving intersection efficiency [1]. However, the effectiveness of Q-Learning decreases significantly as the size of the state space increases because all state-action combinations must be stored and updated within a Q-table.

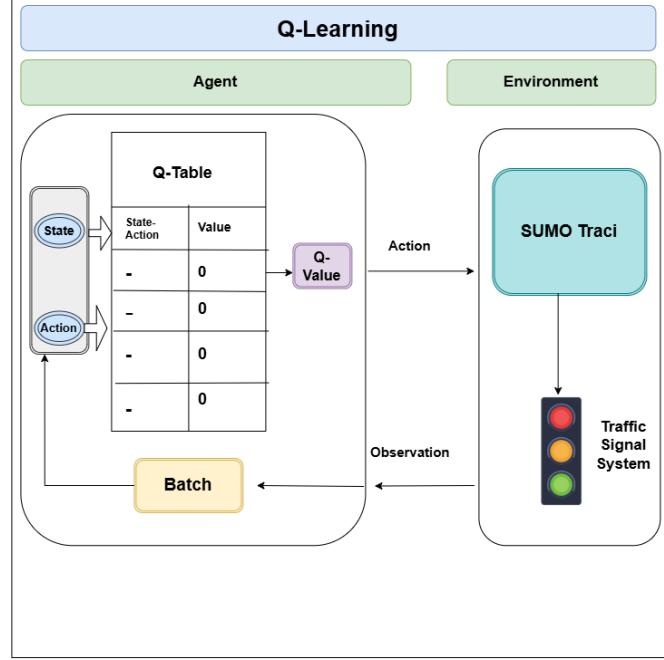


Figure 2.1: Conceptual architecture of a Q-Learning-based traffic signal controller. The agent interacts with the SUMO–TraCI environment through state observations and signal-control actions. Adapted from Ouyang et al. [1].

2.2.2 Deep Q-Network (DQN)

Deep Q-Networks extend conventional Q-Learning by replacing the Q-table with a deep neural network that approximates the Q-value function. Instead of storing values for every possible state-action pair, the network receives a traffic-state representation as input and predicts the expected reward associated with each available action [1]. The DQN model typically consists of an input layer, one or more hidden layers, and an output layer representing available actions. During training, the network learns the relationship between observed traffic conditions and optimal signal-control decisions. This approach enables DQN to operate effectively in high-dimensional environments where traditional Q-Learning becomes impractical. Because traffic management problems often involve large state spaces and continuously changing traffic conditions, DQN controllers have demonstrated improved scalability and adaptability compared with conventional Q-Learning approaches. Ouyang et al. reported that DQN-based traffic control achieved higher av-

erage vehicle speeds and lower lane occupancy rates than Q-Learning in SUMO-based simulations, indicating improved traffic-management performance under dynamic traffic conditions [1].

2.3 Justification for Selecting DQN

The proposed intelligent traffic management framework utilises a Deep Q-Network (DQN) rather than traditional Q-Learning because the traffic-control problem considered in this study involves a multi-dimensional state representation containing lane-specific queue lengths and signal phase information. Using a conventional Q-table would require storing and updating a large number of state-action combinations, resulting in increased memory requirements and reduced scalability [56], [57].

By employing a neural network to approximate Q-values, the DQN controller can generalise across unseen traffic states and learn more complex traffic-management strategies. This capability enables DQN to operate effectively in high-dimensional environments where traditional tabular reinforcement learning methods become impractical [57], [58]. Furthermore, previous research has demonstrated that DQN-based traffic controllers outperform traditional Q-Learning approaches in dynamic traffic environments and achieve superior congestion-reduction performance. Ouyang et al. reported that DQN achieved higher average vehicle speeds and lower lane occupancy rates than Q-Learning within SUMO-based traffic simulations [1].

Similar findings have been reported in reinforcement learning traffic signal control studies, where deep learning-based approaches achieved improved waiting times, queue lengths, and overall traffic-flow efficiency when compared with conventional reinforcement learning methods [47], [44]. Consequently, DQN was selected as the most appropriate reinforcement learning architecture for the proposed framework because of its scalability, ability to generalise across previously unseen traffic states, and suitability for complex traffic management problems involving large state spaces and dynamic traffic conditions.

2.4 YOLO-Based Vehicle Detection and Computer Vision in Traffic Management

YOLO (You Only Look Once) has become one of the most widely adopted object-detection frameworks because it performs object localisation and classification within a single network architecture, enabling real-time detection with relatively high accuracy and computational efficiency [15], [59]. Unlike conventional two-stage detectors, YOLO

processes the entire image in a single forward pass, making it particularly suitable for traffic-management applications that require rapid decision-making and continuous monitoring.

Early YOLO-based studies focused primarily on vehicle detection, classification, and counting, allowing traffic authorities to estimate traffic density and congestion more effectively than traditional sensor-based approaches [15], [60]. As the architecture evolved, researchers extended its application to traffic-flow analysis, queue monitoring, violation detection, and real-time road surveillance [59]. These developments demonstrate that camera-based perception systems can provide richer contextual information than conventional sensing technologies by simultaneously capturing vehicle presence, category, and movement characteristics. Recent studies have explored the integration of YOLOv8 within intelligent transportation systems. YOLOv8 has demonstrated the ability to detect multiple vehicle categories under complex urban traffic conditions while maintaining real-time performance [59], [32]. By extracting vehicle counts, lane occupancy information, and vehicle-class data, YOLOv8 can generate traffic-state information suitable for adaptive control systems. Furthermore, several studies have utilised YOLO-based detection to identify emergency vehicles, enabling traffic controllers to respond dynamically to priority traffic movements [31], [45].

Despite these advantages, many YOLO-based traffic-management frameworks remain focused on perception and monitoring rather than direct traffic optimisation. In most cases, detection outputs are used primarily for traffic analysis, while signal-control decisions continue to rely on predefined rules or conventional control strategies [31], [32]. In addition, detection accuracy may be affected by environmental factors such as illumination changes, weather conditions, camera-angle variations, and object occlusion [60], [59]. Consequently, a gap remains between traffic perception and intelligent traffic control. The YOLO family has evolved significantly since the introduction of the original architecture by Redmon et al. [15]. Subsequent versions improved localisation accuracy and computational efficiency through architectural refinements. YOLOv4 introduced Cross Stage Partial (CSP) connections and several optimisation strategies that improved feature extraction and localisation performance [60]. More recently, YOLOv8 introduced anchor-free detection and C2f modules, providing a balance between computational efficiency and object-detection accuracy [59]. These characteristics make YOLOv8 particularly suitable for integration with reinforcement learning-based traffic management systems.

Another important consideration is dataset composition. Several studies have reported that object-detection performance is strongly influenced by dataset size, class balance, and annotation quality. Underrepresented vehicle categories frequently achieve

lower detection accuracy because fewer training examples are available during model optimisation [58, 59]. Consequently, dataset design remains a critical factor influencing the robustness and generalisation capability of YOLO-based traffic perception systems.

2.5 Emergency Vehicle Detection and Prioritisation Methods

Emergency vehicle (EV) prioritisation is an important area of Intelligent Transportation Systems (ITS) research because delays experienced by ambulances, fire engines, and police vehicles can directly affect emergency response performance and public safety [45], [44]. Signalised intersections often become major bottlenecks for emergency vehicles, particularly within congested urban environments, motivating the development of priority-based traffic management strategies [45].

Early emergency vehicle prioritisation systems primarily relied on technologies such as GPS tracking, RFID tags, wireless communication networks, and Vehicle-to-Infrastructure (V2I) communication to identify approaching emergency vehicles and trigger signal pre-emption mechanisms [24], [45].

Although these approaches improved emergency vehicle mobility, their effectiveness depended on dedicated infrastructure and reliable communication systems, increasing implementation and maintenance costs [24], [45]. Recent advances in computer vision have provided alternative solutions based on camera systems and deep learning. In particular, YOLO-based object detection models have demonstrated strong performance in recognising emergency vehicles in real time while maintaining computational efficiency [15], [59]. Several studies have shown that integrating YOLOv8 with traffic management systems enables dynamic signal adjustment and improves emergency vehicle progression through signalised intersections [31], [32]. Compared with conventional sensing technologies, computer-vision-based approaches provide richer traffic information, including vehicle classification, trajectory estimation, traffic density, and emergency vehicle identification from a single camera source [24], [46].

To further improve emergency vehicle handling, reinforcement learning has increasingly been incorporated into traffic signal control. Frameworks such as EMVLight utilise Multi-Agent Reinforcement Learning (MARL) to coordinate emergency vehicle prioritisation while maintaining acceptable traffic conditions for ordinary road users [33], [52]. However, many existing studies continue to depend on simulator-generated traffic states and assume complete traffic observability, limiting their applicability in real-world trans-

portation environments [33], [48].

2.6 SUMO Simulation Environment in Intelligent Transportation Research

Simulation platforms play an important role in intelligent transportation research because they enable traffic management strategies to be evaluated in a safe and controlled environment before real-world deployment. Among available traffic simulators, the *Simulation of Urban Mobility* (SUMO) platform has become one of the most widely adopted tools due to its open-source nature, scalability, and ability to model traffic behaviour at the microscopic level [61], [62], [63], [64]. Unlike macroscopic models that analyse aggregated traffic patterns, SUMO represents individual vehicle movements, allowing detailed simulation of lane changing, route selection, acceleration, and intersection interactions [61], [62].

A major advantage of SUMO is its integration with the Traffic Control Interface (TraCI), which enables real-time communication between the simulation environment and external applications. Through TraCI, traffic controllers can retrieve information such as queue lengths, waiting times, vehicle counts, and signal states while simultaneously modifying traffic signal operations during simulation execution [65], [46]. This capability has made SUMO particularly attractive for reinforcement learning research, where continuous interaction between the learning agent and the environment is required [46], [48].

Recent studies have successfully used SUMO to evaluate adaptive traffic signal control algorithms. Reinforcement learning frameworks integrated with SUMO have demonstrated improvements in congestion management, waiting-time reduction, and traffic-flow efficiency when compared with conventional fixed-time strategies [50], [47]. Similarly, multi-agent approaches have shown that coordination between neighbouring intersections can improve network-wide traffic performance while maintaining scalability in larger urban networks [52], [46].

2.7 Comparative Analysis of Existing Studies

Recent research has increasingly focused on Reinforcement Learning (RL) and Deep Reinforcement Learning (DRL) for adaptive traffic signal control. Frameworks such as In-

telliLight and PressLight demonstrated that learning-based controllers can reduce waiting times, minimise queue lengths, and improve traffic flow through dynamic signal optimisation [50], [51]. Multi-agent approaches have further improved scalability and coordination between neighbouring intersections [55], [52]. However, many RL-based systems continue to rely on simulator-generated traffic states and assume complete traffic observability, limiting their practical applicability [46], [48].

Table 2.1: Comparison of Existing Traffic Management Studies

Study	Technology	RL	YOLO	EV	SUMO	Strengths	Limitations
IntelliLight [50]	DQN	✓	×	×	✓	Reduces waiting time and improves signal efficiency through adaptive traffic signal control.	No emergency vehicle support and no computer-vision-based traffic perception.
PressLight [51]	Deep RL	✓	×	×	✓	Optimises traffic flow at signalised intersections using pressure-based rewards.	Relies on simulator-generated traffic states and lacks real-time perception.
EMVLight [33]	Multi-Agent RL	✓	×	✓	✓	Supports emergency vehicle prioritisation while maintaining overall traffic performance.	Does not include computer vision for vehicle detection.
YOLOv8-Based EV Detection [31]	YOLOv8	×	✓	✓	×	Provides accurate real-time vehicle and emergency vehicle detection.	No adaptive signal control or reinforcement learning component.
YOLOv8 + RL Framework [32]	YOLOv8 + RL	✓	✓	×	✓	Combines traffic perception with intelligent signal control.	Limited emergency vehicle management and small-scale evaluation.
SUMO-RL [30]	RL + SUMO	✓	×	×	✓	Widely used benchmark framework for adaptive traffic signal optimisation.	Traffic information is obtained directly from the simulator without vision-based perception.
Emergency Priority System [45]	GPS + Pre-emption	×	×	✓	×	Reduces emergency vehicle delay using signal pre-emption strategies.	Does not perform adaptive traffic optimisation or traffic perception.
Proposed Framework	YOLOv8 + DRL	✓	✓	✓	✓	Integrates real-time traffic perception, adaptive signal optimisation, emergency vehicle prioritisation, and simulation-based evaluation within a unified intelligent traffic management framework.	

Table 2.1 further highlights these limitations and demonstrates that few existing studies combine traffic perception, adaptive signal optimisation, emergency vehicle prioritisation, and simulation-based evaluation within a single architecture. The proposed framework addresses these gaps by integrating all four components into a unified intelligent traffic management system capable of supporting both traffic efficiency and emergency response requirements.

2.8 Chapter Summary

This chapter reviewed the key concepts, technologies, and research developments relevant to intelligent traffic management. The literature demonstrated that adaptive traffic signal control has evolved from traditional fixed-time and sensor-based approaches towards more intelligent methods based on Reinforcement Learning (RL) and Deep Reinforcement Learning (DRL). Existing studies have shown that learning-based controllers can reduce congestion, waiting times, and queue lengths by adapting signal timings according to traffic conditions [50], [51], [47]. However, many of these approaches depend on simulator-generated traffic states and do not consider the uncertainties associated with real-world traffic perception [44], [48].

A critical analysis of the literature revealed three principal research gaps. First, many reinforcement learning-based traffic signal control frameworks rely heavily on simulator-generated traffic information and do not incorporate real-time traffic perception technologies, limiting their ability to operate under practical deployment conditions [44, 48, 46]. Second, although YOLO-based systems have demonstrated strong performance in vehicle detection, traffic monitoring, and emergency vehicle recognition, the information generated by these models is rarely integrated directly into adaptive traffic signal optimisation frameworks [31, 32, 59]. Third, emergency vehicle prioritisation is commonly implemented as a standalone component using signal pre-emption, communication technologies, or predefined priority rules, rather than as an integrated element of a unified intelligent traffic management architecture [33, 45, 48].

These limitations highlight the need for an intelligent framework capable of combining real-time traffic perception, adaptive traffic signal optimisation, and emergency vehicle management within a single decision-making system. Such integration has the potential to reduce the gap between traffic monitoring and traffic control while improving both traffic efficiency and emergency response performance.

Chapter 3

Research Methodology

This chapter outlines the methodology adopted to design, develop, and evaluate the proposed intelligent traffic management framework. The methodology provides the foundation for investigating how computer vision, reinforcement learning, and emergency vehicle prioritisation can be integrated within a single traffic management system. A methodological framework is essential because it establishes the procedures through which research objectives are addressed and ensures that findings are generated systematically and reliably [66], [67].

3.1 Research Philosophy

This research adopts a positivist research philosophy. Positivism assumes that knowledge can be generated through the observation and measurement of objective phenomena and that conclusions should be derived from empirical evidence rather than subjective interpretation [66], [67]. The philosophy is particularly relevant to computing, engineering, and intelligent transportation research because system performance can be assessed using measurable indicators and quantitative analysis. The choice of a positivist perspective aligns with the objectives of this study, which focus on evaluating the effectiveness of an intelligent traffic management framework.

The proposed system integrates YOLOv8-based vehicle detection, Deep Reinforcement Learning (DRL), emergency vehicle prioritisation, and traffic simulation. The effectiveness of these components can be assessed through objective performance measures, including Precision, Recall, mean Average Precision (mAP@50), average waiting time, queue length, throughput, and emergency vehicle delay. Rather than relying on theoretical assumptions alone, the study evaluates the proposed framework through controlled simulation experiments conducted within the SUMO environment. Previous research

has shown that reinforcement learning can improve adaptive traffic signal control, while computer vision techniques provide effective traffic perception capabilities [50], [51], [59].

Building on these findings, the present study investigates whether combining these technologies within a unified framework can improve overall traffic management performance. The research therefore relies on quantitative data obtained from two principal sources. The first consists of annotated traffic image datasets used for YOLOv8 model training and evaluation, while the second comprises simulation-generated traffic measurements collected from SUMO through the Traffic Control Interface (TraCI) [68], [59], [62], [61], [65]. Since the objective of the study is to evaluate system performance using measurable indicators and controlled experimentation, qualitative methods such as interviews, surveys, and questionnaires were not considered appropriate.

3.2 Research Approach

This study adopts a deductive research approach. Deductive reasoning begins with established theories, concepts, and findings reported in previous research and evaluates their applicability within a specific research context through empirical investigation [66, 67].

In contrast to inductive approaches, which seek to develop new theories from observed data, a deductive approach aims to test and validate existing knowledge using structured experimentation and objective analysis. The selection of a deductive approach is appropriate because the proposed framework builds upon established research in intelligent transportation systems, reinforcement learning, computer vision, and adaptive traffic signal control. Previous studies have demonstrated that reinforcement learning can improve traffic signal performance through adaptive decision-making, while modern computer vision techniques can provide reliable traffic perception and vehicle recognition capabilities in real time [50, 51, 59, 31]. These findings provide a theoretical foundation for the development of the proposed intelligent traffic management framework.

Drawing upon the literature reviewed in Chapter 2, the present research investigates whether integrating YOLOv8-based traffic perception, Deep Q-Network (DQN) traffic signal control, and emergency vehicle prioritisation can improve traffic management performance when compared with conventional fixed-time traffic signal control. Rather than developing new theoretical concepts, the research examines how existing techniques can be combined within a unified architecture and evaluates their effectiveness through controlled simulation experiments. The deductive approach is operationalised through the research objectives defined in Chapter 1.

These objectives are addressed through system development, experimentation, and quantitative performance evaluation using measures such as Precision, Recall, mean Average Precision (mAP@50), average waiting time, queue length, throughput, and emergency vehicle delay. The resulting evidence is used to determine whether the proposed framework delivers measurable improvements in traffic perception and traffic signal control performance.

3.3 Research Strategy

This study adopts an experimental research strategy. Experimental research is widely used in intelligent transportation systems because it enables traffic management approaches to be evaluated under controlled, repeatable, and measurable conditions [66].

The strategy is appropriate for this research because the objective is to assess whether integrating YOLOv8-based traffic perception, Deep Reinforcement Learning (DRL), and emergency vehicle prioritisation can improve traffic signal performance compared with conventional fixed-time control. The proposed framework was evaluated through a series of simulation-based experiments designed to measure its impact on traffic flow, congestion levels, and emergency vehicle movement. By conducting experiments in a controlled environment, the study ensures that performance differences can be attributed to the traffic management strategy rather than external influences. Real-world traffic experimentation was considered impractical due to safety concerns, infrastructure requirements, financial costs, and difficulties associated with controlling variables such as traffic demand, driver behaviour, and environmental conditions.

Consequently, a simulation-based environment was selected. Simulation provides a safe and cost-effective platform for reproducing traffic scenarios consistently while maintaining full control over experimental parameters. The Simulation of Urban Mobility (SUMO) platform was selected because it is widely used in intelligent transportation research and supports microscopic traffic modelling, traffic signal control, detector integration, and real-time communication through the Traffic Control Interface (TraCI) [61, 62]. These capabilities make SUMO suitable for developing and evaluating adaptive traffic management algorithms. Two traffic control approaches were evaluated under identical traffic conditions. The first was a conventional fixed-time traffic signal controller used as a baseline. The second was the proposed intelligent traffic management framework incorporating YOLOv8-based vehicle detection, a Deep Q-Network (DQN) controller, and emergency vehicle prioritisation.

3.4 Research Design

The research design provided the overall framework for the development and evaluation of the proposed intelligent traffic management system. Consistent with the positivist philosophy and deductive research approach adopted in this study, the design emphasised objective measurement, controlled experimentation, and quantitative performance evaluation. The aim was not only to develop an intelligent traffic management framework but also to assess its effectiveness using measurable performance indicators.

The design was informed by the key research gaps identified in Chapter 2. Existing reinforcement learning-based traffic control systems frequently depend on simulator-generated traffic information, computer vision solutions are often used only for traffic monitoring, and emergency vehicle prioritisation is commonly implemented as a standalone function rather than as part of an integrated traffic management framework [44, 59, 33]. These limitations provided the motivation for the development of the proposed framework. The study was organised into a sequence of interconnected stages. The first stage involved dataset collection, annotation, and preparation for vehicle detection model development. The second stage focused on training a YOLOv8 model capable of recognising both ordinary and emergency vehicles. The third stage involved the generation of traffic-state information from vehicle detections, including queue lengths, vehicle counts, traffic density estimates, and signal-state observations. These data were subsequently supplied to a Deep Q-Network (DQN) controller responsible for adaptive traffic signal optimisation. The fourth stage focused on implementing emergency vehicle prioritisation and integrating all system components within the SUMO simulation environment through the Traffic Control Interface (TraCI). The final stage involved evaluating the completed framework through controlled simulation experiments and comparing its performance with a conventional fixed-time traffic signal controller operating under equivalent traffic demand conditions.

Table 3.1: Research Design and Methodology Overview

Research Element	Description
Research Topic	Reinforcement Learning-Based Adaptive Traffic Signal Control with Emergency Vehicle Pre-emption Using YOLOv8 and SUMO.
Research Aim	To develop and evaluate an intelligent traffic management framework that combines computer vision, reinforcement learning, and emergency vehicle prioritisation for adaptive traffic signal control.
Research Philosophy	Positivism. The study relies on objective measurements obtained through simulation experiments and quantitative performance evaluation.
Research Approach	Deductive approach based on existing theories of reinforcement learning, computer vision, and intelligent transportation systems.
Research Strategy	Experimental research strategy using simulation-based evaluation within the SUMO environment.
Research Method	Quantitative research using computer vision training, reinforcement learning experiments, and statistical performance analysis.
Traffic Perception Technique	YOLOv8 object detection model for vehicle and emergency vehicle recognition.
Traffic Control Technique	Deep Q-Network (DQN) reinforcement learning agent for adaptive traffic signal optimisation.
Emergency Vehicle Management	Priority-based signal pre-emption through green extension and adaptive phase switching.
Simulation Platform	SUMO (Simulation of Urban Mobility) with TraCI integration for real-time interaction between the controller and traffic environment.
Development Tools	Python, TensorFlow, YOLOv8, OpenCV, TraCI, SUMO, NumPy, Matplotlib, and CustomTkinter.
Data Sources	Traffic image datasets for vehicle detection training and simulation-generated traffic measurements obtained from SUMO experiments.

Continued on next page

Table 3.1 – Continued from previous page

Research Element	Description
Data Collection Methods	Annotated traffic image datasets, YOLOv8 detection outputs, SUMO–TraCI traffic measurements, lane-specific queue observations, vehicle counts, signal-state information, waiting-time records, throughput measurements, emergency vehicle travel-time data, and simulation performance logs.
Validation Strategy	Testing using unseen images, repeated simulation experiments, and comparison against a fixed-time signal controller.
Vehicle Detection Metrics	Precision, Recall, and mean Average Precision (mAP@50).
Traffic Performance Metrics	Average Waiting Time, Average Queue Length, Throughput, and Emergency Vehicle Delay.
Baseline Comparison	Traditional fixed-time traffic signal control implemented within the SUMO simulation environment.
Expected Outcome	Improved traffic efficiency, reduced congestion, reduced emergency vehicle delay, and enhanced adaptive traffic signal performance.

Table 3.1 summarises the methodological framework adopted throughout the study, including the research philosophy, research approach, development tools, data sources, validation procedures, and evaluation metrics.

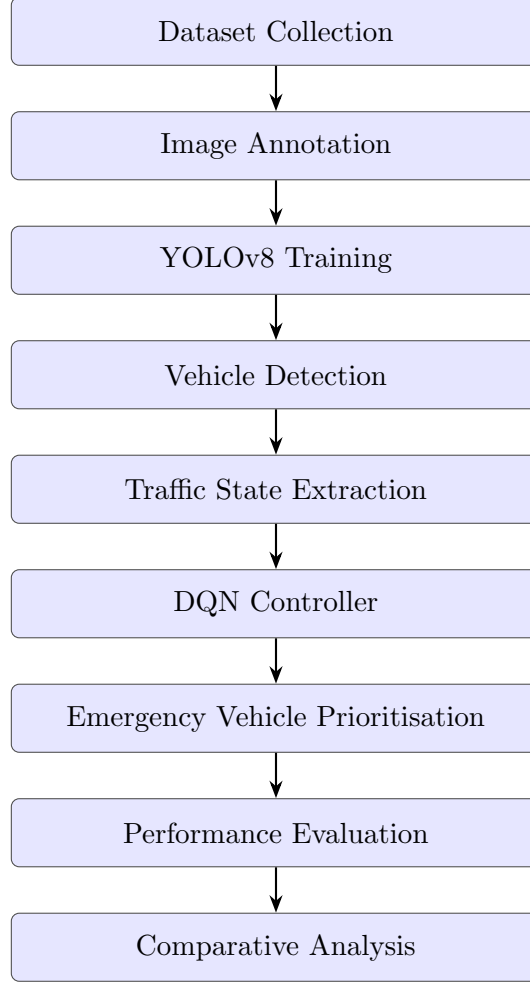


Figure 3.1: Research workflow adopted for the development and evaluation of the proposed intelligent traffic management framework.

Figure 3.1 illustrates the overall workflow adopted in this research. The process begins with dataset preparation and YOLOv8 model development, followed by traffic-state generation and Deep Q-Network control. Emergency vehicle prioritisation is subsequently incorporated into the traffic management process. The final stage focuses on system evaluation through quantitative performance analysis and comparison with a conventional fixed-time traffic signal controller.

3.5 Data Collection Methods

The proposed intelligent traffic management framework relied on multiple data sources to support both vehicle detection and traffic signal control evaluation. A multi-source data

collection strategy was adopted, combining annotated traffic image datasets, simulation-generated traffic measurements, and publicly available traffic statistics.

The primary dataset consisted of traffic images used for the development of the YOLOv8 vehicle detection model. Images were obtained from publicly available vehicle datasets and supplemented with manually annotated traffic scenes containing both ordinary vehicles and emergency vehicles [68], [59]. All images were labelled using bounding boxes and class annotations before being divided into training, validation, and testing subsets. The training set was used for model optimisation, the validation set monitored learning performance, and the testing set was reserved for final evaluation. This separation is a widely accepted machine learning practice because it supports objective assessment of model generalisation capability [58], [59].

To improve model robustness, the dataset included traffic scenes captured under different traffic densities, viewpoints, scales, and environmental conditions. Emergency vehicle categories, including ambulances, fire engines, and police vehicles, were specifically incorporated to support the emergency vehicle prioritisation component of the framework.

The second source of data was generated within the SUMO simulation environment. Through the Traffic Control Interface (TraCI), traffic-related information such as queue lengths, vehicle counts, waiting times, throughput, signal phases, traffic density estimates, and emergency vehicle delays was collected during simulation experiments [62], [61], [65]. These measurements were used both for reinforcement learning and for the quantitative evaluation of system performance.

Traffic observations obtained through SUMO and TraCI were transformed into state representations consisting of lane-specific queue lengths and the current traffic signal phase. These state vectors formed the inputs supplied to the Deep Q-Network (DQN) controller throughout training and evaluation.

3.5.1 Traffic Demand Data

To improve the realism of the simulation environment, publicly available traffic statistics published by the United Kingdom Department for Transport (DfT) were consulted during the construction of traffic demand scenarios. The DfT Road Traffic Statistics database provides information on traffic volume, vehicle counts, and traffic-flow patterns across Great Britain [69]. Although the DfT statistics were not used directly for YOLOv8 model training, they provided a reference for defining low-, medium-, and high-density traffic scenarios within the SUMO environment. Using official traffic statistics improved

the external validity of the simulation by reducing reliance on purely synthetic traffic assumptions. Similar approaches have been adopted in intelligent transportation research to support realistic simulation design and traffic-flow modelling [62], [61].

Table 3.2: Summary of Data Sources Used in the Research

Data Source	Purpose	Output Generated
Traffic Image Dataset [68]	YOLOv8 training, validation, and testing	Vehicle detection model capable of recognising ordinary and emergency vehicles
SUMO–TraCI Simulation Data	Reinforcement learning training and traffic evaluation	Queue lengths, waiting times, throughput, vehicle counts, signal states, and emergency vehicle delays
DfT Road Traffic Statistics [69]	Traffic demand scenario development	Reference traffic volumes and realistic traffic-flow patterns

Table 3.2 summarises the principal data sources used throughout the study and their respective roles within the proposed framework.

3.6 Data Analysis

The collected data were analysed using quantitative methods to evaluate both the vehicle detection component and the traffic management component of the proposed framework. The analysis focused on determining whether the integration of YOLOv8-based traffic perception, Deep Reinforcement Learning (DQN), and emergency vehicle prioritisation improved traffic management performance when compared with conventional fixed-time traffic signal control.

3.6.1 Vehicle Detection Analysis

The performance of the YOLOv8 model was evaluated using Precision, Recall, and mean Average Precision (mAP@50), which are widely used metrics in object detection research because they assess both classification accuracy and localisation quality [59]. Precision measures the proportion of detected objects that are correctly classified:

$$Precision = \frac{TP}{TP + FP} \quad (3.1)$$

where TP denotes true positive detections and FP represents false positive detections. Recall measures the proportion of relevant objects successfully detected:

$$Recall = \frac{TP}{TP + FN} \quad (3.2)$$

where FN denotes false negative detections. Mean Average Precision at an Intersection-over-Union threshold of 0.5 (mAP@50) was used as the primary detection-performance indicator because it combines classification and localisation performance into a single measure. Higher values indicate stronger object-detection capability and greater suitability for traffic perception applications.

3.6.2 Traffic Management Analysis

The effectiveness of the traffic management framework was assessed using quantitative traffic-engineering metrics commonly employed in intelligent transportation research [50], [47], [44]. The selected measures were:

- Average Waiting Time
- Average Queue Length
- Throughput
- Emergency Vehicle Delay

Average Waiting Time and Average Queue Length were used as indicators of congestion, Throughput was used to evaluate network efficiency, and Emergency Vehicle Delay was used to assess the effectiveness of the emergency vehicle prioritisation mechanism. Together, these metrics provide a comprehensive assessment of traffic performance and emergency response effectiveness.

3.6.3 Comparative Analysis

To evaluate the effectiveness of the proposed framework, the obtained results were compared with those generated by a conventional fixed-time traffic signal controller operating under equivalent traffic conditions. Performance improvements were quantified using percentage improvement calculations:

$$Improvement(\%) = \frac{M_{baseline} - M_{proposed}}{M_{baseline}} \times 100 \quad (3.3)$$

where:

- $M_{baseline}$ represents the performance value of the fixed-time controller.

- $M_{proposed}$ represents the performance value of the proposed framework.

This measure provides a standardised method for assessing the degree of improvement achieved by the adaptive traffic management system.

3.7 Achievement of Research Objectives

The evaluation framework adopted in this research was designed to assess the extent to which the objectives defined in Chapter 1 were achieved. Each objective was associated with specific development activities and quantitative performance measures to ensure objective assessment of the proposed intelligent traffic management framework.

Objective 1: To critically review existing literature on adaptive traffic signal control, reinforcement learning, computer vision-based vehicle detection, and emergency vehicle prioritisation to identify current trends and research gaps. This objective was achieved through the literature review presented in Chapter 2. The review examined conventional and intelligent traffic management approaches, identified their limitations, and established the research gap addressed by the proposed framework.

Objective 2: To develop and simulate an intelligent traffic signal control framework using Deep Reinforcement Learning and YOLO-based emergency vehicle detection within the SUMO environment to optimise traffic flow and prioritise emergency vehicles. This objective was achieved through the successful design and implementation of the proposed framework. The completed system integrates a YOLOv8 vehicle detection model, a Deep Q-Network (DQN) traffic signal controller, emergency vehicle prioritisation mechanisms, and SUMO-based traffic simulation.

Objective 3: To evaluate the effectiveness of the proposed framework using waiting time, queue length, throughput, and emergency vehicle delay metrics and compare the results with conventional fixed-time traffic signal control systems. This objective was achieved through controlled simulation experiments and comparative analysis. Performance was evaluated using quantitative indicators including average waiting time, queue length, throughput, emergency vehicle delay, and vehicle detection performance metrics.

Table 3.3: Relationship Between Research Objectives and Evaluation Metrics

Objective	Evaluation Focus	Performance Indicator
Objective 1	Review current approaches and identify research gaps	Literature Review Findings
Objective 2	Develop and implement the intelligent traffic management framework	Successful System Implementation
Objective 3	Evaluate framework effectiveness	Precision, Recall, mAP@50, Waiting Time, Queue Length, Throughput, Emergency Vehicle Delay

Table 3.3 summarises the relationship between the research objectives and the evaluation measures used throughout the study. These measures provide a structured basis for assessing both the computer vision and traffic management components of the proposed framework. The corresponding results and discussions are presented in Chapter 6.

3.8 Evaluation Framework

A multi-stage evaluation framework was adopted to assess both the computer vision and traffic management components of the proposed system. The framework was designed to provide an objective assessment of vehicle detection performance, adaptive traffic signal control, and emergency vehicle prioritisation.

The first stage focused on evaluating the YOLOv8 perception layer. The objective was to determine the model’s ability to detect and classify vehicles and emergency vehicles using standard object-detection metrics. The second stage evaluated the effectiveness of the Deep Q-Network (DQN) controller in managing traffic conditions within the SUMO simulation environment. The final stage compared the performance of the proposed framework with a conventional fixed-time traffic signal controller operating under equivalent traffic scenarios.

The evaluation process was aligned directly with the research objectives defined in Chapter 1. Vehicle detection results were used to assess the reliability of the perception layer, while traffic-management metrics were used to evaluate congestion reduction, traffic efficiency, and emergency vehicle handling. The results generated through this framework are presented and discussed in Chapter 6.

Table 3.4: Evaluation Framework

Evaluation Component	Method	Metrics
Vehicle Detection	YOLOv8 Testing	Precision, Recall, mAP@50
Traffic Signal Control	SUMO Experiments	Average Waiting Time, Average Queue Length, Throughput
Emergency Vehicle Priority	SUMO Experiments	Emergency Vehicle Travel Time, Emergency Vehicle Delay

3.9 Chapter Summary

This chapter presented the methodological framework adopted for the research. The discussion examined the research philosophy, research approach, experimental strategy, research design, data collection methods, data analysis procedures, validation techniques, and evaluation framework that guided the study. The methodology established a structured and objective approach for developing and evaluating the proposed intelligent traffic management framework. By combining computer vision datasets, simulation-generated traffic data, and quantitative performance evaluation, the chapter provided a clear foundation for the system design presented in Chapter 4 and the implementation activities described in Chapter 5.

Chapter 4

Design Process

The chapter presents the overall system architecture, traffic dataset design, YOLOv8-based vehicle detection subsystem, Deep Q-Network (DQN) controller design, emergency vehicle prioritisation strategy, and the simulation environment used for system development and evaluation. Collectively, these design elements provide the foundation for the implementation presented in Chapter 5.

4.1 Overall Architecture

The proposed intelligent traffic management framework was designed as an integrated and adaptive control system that combines computer vision, reinforcement learning, emergency vehicle prioritisation, and traffic simulation. The architecture enables continuous information exchange between traffic perception, decision-making, and traffic control components in order to improve traffic flow and intersection efficiency.

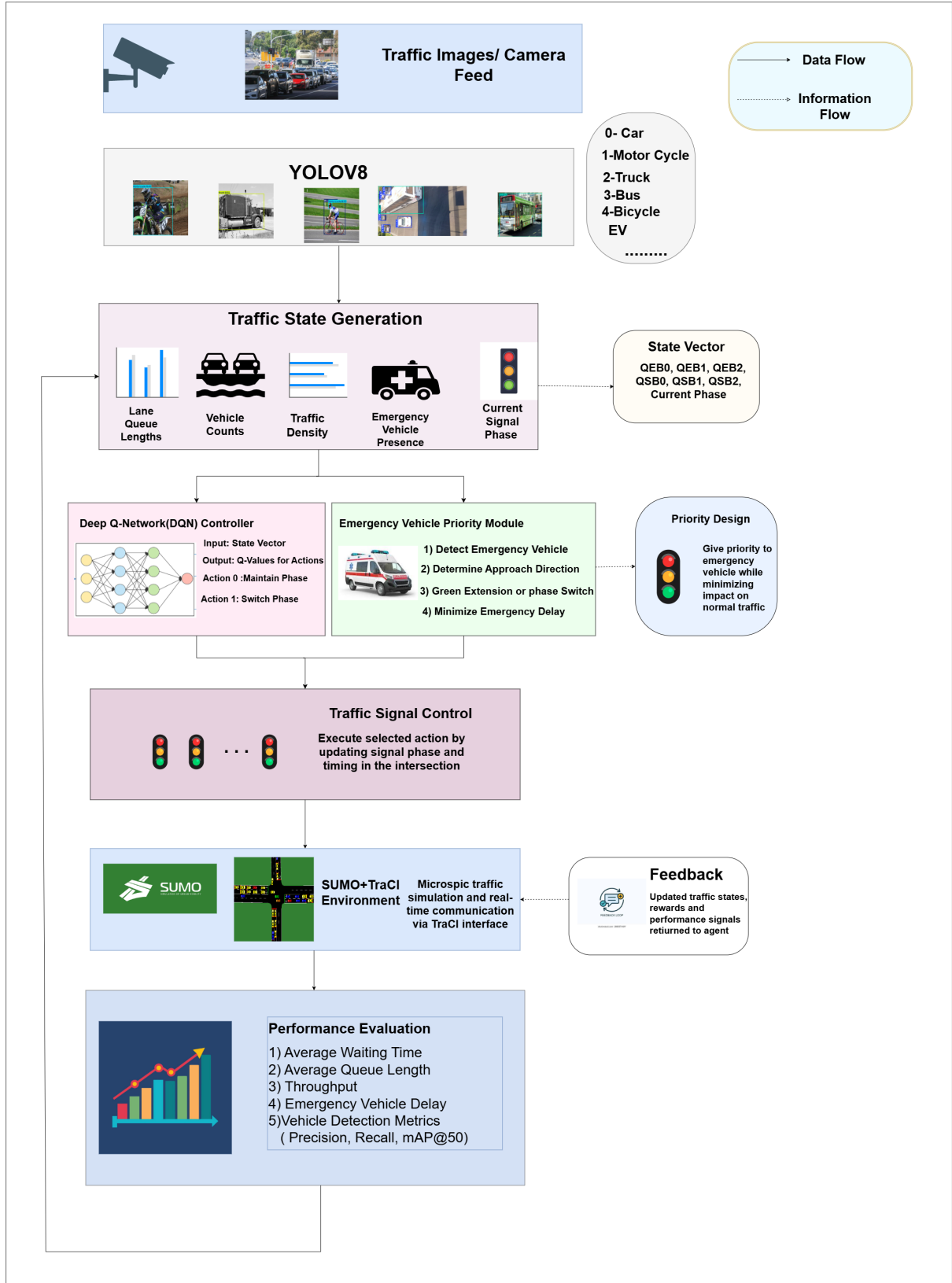


Figure 4.1: Overall architecture of the proposed intelligent traffic management framework. The framework integrates YOLOv8-based traffic perception, traffic-state generation, Deep Q-Network traffic signal control, emergency vehicle prioritisation, SUMO–TraCI simulation, and performance evaluation within a unified system.

Table 4.1: Design Components of the Proposed Framework

Component	Function
YOLOv8 Detection Layer	Vehicle and emergency vehicle recognition
Traffic-State Generation	Vehicle counts, queue lengths and density estimation
DQN Controller	Adaptive traffic signal optimisation
Emergency Vehicle Module	Signal pre-emption and green extension
SUMO–TraCI Environment	Traffic simulation and controller interaction

4.2 Traffic Dataset Design

The vehicle detection component of the proposed framework requires a representative dataset capable of supporting both general vehicle recognition and emergency vehicle identification. Consequently, a custom traffic dataset was constructed using publicly available vehicle images together with additional manually annotated traffic scenes. The initial dataset was obtained from a publicly available vehicle detection dataset and supplemented with images containing emergency vehicles such as ambulances, police vehicles, and fire engines [68].

Two separate datasets were prepared for model development. The first dataset was used to train a general vehicle detection model and contained common road-user categories including cars, buses, trucks, motorcycles, and bicycles. A second dataset was prepared specifically for emergency vehicle recognition and contained ambulances, police vehicles, and fire engines. This separation enabled independent optimisation of normal traffic perception and emergency vehicle recognition, reflecting the different operational roles of these components within the proposed framework. All images were manually annotated using the custom annotation tool developed during this project, enabling the generation of YOLO-compatible bounding-box labels for supervised learning.

To improve model robustness and generalisation capability, the dataset was designed to include a variety of traffic conditions, vehicle categories, camera viewpoints, object scales, and traffic densities. Particular attention was given to ensuring that emergency vehicles were represented because emergency vehicle recognition forms a central component of the proposed traffic management framework. The completed dataset was subsequently organised into training, validation, and testing subsets to support model development, parameter optimisation, and objective performance evaluation. This structure provided the foundation for the YOLOv8-based traffic perception layer used throughout the study.

4.3 Vehicle Detection System Design

The traffic perception layer of the proposed framework was developed using YOLOv8. The model was selected because it provides a balance between detection accuracy and computational efficiency, making it suitable for real-time traffic monitoring applications. Unlike conventional multi-stage object detection approaches, YOLOv8 performs object localisation and classification within a single network architecture, enabling rapid processing of traffic scenes while maintaining reliable detection performance.

The detector was designed to recognise both ordinary vehicles and emergency vehicles, allowing traffic information to be extracted directly from image data. The resulting detections are subsequently transformed into traffic-state information, including vehicle counts, traffic density estimates, queue-length observations, and emergency vehicle presence. This information forms the perception layer of the proposed intelligent traffic management framework and serves as the primary input to the adaptive traffic signal controller. The selection of YOLOv8 was also motivated by its support for real-time operation, anchor-free detection mechanism, and improved feature extraction capabilities, which have been shown to enhance object detection performance in transportation-related applications. Consequently, YOLOv8 provides a suitable foundation for integrating computer vision with adaptive traffic signal optimisation and emergency vehicle prioritisation.

4.3.1 Overall YOLOv8 Architecture

The YOLOv8 detector employed in this research consists of three principal components: the backbone network, the neck network, and the detection head.

- **Backbone Network:** Responsible for extracting hierarchical visual features from the input image using convolutional layers, C2f modules, and a Spatial Pyramid Pooling Fast (SPPF) block.
- **Neck Network:** Performs multi-scale feature fusion using Feature Pyramid Network (FPN) and Path Aggregation Network (PAN) structures, improving detection capability across different object sizes.
- **Detection Head:** Produces class labels, confidence scores, and bounding-box coordinates through an anchor-free detection mechanism.

The architecture follows the general design principles of YOLOv8 described in recent literature while being adapted specifically for traffic vehicle and emergency vehicle recognition [59], [70]. The generated detections are transformed into traffic-state information

including vehicle counts, queue lengths, traffic density estimates, emergency vehicle presence, and current signal phase observations. These variables form the inputs supplied to the Deep Q-Network traffic signal controller.

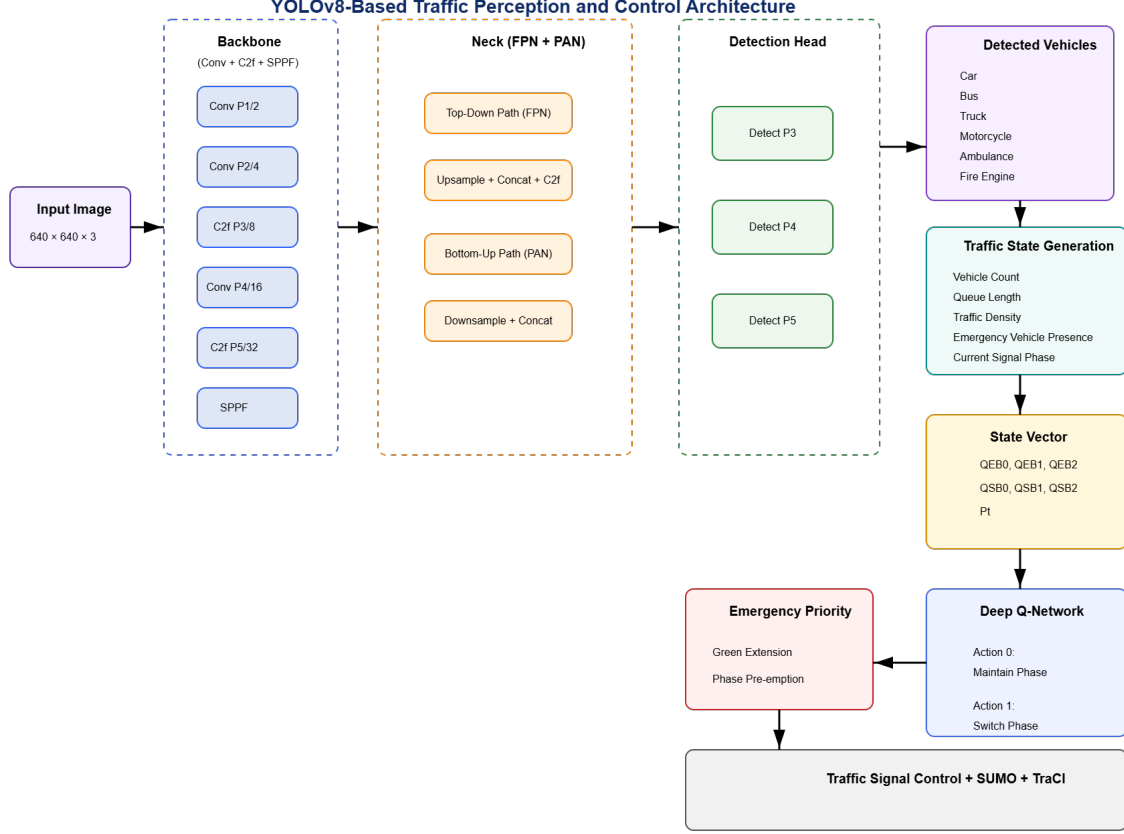


Figure 4.2: YOLOv8-based traffic perception architecture and traffic-state generation process used within the proposed framework.

4.3.2 Vehicle Detection Pipeline

The vehicle detection pipeline was designed to transform raw traffic images into structured traffic-state information that could be used by the adaptive traffic signal controller. The process begins with image acquisition from traffic camera data. Each image is supplied to the YOLOv8 detector, where the backbone, neck, and detection head collaboratively perform feature extraction, feature fusion, and object localisation.

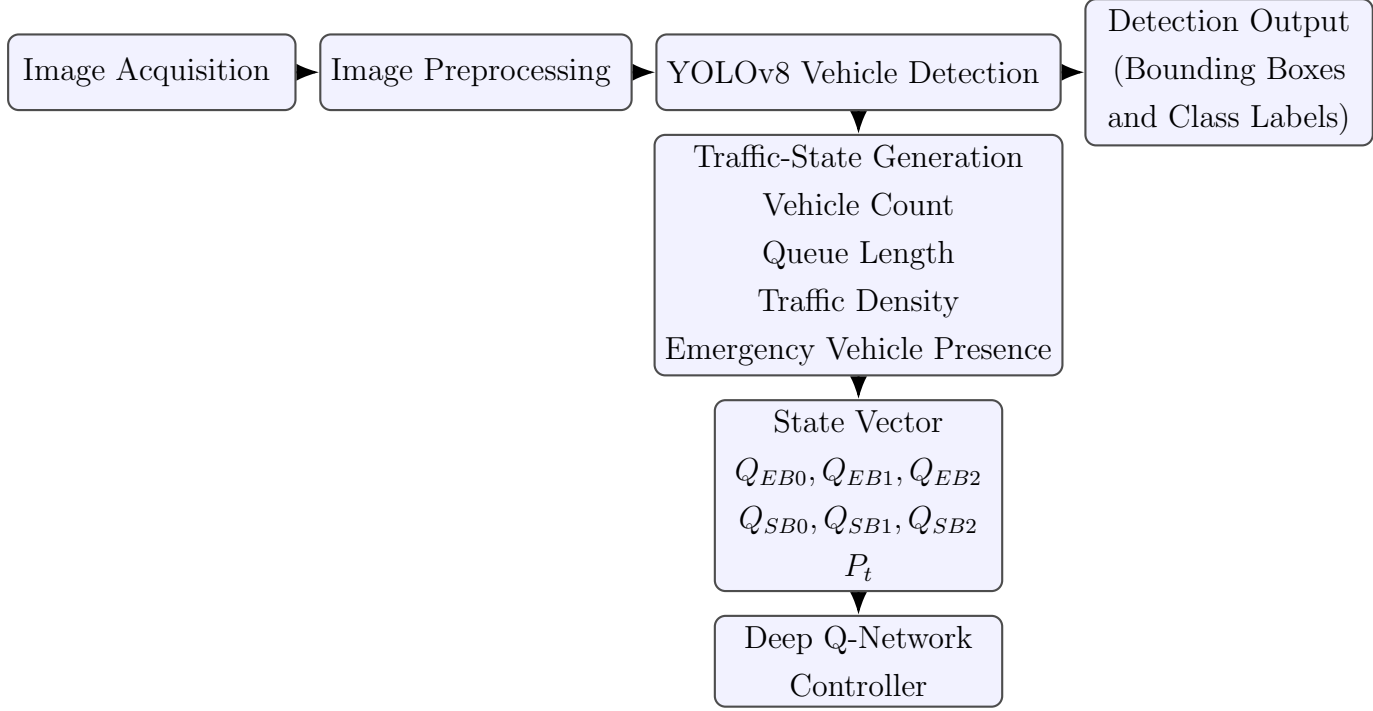


Figure 4.3: Vehicle detection pipeline used to transform traffic images into traffic-state information for adaptive traffic signal control.

4.4 Reinforcement Learning Controller Design

The adaptive traffic signal controller was designed using a Deep Q-Network (DQN) reinforcement learning architecture. Reinforcement learning was selected because it enables traffic signal decisions to be learned directly through interaction with the traffic environment rather than relying on predefined timing plans. Traffic signal control was formulated as a Markov Decision Process (MDP):

$$M = (S, A, R, P, \gamma) \quad (4.1)$$

where:

- S represents the traffic state space;
- A represents the available signal-control actions;
- R represents the reward function;
- P represents state-transition probabilities;
- γ represents the discount factor.

The controller continuously observes current traffic conditions, selects signal actions, evaluates the resulting traffic performance, and updates its policy according to the received rewards.

4.4.1 State Representation

The state representation was designed to provide the controller with information regarding current congestion levels while maintaining a compact feature space. The state vector is represented as:

$$S_t = \{Q_{EB0}, Q_{EB1}, Q_{EB2}, Q_{SB0}, Q_{SB1}, Q_{SB2}, P_t\} \quad (4.2)$$

where:

- Q_{EB0} , Q_{EB1} and Q_{EB2} denote queue lengths on eastbound lanes;
- Q_{SB0} , Q_{SB1} and Q_{SB2} denote queue lengths on southbound lanes;
- P_t denotes the current traffic signal phase.

These variables are extracted directly from SUMO lane-area detectors through TraCI communication.

4.4.2 Action Space

The controller operates using a discrete action space consisting of two possible traffic signal actions:

$$A = \{0, 1\} \quad (4.3)$$

where:

- Action 0 = Maintain the current signal phase;
- Action 1 = Switch to the next phase.

A minimum green-time constraint was introduced to prevent excessive signal switching and improve traffic stability.

4.4.3 Reward Function Design

The reward function was designed to minimise congestion by penalising queue accumulation within the monitored traffic approaches. The reward function is defined as:

$$R_t = - \sum_{i=1}^n Q_i \quad (4.4)$$

where:

- R_t represents the reward at time step t ;
- Q_i represents the queue length observed on lane i ;
- n represents the total number of monitored lanes.

This formulation encourages the controller to select signal actions that reduce traffic congestion and improve intersection performance.

4.4.4 Deep Q-Network Architecture

The reinforcement learning controller was implemented using a feedforward neural network consisting of two hidden layers. The network receives traffic-state information and outputs Q-values representing the expected reward of each available traffic signal action. The network architecture consists of:

- Input Layer (7 neurons);
- Hidden Layer 1 (24 neurons, ReLU activation);
- Hidden Layer 2 (24 neurons, ReLU activation);
- Output Layer (2 neurons).

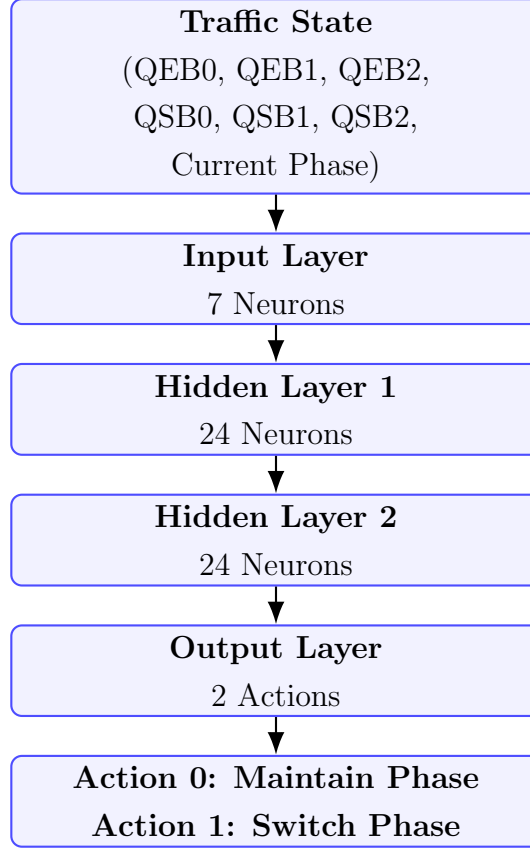


Figure 4.4: Deep Q-Network architecture used for adaptive traffic signal control.

Figure 4.4 illustrates the Deep Q-Network architecture adopted in this study. The network receives traffic-state information consisting of lane-specific queue lengths and the current signal phase. The input layer contains seven neurons corresponding to the state variables. Two fully connected hidden layers with 24 neurons each are used to learn the relationship between traffic conditions and signal-control actions. The output layer contains two neurons representing the available actions: maintaining the current signal phase or switching to the next phase. The action with the highest predicted Q-value is selected by the controller during operation.

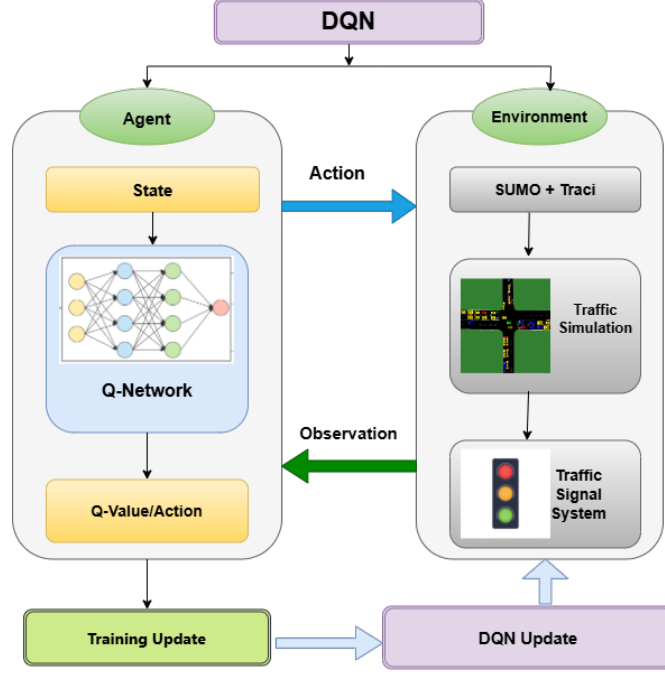


Figure 4.5: Conceptual architecture of the Deep Q-Network controller used for adaptive traffic signal control. Adapted from Ouyang et al. [1].

Figure 4.5 illustrates the conceptual operation of the Deep Q-Network controller adopted in this study. Traffic-state information is obtained from the SUMO simulation environment through TraCI communication and supplied to the DQN agent. The neural network estimates Q-values for the available traffic signal actions and selects the action expected to maximise long-term reward. Following each interaction, the environment provides updated traffic observations and rewards that are used to improve the network through iterative training updates. This process enables the controller to learn adaptive signal-control policies capable of responding to changing traffic conditions.

4.4.5 DQN Learning Algorithm

The Deep Q-Network controller learns traffic signal control policies through continuous interaction with the SUMO simulation environment. At each simulation step, the controller observes the current traffic state, selects an action using an ϵ -greedy policy, receives a reward based on queue lengths, and updates the neural network parameters accordingly. Algorithm 1 summarises the DQN learning process adopted in this study.

Algorithm 1 Deep Q-Network Traffic Signal Control

```
1: Initialise DQN network
2: Initialise traffic state  $S_t$ 
3: while simulation active do
4:   Observe current traffic state  $S_t$ 
5:   Select action  $A_t$  using  $\epsilon$ -greedy policy
6:   Apply signal-control action
7:   Advance SUMO simulation
8:   Observe new state  $S_{t+1}$ 
9:   Compute reward  $R_t$ 
10:  Update DQN parameters
11:  Set  $S_t \leftarrow S_{t+1}$ 
12: end while
```

4.5 Emergency Vehicle Prioritisation Design

Emergency vehicle prioritisation was incorporated into the framework to minimise delays experienced by ambulances, police vehicles, and fire engines. The prioritisation subsystem continuously monitors detected vehicles and evaluates whether emergency vehicle intervention is required. The module operates alongside the traffic signal controller and temporarily modifies signal operation when emergency vehicles approach an intersection.

4.5.1 Emergency Vehicle Detection

Emergency vehicles are identified using vehicle classification outputs generated by the YOLOv8 perception module and simulation-based vehicle type information available within SUMO. The supported emergency vehicle classes include:

- Ambulance
- Fire Engine
- Police Vehicle

Detection of any of these categories activates the prioritisation process.

4.5.2 Direction Classification

The current direction of travel is determined using the road segment occupied by the emergency vehicle. Two movement groups are considered:

- North-South (NS)

- East-West (EW)

The determined direction is used to identify the signal phase that should receive priority.

4.5.3 Signal Pre-emption Strategy

The prioritisation mechanism operates according to two control rules. If the desired traffic signal phase is already active, the green phase duration is extended:

$$G_{new} = G_{current} + \Delta G \quad (4.5)$$

where:

- $G_{current}$ represents the current green duration;
- ΔG represents the additional green-time allocation.

If the desired phase is not active, the remaining duration of the current phase is reduced to accelerate transition to the required signal state.

4.5.4 Recovery Mechanism

After the emergency vehicle passes the intersection, all temporary signal modifications are removed and normal traffic signal control resumes automatically. The recovery mechanism prevents unnecessary disruption to ordinary traffic flows while preserving emergency vehicle priority benefits.

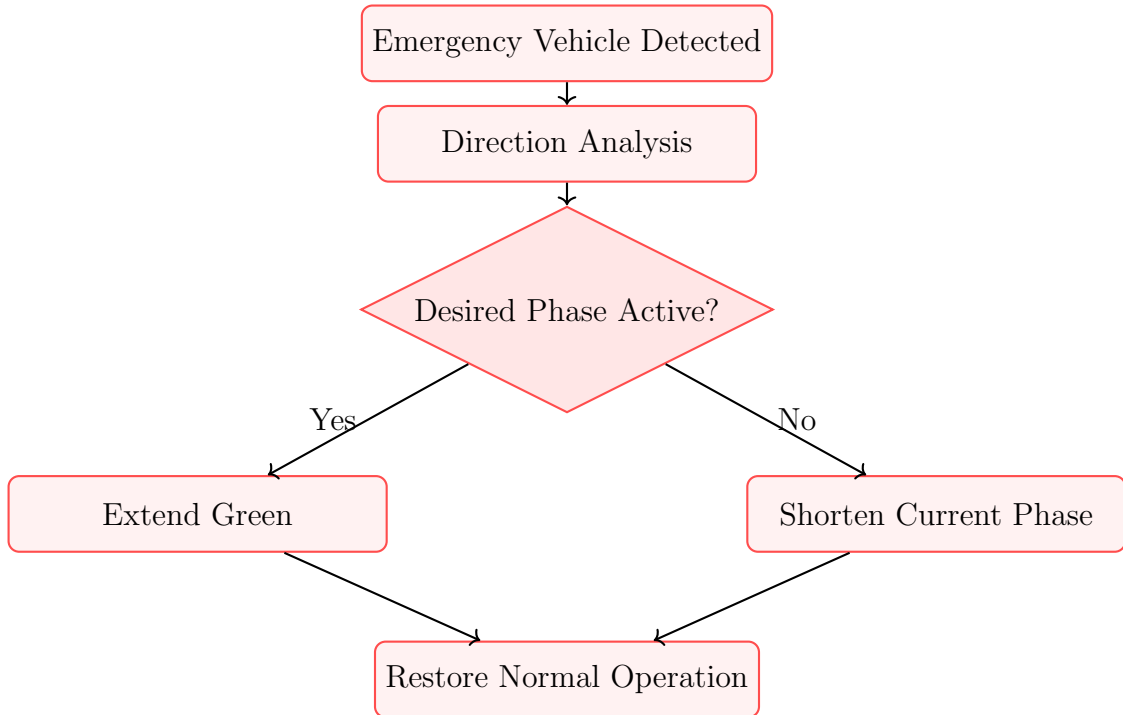


Figure 4.6: Emergency vehicle prioritisation workflow.

Emergency vehicle prioritisation process:

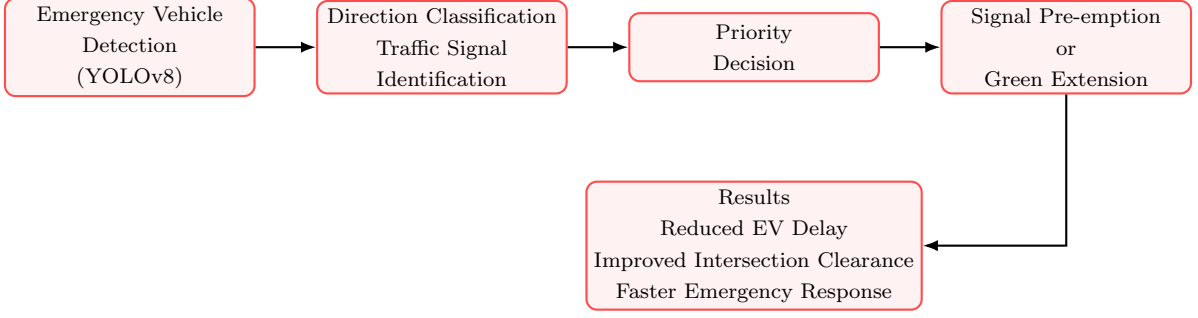


Figure 4.7: Emergency vehicle prioritisation process adopted in the proposed framework.

Figure 4.7 illustrates the emergency vehicle prioritisation process adopted in the proposed framework. Emergency vehicles are first detected using the YOLOv8 perception module. The system then determines the vehicle movement direction and identifies the relevant traffic signal. Based on this information, a priority decision is made. Depending on the current signal state, the controller either extends the active green phase or initiates signal pre-emption to facilitate emergency vehicle movement. These actions aim to reduce emergency vehicle delay, improve intersection clearance, and support faster emergency response.

4.6 Evaluation Design

A structured evaluation strategy was developed to assess the effectiveness of the proposed intelligent traffic management framework. The primary objective of the evaluation is to determine whether the integration of YOLOv8-based traffic perception, Deep Reinforcement Learning (DQN), and emergency vehicle prioritisation produces measurable improvements when compared with a conventional fixed-time traffic signal controller. The evaluation process consists of two complementary stages. The first stage focuses on the performance of the YOLOv8 vehicle detection model, while the second stage evaluates the effectiveness of the adaptive traffic signal control framework within the SUMO simulation environment. Four primary traffic-performance indicators were selected:

- Average Waiting Time
- Average Queue Length
- Throughput
- Emergency Vehicle Delay

These metrics were selected because they collectively capture congestion levels, traffic efficiency, network utilisation, and emergency response performance, and are widely used

in intelligent transportation research [50], [47], [44]. To evaluate the computer vision component, object detection performance was assessed using Precision, Recall, and mean Average Precision (mAP@50). These measures provide a comprehensive assessment of vehicle classification accuracy and localisation performance [59].

Chapter 5

Implementation

This chapter describes how the proposed intelligent traffic management framework was implemented. The implementation transformed the design concepts presented in Chapter 4 into a functioning system capable of detecting vehicles, managing traffic signals adaptively, and supporting emergency vehicle movement within a simulated environment.

The completed framework consists of five main components: a dataset annotation application, a YOLOv8 training environment, a traffic-state acquisition module, a Deep Q-Network (DQN) traffic signal controller, and an emergency vehicle priority subsystem. These components were connected through the Traffic Control Interface (TraCI) and evaluated using the SUMO simulation platform.

5.1 Dataset Annotation Tool Implementation

A dedicated annotation application was developed to produce YOLO-compatible labels for vehicle detection training. The software was created using Python and incorporates OpenCV, Pillow, NumPy, and CustomTkinter libraries. The tool allows images to be loaded individually, vehicle classes to be selected, and bounding boxes to be drawn manually around traffic participants. Once annotation is complete, label files are exported automatically in YOLO format. The decision to develop a custom tool rather than rely entirely on existing annotation software was motivated by the need to support a simplified workflow and project-specific vehicle categories.

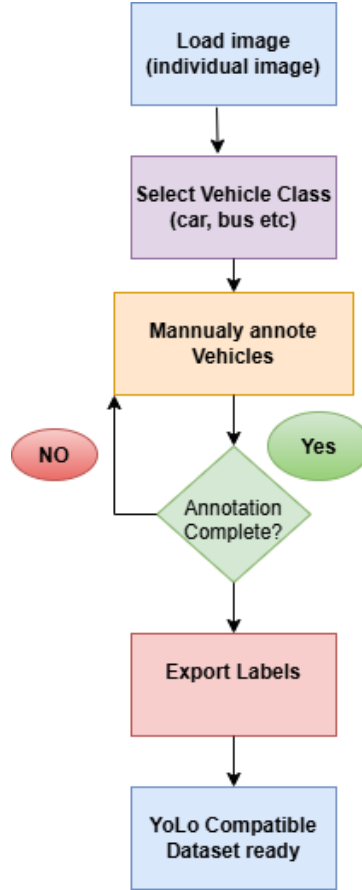


Figure 5.1: Custom image annotation application

Table 5.1: Vehicle Classes Used During Annotation for normal vehicles

Class ID	Vehicle Category
0	Car
1	Motorcycle
2	Truck
3	Bus
4	Bicycle

5.2 YOLOv8 Training Environment

A separate training application was implemented to manage dataset validation, training configuration, model training, and performance monitoring. The application verifies directory structures automatically before training begins and reduces the likelihood of errors caused by missing images, annotations, or dataset configuration files. Transfer learning was adopted during implementation. Pretrained YOLOv8 weights were used as an initial starting point and fine-tuned using the prepared traffic dataset. This strat-

egy reduces training time and commonly produces better performance than training a detector entirely from random parameter initialisation [59].

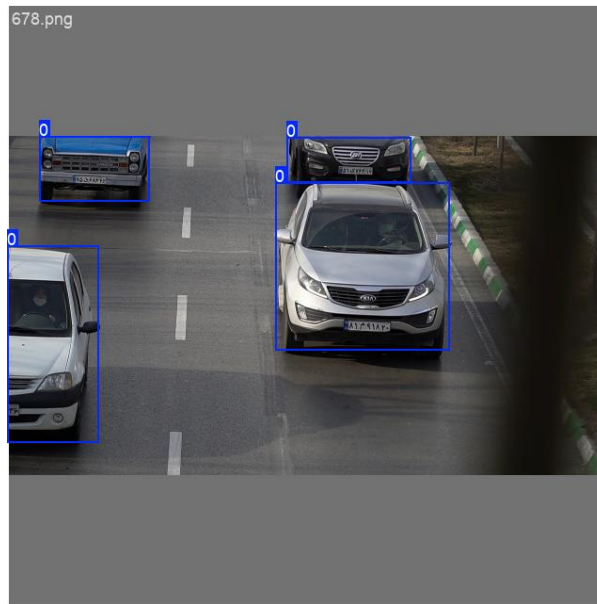


Figure 5.2: **Example annotated training image used during YOLOv8 model development.**

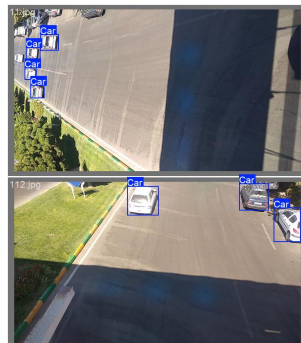


Figure 5.3: **Ground-truth validation annotations.**

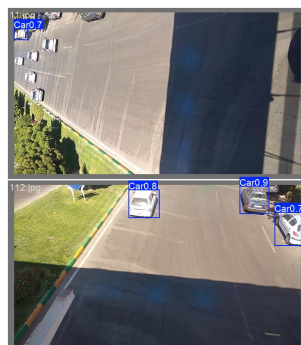


Figure 5.4: **YOLOv8 prediction output on validation data.**

5.3 Traffic State Acquisition

Traffic information was collected through lane-area detectors positioned around the simulated intersection. Detector measurements were retrieved continuously through the TraCI interface and converted into a compact state representation suitable for reinforcement learning. The implementation utilised queue length measurements obtained from six monitored lanes together with the currently active signal phase:

$$S_t = \{Q_{EB0}, Q_{EB1}, Q_{EB2}, Q_{SB0}, Q_{SB1}, Q_{SB2}, P_t\} \quad (5.1)$$

This representation provides sufficient information for traffic-control decisions while avoiding unnecessary computational complexity.

Table 5.2: Traffic-State Variables

Variable	Description
QEB0	Eastbound lane queue length
QEB1	Eastbound lane queue length
QEB2	Eastbound lane queue length
QSB0	Southbound lane queue length
QSB1	Southbound lane queue length
QSB2	Southbound lane queue length
Pt	Current signal phase

5.4 DQN Traffic Signal Controller

The adaptive signal controller was implemented using TensorFlow and Keras. A fully connected neural network was employed to estimate state-action values and generate signal-control decisions. The implemented network contains an input layer representing the traffic state, two hidden layers containing 24 neurons each, and an output layer representing the available actions. At every simulation step, the controller receives the current traffic state, predicts action values, selects an action using an epsilon-greedy policy, and applies the selected traffic signal operation through TraCI. The controller implementation follows the DQN learning process presented in Algorithm 1.

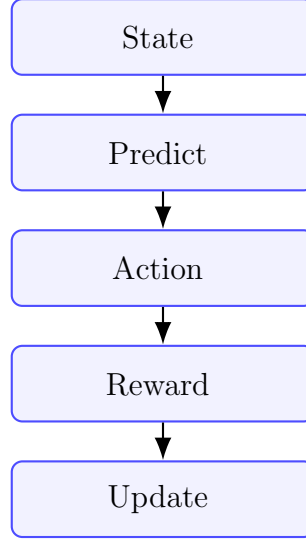


Figure 5.5: Simplified DQN decision-making cycle used for traffic signal control.

5.5 Reward Implementation

The reward mechanism was implemented using total queue length. This design encourages actions that reduce vehicle accumulation within the network. The reward function adopted during implementation is given by:

$$R_t = - \sum_{i=1}^n Q_i \quad (5.2)$$

Negative reward values were assigned to queue accumulation. Consequently, the controller was encouraged to minimise congestion and reduce waiting vehicles.

5.6 Emergency Vehicle Priority Implementation

An emergency vehicle management module was implemented alongside the reinforcement learning controller. During simulation, all active vehicles are inspected continuously. When an emergency vehicle is detected, its movement direction and upcoming signal location are identified. The next signal is subsequently modified to provide priority whenever required. If the required phase is already active, the green period can be extended temporarily. If the required phase is currently inactive, the remaining duration of the current phase may be reduced to accelerate transition toward the desired movement.

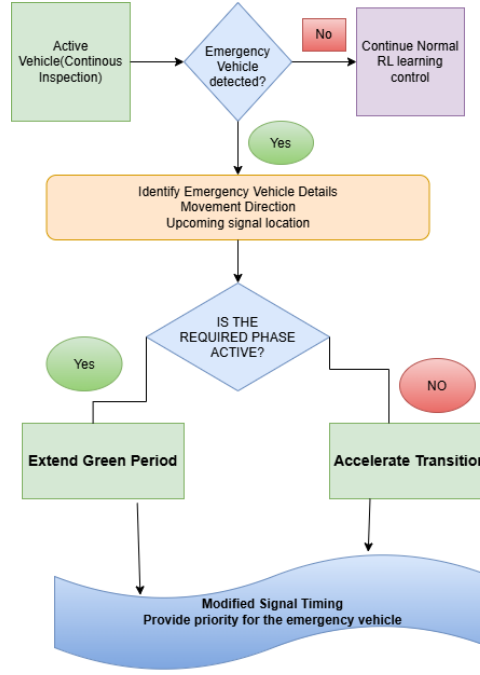


Figure 5.6: Emergency vehicle priority procedure

Table 5.3: Emergency Vehicle Priority Operations

Condition	Action
Desired phase active	Extend green phase
Desired phase inactive	Reduce current phase duration
Emergency vehicle clears intersection	Restore normal signal operation

5.7 SUMO–TraCI Integration

Communication between the controller and the simulation environment was achieved through TraCI. The interface enables bidirectional communication between Python and SUMO, allowing traffic information to be collected and signal-control actions to be applied in real time [65]. The implemented control loop follows a sequence consisting of state acquisition, action generation, simulation advancement, reward calculation, and policy update. This process continues throughout execution and enables the controller to learn directly from interaction with the environment.

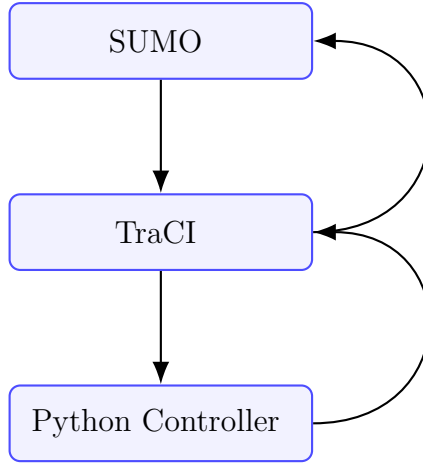


Figure 5.7: Communication architecture between SUMO, TraCI, and the external Python controller.

Figure 5.7 illustrates the communication structure adopted in the proposed framework. SUMO functions as the traffic simulation environment, while the Traffic Control Interface (TraCI) provides bidirectional communication between SUMO and the external Python-based controller. Through TraCI, the controller retrieves traffic-state information including queue lengths, vehicle counts, waiting times, and traffic signal states. In return, the controller transmits traffic signal actions back to SUMO, enabling real-time interaction between the reinforcement learning agent and the simulation environment.

5.7.1 SUMO Simulation Configuration

The SUMO simulation environment requires several configuration files to define the road network, traffic demand, detector configuration, and controller interaction.

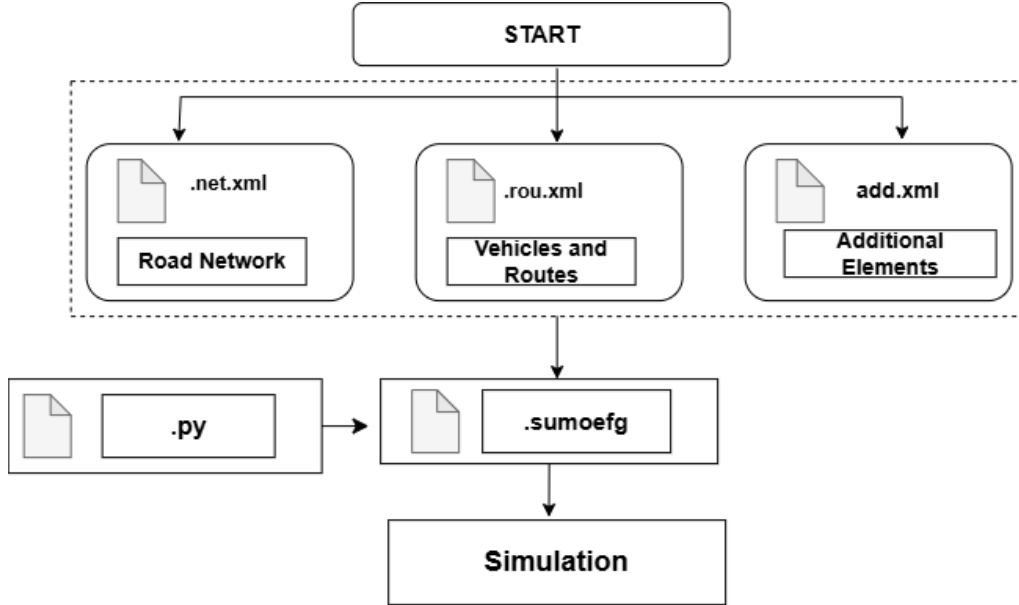


Figure 5.8: SUMO simulation configuration workflow showing the relationship between the road network file, route definitions, additional configuration files, Python controller, and SUMO configuration file.

Figure 5.8 illustrates the configuration structure used within the SUMO environment. The road network is defined using a network file (`.net.xml`), while vehicle demand and route definitions are specified through route files (`.rou.xml`). Additional simulation elements, including detectors and traffic signal definitions, are provided through supplementary configuration files (`.add.xml`). The simulation is executed through a SUMO configuration file (`.sumocfg`), which references all required network and traffic inputs. Communication between the SUMO environment and the external Python controller is achieved through the Traffic Control Interface (TraCI), enabling real-time retrieval of traffic information and dynamic traffic signal control. This configuration structure provides the foundation for integrating vehicle detection, reinforcement learning, and emergency vehicle prioritisation within a single simulation environment.

5.8 Implementation Challenges

A number of technical challenges emerged during development. The preparation of training data required substantial manual effort, particularly for emergency vehicle classes that were less common than ordinary traffic participants. The integration of reinforcement learning and traffic simulation also introduced debugging challenges because communication between SUMO and the external controller had to remain synchronised throughout execution. An additional challenge involved balancing emergency vehicle priority actions with normal traffic operation. Excessive priority intervention occasionally increased de-

lays for surrounding traffic and therefore required careful tuning.

Table 5.4: Implementation Challenges and Solutions

Challenge	Resolution
Limited emergency vehicle data	Additional image collection and annotation
SUMO–TraCI synchronisation	Repeated testing and debugging
Traffic signal instability	Minimum green-time constraint

5.9 Chapter Summary

This chapter presented the implementation of the intelligent traffic management framework. The discussion covered the annotation application, training environment, DQN controller, emergency vehicle priority mechanism, and SUMO integration process. The following chapter evaluates the completed framework using quantitative performance indicators.

Chapter 6

Evaluation of Results

This chapter evaluates the performance of the proposed intelligent traffic management framework. The evaluation focuses on determining whether the integration of YOLOv8-based traffic perception, Deep Q-Network (DQN) traffic signal control, and emergency vehicle prioritisation can improve traffic management performance within a SUMO-based simulation environment.

6.1 Vehicle Detection Performance

The performance of the YOLOv8 vehicle detection model was evaluated using a dedicated testing dataset. Evaluation was conducted using Precision, Recall, and mean Average Precision at an Intersection-over-Union threshold of 0.5 (mAP@50), which are widely used measures for assessing object-detection performance because they quantify both classification accuracy and localisation quality [59].

Table 6.1: YOLOv8 Vehicle Detection Performance

Metric	Value
Precision	42.59%
Recall	33.05%
mAP@50	34.50%
mAP@50–95	19.86%
Best F1 Score	40.00%
Optimal Confidence Threshold	0.306

The detector successfully recognised vehicles within traffic scenes and generated traffic information suitable for adaptive traffic signal control. However, the achieved Recall and mAP values indicate that some vehicles were not detected successfully, particularly under more challenging traffic conditions.

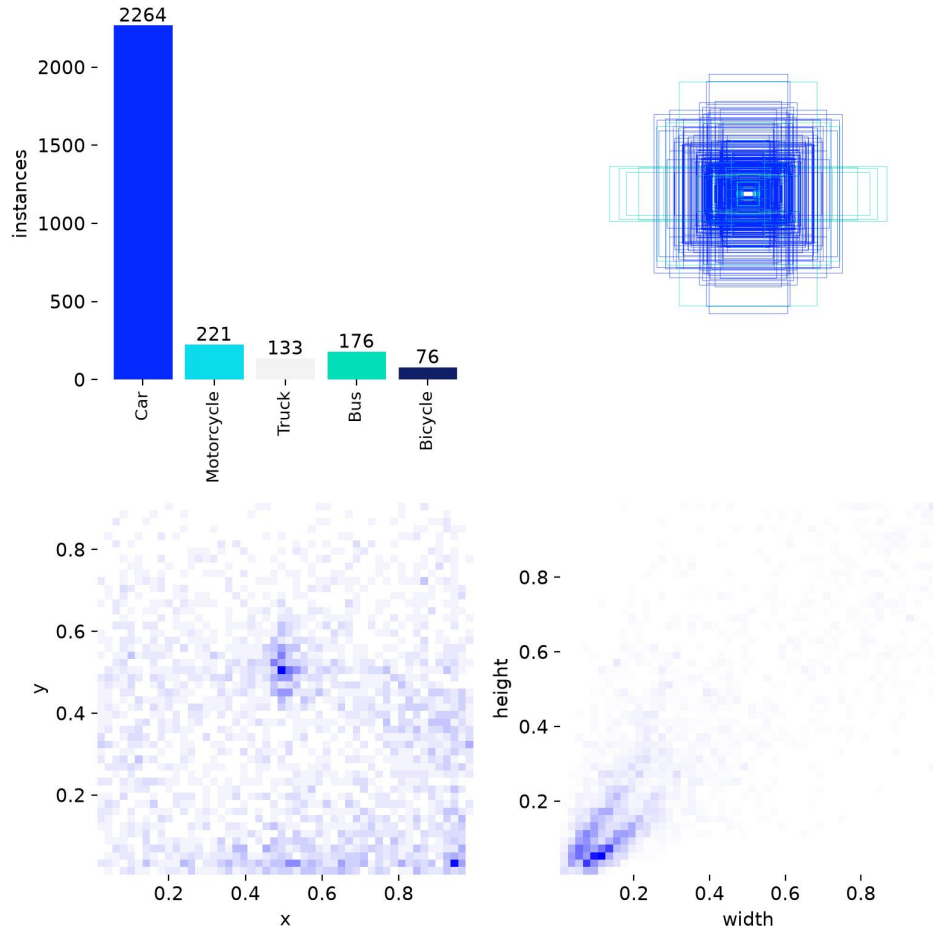


Figure 6.1: Distribution of labelled vehicle instances across training categories.

Figure 6.1 illustrates the distribution of labelled vehicle instances within the training dataset. Car instances represented the majority of annotations, whereas buses, trucks, motorcycles, and bicycles were comparatively less common. This class imbalance may have contributed to variations in class-specific detection performance.

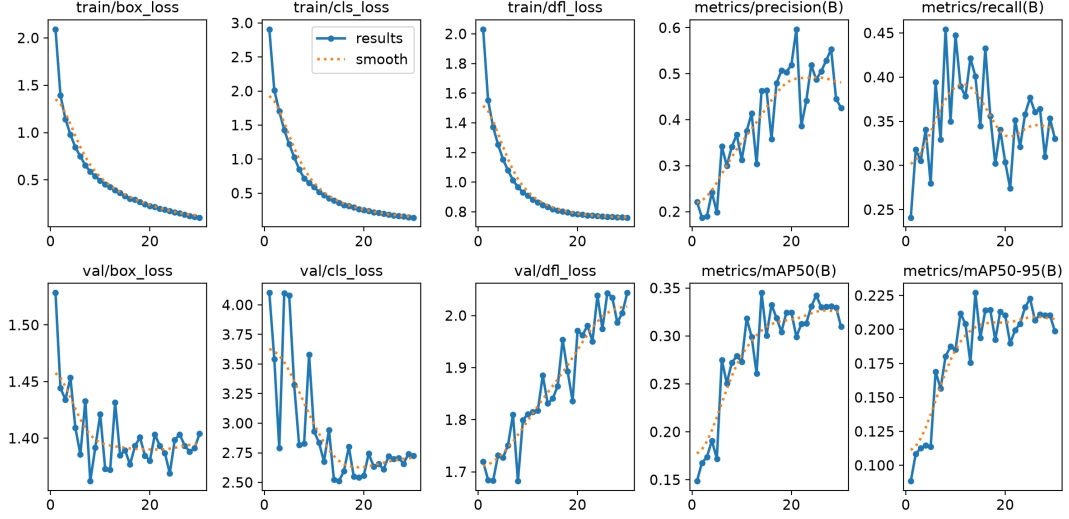


Figure 6.2: YOLOv8 training performance over 30 epochs.

Figure 6.2 presents the training behaviour of the YOLOv8 model. The reduction in training losses and gradual improvement in validation metrics indicate successful model convergence. Nevertheless, the final results suggest that additional training data and a more balanced dataset could further improve performance.

6.2 Visual Evaluation of Detection Results

In addition to quantitative evaluation, a visual assessment was conducted to examine the quality of the detections generated by the YOLOv8 model. Ground-truth annotations were compared directly with model predictions to evaluate localisation accuracy and classification performance.

0.48

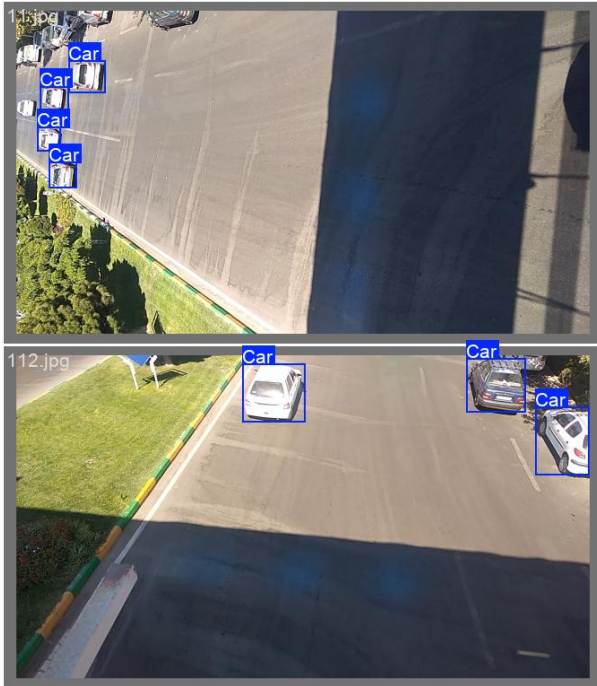


Figure 6.3: Ground-truth annotations

0.48

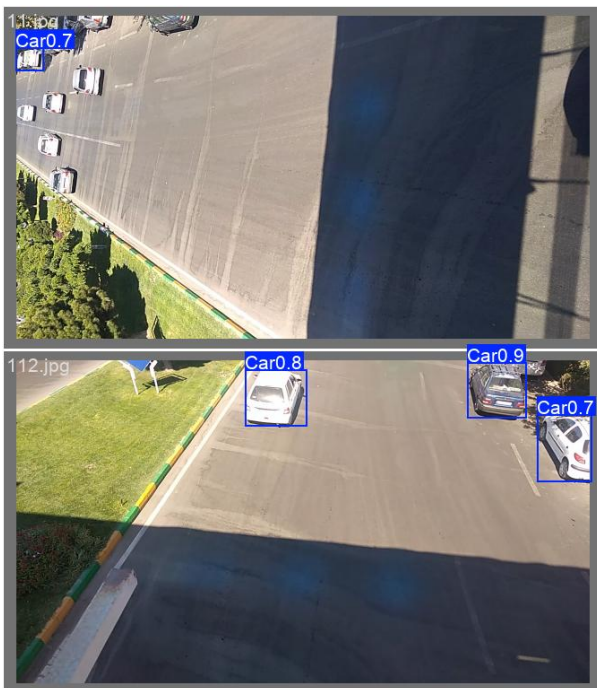


Figure 6.4: YOLOv8 predictions

Figure 6.5: Comparison between validation annotations and YOLOv8 predictions for Validation Set 1.



Figure 6.6: Ground-truth annotations

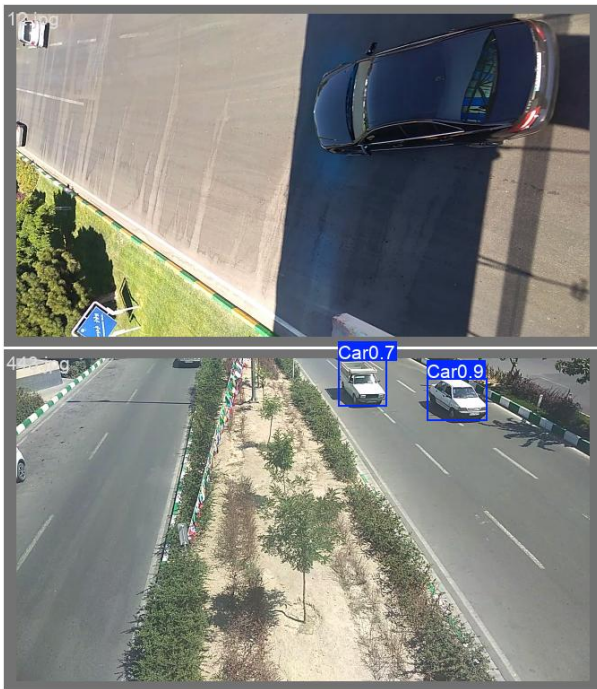


Figure 6.7: YOLOv8 predictions

Figure 6.8: Comparison between validation annotations and YOLOv8 predictions for Validation Set 2.

Figures 6.5 and 6.8 demonstrate that the detector successfully identified the majority of visible vehicles and generated bounding boxes that aligned closely with the corre-

sponding annotations. Most prediction confidence values ranged between 0.70 and 1.00, indicating a reasonable degree of confidence in the detections. Although some missed detections and localisation inaccuracies were observed, particularly for less represented vehicle classes, the resulting outputs were sufficiently accurate for traffic-state generation and adaptive traffic signal control.

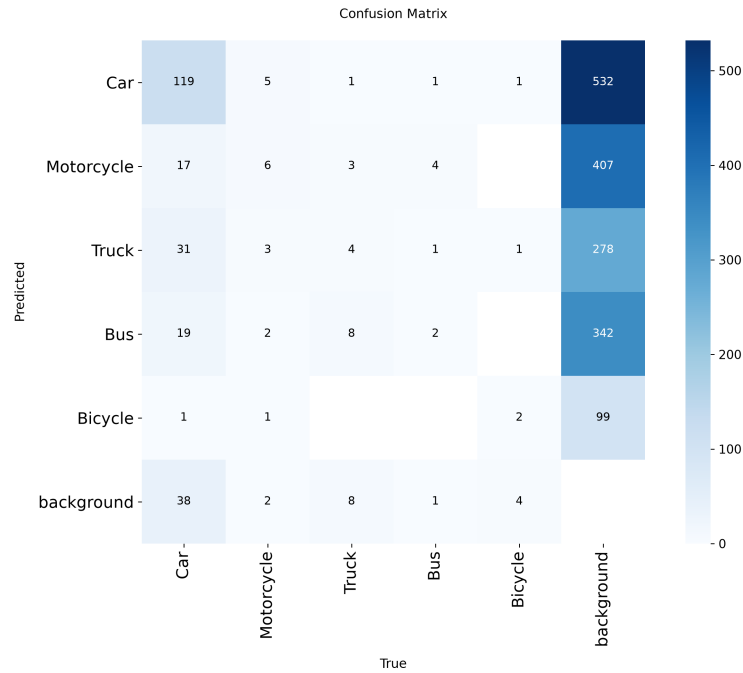


Figure 6.9: YOLOv8 confusion matrix.

The confusion matrix indicates that detection performance was strongest for the car category, which represented the largest proportion of training examples. Lower accuracy was observed for buses, trucks, motorcycles, and bicycles, which is consistent with the class imbalance shown in Figure 6.1.

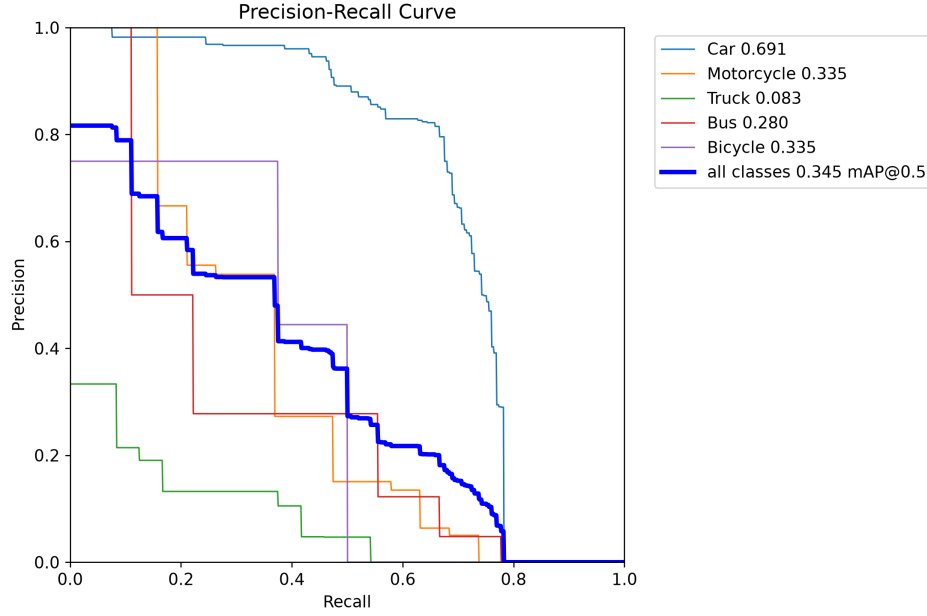


Figure 6.10: Precision–Recall curve for YOLOv8.

The Precision–Recall analysis demonstrates variation in detection performance across vehicle categories. Cars achieved the highest Average Precision values, while minority classes exhibited lower detection performance. The overall mAP@50 of approximately 34.5% confirms that the detector provides usable traffic observations for intelligent traffic management, although additional optimisation could further improve detection reliability.

6.3 Reinforcement Learning Behaviour

Learning performance was analysed using the cumulative reward generated throughout the simulation. Because the reward function is based on queue-length penalties, cumulative reward provides an indication of how effectively the controller learns traffic signal control policies over time.

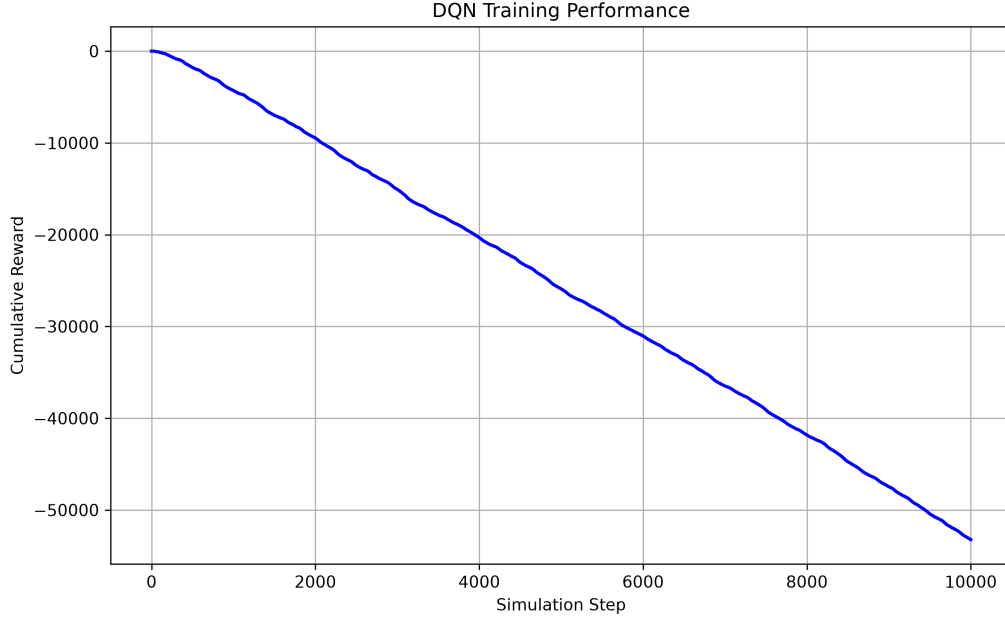


Figure 6.11: DQN reward behaviour during simulation.

Figure 6.11 illustrates the cumulative reward obtained by the DQN controller. Although reward values decrease throughout the simulation due to the negative queue-based reward formulation, the overall trend demonstrates stable learning behaviour. Compared with the fixed-time controller, the DQN framework accumulated fewer congestion-related penalties, indicating that adaptive signal-control actions were more effective at managing traffic demand. These results suggest that the reinforcement learning agent successfully learned control policies capable of reducing congestion and improving intersection performance.

6.4 Queue Length Analysis

Queue length was selected as a key congestion indicator because it directly reflects the accumulation of vehicles at signalised approaches.



Figure 6.12: Queue length variation under DQN traffic signal control.



Figure 6.13: Queue length variation under fixed-time traffic signal control.

Figures 6.12 and 6.13 compare queue-length behaviour under adaptive DQN control and conventional fixed-time control. The DQN controller maintained a more stable queue profile and reduced prolonged congestion periods. Although temporary queue growth occurred during high-demand conditions, queue dissipation was generally faster under adaptive control than under fixed-time operation. These observations indicate that the

DQN controller responded more effectively to changing traffic conditions and allocated green time more efficiently than the baseline controller.

6.5 Comparative Evaluation

Table 6.2 summarises the performance of the proposed DQN framework and the fixed-time controller.

Table 6.2: Comparison Between Fixed-Time Control and the Proposed DQN Framework

Metric	Fixed-Time	DQN Framework	Improvement (%)
Average Waiting Time (s)	14.28	2.80	80.40
Average Queue Length	6.14	5.32	13.36
Throughput	1315	1205	-8.37

The proposed DQN framework achieved substantial improvements in congestion-related performance measures. Average waiting time decreased from 14.28 seconds under fixed-time control to 2.80 seconds, corresponding to an improvement of 80.40%. Similarly, average queue length was reduced from 6.14 vehicles to 5.32 vehicles, representing a reduction of 13.36%. Although throughput was slightly lower than that achieved by the fixed-time controller, the adaptive framework prioritised congestion reduction and traffic-flow stability. Consequently, the reduction in throughput may be interpreted as a trade-off between minimising congestion and maximising vehicle discharge rates.

6.6 Emergency Vehicle Evaluation

The emergency vehicle prioritisation mechanism was evaluated using travel-time and delay measurements.

Table 6.3: Emergency Vehicle Performance

Metric	Value
Average EV Travel Time (s)	21.05
Average EV Delay (s)	11.05

Results indicate that emergency vehicles required an average of 21.05 seconds to complete their journeys through the simulated road network. Relative to the selected benchmark travel time, an average delay of 11.05 seconds was observed. Visual inspection of simulation behaviour demonstrated that signal pre-emption and green-extension mechanisms successfully reduced interruptions experienced by emergency vehicles. However, temporary increases in queue length occasionally occurred on non-prioritised approaches, highlighting the trade-off between emergency vehicle priority and overall traffic fairness.

6.7 Achievement of Research Objectives

The evaluation results presented throughout this chapter were used to assess the extent to which the research objectives defined in Chapter 1 were achieved.

Objective 1 required a critical review of existing research relating to adaptive traffic signal control, reinforcement learning, computer vision, and emergency vehicle prioritisation. This objective was achieved through the literature review presented in Chapter 2, which identified current limitations and established the research gap addressed by the proposed framework.

Objective 2 focused on the development of an intelligent traffic management framework integrating YOLOv8-based vehicle detection, Deep Reinforcement Learning, and emergency vehicle prioritisation within the SUMO environment. This objective was successfully achieved through the design and implementation of the complete framework described in Chapters 4 and 5.

Objective 3 required evaluation of the proposed framework using quantitative performance indicators and comparison with conventional fixed-time traffic signal control. This objective was achieved through the experimental analysis presented in this chapter, using vehicle detection metrics, congestion-related performance measures, and emergency vehicle evaluation results.

Overall, the findings indicate that all research objectives were successfully addressed and the proposed framework was capable of supporting both adaptive traffic management and emergency vehicle prioritisation within a unified system.

6.8 Chapter Summary

This chapter evaluated the proposed intelligent traffic management framework using both computer vision and traffic-engineering performance measures. The YOLOv8 detector, Deep Q-Network controller, and emergency vehicle prioritisation mechanism were assessed individually and collectively using quantitative evaluation metrics. The results demonstrated that the framework was capable of generating useful traffic-state information, reducing congestion-related indicators, and supporting emergency vehicle movement. In comparison with conventional fixed-time traffic signal control, the proposed framework achieved improvements in traffic management performance while maintaining operational effectiveness within the SUMO simulation environment. The findings presented in this chapter provide the basis for the conclusions, limitations, and recommendations discussed

in Chapter 7.

Chapter 7

Conclusions and Recommendations

This chapter presents the overall conclusions drawn from the research and discusses the extent to which the project objectives were achieved. The chapter summarises the main findings, highlights the contributions of the study, outlines its limitations, and provides recommendations for future research. The purpose of this chapter is to evaluate the significance of the developed intelligent traffic management framework and to position the work within the broader context of intelligent transportation systems.

7.1 Research Summary

Traffic congestion continues to present significant challenges for urban transportation systems, contributing to increased travel times, environmental impacts, and reduced network efficiency. Conventional fixed-time traffic signal controllers often struggle to respond effectively to changing traffic conditions, while emergency vehicles may experience delays at signalised intersections. The aim of this research was to develop and evaluate an intelligent traffic management framework capable of combining traffic perception, adaptive traffic signal control, and emergency vehicle prioritisation within a unified system.

To achieve this objective, the study integrated YOLOv8-based vehicle detection, a Deep Q-Network (DQN) traffic signal controller, emergency vehicle priority mechanisms, and the SUMO simulation environment. A dedicated traffic perception layer was developed to detect both ordinary vehicles and emergency vehicles and generate traffic-state information for decision-making. The adaptive controller used these observations to optimise traffic signal operations dynamically, while the emergency vehicle module provided priority handling through signal pre-emption and green-phase extension strategies. Experimental evaluation demonstrated that the proposed framework was capable of reducing congestion-related performance indicators when compared with conventional fixed-time traffic signal control.

The results also confirmed the feasibility of integrating computer vision, reinforcement learning, and emergency vehicle management within a single intelligent transportation framework. Overall, the research demonstrates how traffic perception and adaptive decision-making can be combined to support more responsive and efficient traffic management systems.

7.2 Key Findings

Several important findings emerged from the evaluation of the proposed framework. First, the YOLOv8-based perception layer successfully detected ordinary and emergency vehicles and generated traffic-state information suitable for adaptive traffic signal control. Although the achieved detection performance indicates opportunities for further improvement, the model provided sufficient perception capability to support the decision-making process. Second, the Deep Q-Network (DQN) controller demonstrated the ability to learn adaptive traffic signal control policies through interaction with the SUMO environment. The reinforcement learning agent responded dynamically to changing traffic conditions and achieved lower congestion-related penalties than the fixed-time controller. Third, the proposed framework reduced average waiting time and queue length when compared with conventional fixed-time traffic signal control. These improvements indicate that adaptive traffic signal management can respond more effectively to fluctuating traffic demand than predefined timing plans. Finally, the emergency vehicle prioritisation mechanism successfully supported emergency vehicle movement through signalised intersections. Signal pre-emption and green-phase extension reduced interruptions experienced by priority vehicles while maintaining acceptable traffic operation across the network.

Collectively, these findings demonstrate the practical benefits of combining traffic perception, adaptive signal optimisation, and emergency vehicle management within a unified intelligent transportation framework.

7.3 Research Contributions

This research contributes to the field of Intelligent Transportation Systems in several ways. First, the study presents a unified framework that integrates computer vision, reinforcement learning, and emergency vehicle prioritisation within a single traffic management architecture. While many existing studies focus on either traffic perception or traffic optimisation, the proposed framework combines both capabilities and enables interaction between them. Second, the research demonstrates the implementation of a

Deep Q-Network (DQN) controller for adaptive traffic signal control using traffic-state information generated from a perception layer. This approach contributes to reducing the separation that often exists between traffic monitoring and traffic control systems. Third, emergency vehicle recognition and prioritisation were incorporated directly into the traffic management process. Instead of operating as a completely independent sub-system, emergency vehicle handling was integrated with adaptive traffic signal control to support more coordinated traffic management decisions. A further contribution involves the development of supporting software components for dataset preparation, annotation, vehicle detection, and traffic simulation. These tools facilitated the development of the proposed framework and may provide a useful foundation for future intelligent transportation research.

The overall framework addresses several research gaps identified in the literature, particularly the limited integration of real-time traffic perception, adaptive signal optimisation, and emergency vehicle management within a single intelligent transportation system [44, 48].

7.4 Limitations

Although the proposed framework achieved its intended objectives, several limitations should be acknowledged. The most significant limitation relates to the use of simulation. While SUMO provides a realistic and widely adopted traffic simulation environment, it cannot fully replicate the complexity of real-world transportation systems. Factors such as sensing errors, communication delays, environmental conditions, and unpredictable driver behaviour remain difficult to reproduce accurately within a simulation environment [61], [62].

A second limitation concerns the vehicle detection dataset. The detector was trained using a relatively modest dataset when compared with large-scale industrial object-detection applications. Previous research has shown that larger and more diverse datasets generally improve model robustness and generalisation performance [60], [59]. Consequently, additional data collection may further improve detection accuracy. Another limitation relates to the reinforcement learning controller. The implemented DQN architecture was intentionally designed to remain computationally lightweight and suitable for the scope of this research.

More advanced approaches incorporating experience replay, target networks, Double DQN, Dueling DQN, or other deep reinforcement learning techniques may provide improved learning stability and traffic optimisation performance [57], [13]. Finally, evalua-

tion was conducted using a single-intersection traffic scenario. Although this environment was sufficient for demonstrating the feasibility of the proposed framework, larger multi-intersection road networks may exhibit different traffic dynamics and require additional coordination mechanisms.

7.5 Recommendations for Future Work

Although the proposed framework demonstrated encouraging performance, several opportunities exist for further development. One potential enhancement involves the use of larger and more diverse training datasets containing additional vehicle categories, weather conditions, camera viewpoints, and traffic scenarios. Increasing dataset diversity may improve vehicle detection robustness and generalisation performance. Future research could also investigate more advanced reinforcement learning techniques. While the present study employed a Deep Q-Network (DQN) controller, alternative approaches such as Double DQN, Dueling DQN, Proximal Policy Optimisation (PPO), and Multi-Agent Reinforcement Learning (MARL) have reported promising results in complex traffic environments and may provide improved convergence behaviour and traffic optimisation performance [52], [54].

A further extension would be the evaluation of the framework within larger road networks containing multiple interacting intersections. Previous studies have shown that co-operative and multi-agent control strategies can improve scalability and support network-wide traffic optimisation [55], [52]. Expanding the framework beyond a single intersection would therefore provide a more comprehensive assessment of its suitability for deployment in urban transportation systems. Another promising direction involves integrating live traffic camera feeds and real-time traffic data. Such an approach would enable assessment of the framework under practical operating conditions and help bridge the gap between simulation-based evaluation and real-world deployment [31], [44].

Finally, emergency vehicle management could be enhanced through Vehicle-to-Infrastructure (V2I) communication technologies. Providing traffic controllers with advance information regarding emergency vehicle routes and estimated arrival times may improve prioritisation decisions and further reduce emergency response delays [33], [45].

7.6 Final Conclusion

This research investigated the integration of computer vision, reinforcement learning, and emergency vehicle prioritisation within a unified intelligent traffic management frame-

work. The proposed system combined YOLOv8-based traffic perception, a Deep Q-Network adaptive traffic signal controller, emergency vehicle priority management, and SUMO-based simulation to address key challenges associated with urban traffic congestion and emergency vehicle movement.

The evaluation demonstrated that the framework was capable of generating useful traffic-state information, reducing vehicle waiting times, lowering queue lengths, and supporting emergency vehicle progression through signalised intersections. The results indicate that adaptive decision-making informed by real-time traffic perception can provide measurable benefits when compared with conventional fixed-time traffic signal control. Although further development and large-scale validation are required before practical deployment can be considered, the findings demonstrate the potential of combining computer vision and reinforcement learning within intelligent transportation systems. By integrating traffic perception, adaptive signal optimisation, and emergency vehicle management within a common framework, this research contributes to ongoing efforts aimed at improving the efficiency, responsiveness, and resilience of future smart-city transportation infrastructures.

Chapter 8

Reflection

This chapter reflects on the experiences, challenges, and lessons encountered throughout the completion of the project. While the earlier chapters focused on system design, implementation, and evaluation, this chapter considers the personal, technical, and research-related development that resulted from undertaking the study. Particular attention is given to project management, technical skills, research capabilities, and the practical challenges associated with developing an intelligent traffic management framework.

8.1 Project Management Reflection

One of the most valuable lessons gained during this project was the importance of effective planning and time management. At the outset, the scale of the work involved in conducting the literature review, preparing datasets, developing software components, performing experiments, and writing the dissertation was underestimated. As the project progressed, it became apparent that each stage required considerably more effort and time than originally anticipated. This experience highlighted the importance of breaking complex objectives into smaller, manageable tasks and establishing realistic project milestones.

8.2 Technical Skills Development

The project provided valuable opportunities to strengthen technical skills across several areas of computing and artificial intelligence. Prior to undertaking this research, practical experience with reinforcement learning, computer vision, and intelligent transportation systems was limited. Through the development and evaluation of the proposed framework, a deeper understanding of machine learning techniques, neural network architectures, object detection systems, and traffic simulation platforms was acquired. The

implementation of the YOLOv8 perception layer improved knowledge of image annotation, dataset preparation, object-detection workflows, model training, and performance evaluation. Developing the custom annotation tool also provided practical experience in graphical user interface design and dataset engineering.

The design and implementation of the Deep Q-Network controller significantly enhanced understanding of reinforcement learning concepts, including state representation, reward design, exploration strategies, and neural network optimisation. In addition, the integration of SUMO and TraCI provided experience in simulation-based experimentation and real-time interaction between external control systems and traffic simulation environments. Overall, the project strengthened practical skills in Python development, machine learning, computer vision, reinforcement learning, and intelligent transportation technologies. These competencies are increasingly important within modern intelligent transportation and smart-city applications [13], [44], [61], [62].

8.3 Personal Development

Beyond technical knowledge, the project contributed significantly to personal and professional development. Managing a long-term research project improved organisational skills, self-discipline, and the ability to work independently. Over the course of the study, confidence increased when approaching unfamiliar technologies, technical problems, and research challenges. The project also strengthened critical thinking and decision-making abilities. Many stages of the development process required evaluation of alternative approaches before selecting an appropriate solution. This encouraged a more analytical approach to problem solving and improved the ability to justify technical decisions using evidence from both experimentation and literature. In addition, the experience improved communication and academic writing skills. Translating technical concepts into a structured dissertation required careful organisation, reflection, and clear presentation of findings. These skills are expected to remain valuable in future academic research and professional roles within data science, artificial intelligence, and intelligent systems development.

8.4 Reflection Summary

Overall, the project provided valuable experience across research, software development, machine learning, computer vision, reinforcement learning, and intelligent transportation systems. The challenges encountered throughout the study contributed significantly to both technical and personal growth, while the successful completion of the framework reinforced the importance of planning, adaptability, and continuous learning. The project

not only enhanced technical expertise but also improved research, analytical, and project management capabilities. Although the framework represents a relatively small-scale implementation of intelligent traffic management, the experience has provided a strong foundation for future research and professional development in artificial intelligence and intelligent transportation systems.

Bibliography

- [1] C. Ouyang, Z. Zhan, and F. Lv, “A comparative study of traffic signal control based on reinforcement learning algorithms,” *World Electric Vehicle Journal*, vol. 15, no. 6, 2024. [Online]. Available: <https://www.mdpi.com/2032-6653/15/6/246>
- [2] D. Schrank, B. Eisele, and T. Lomax, “Urban mobility report 2019,” 2019.
- [3] R. Arnott and K. Small, “The economics of traffic congestion,” *American scientist*, vol. 82, no. 5, pp. 446–455, 1994.
- [4] A. Moyano, M. Stepniak, B. Moya-Gómez, and J. C. García-Palomares, “Traffic congestion and economic context: changes of spatiotemporal patterns of traffic travel times during crisis and post-crisis periods,” *Transportation*, vol. 48, no. 6, pp. 3301–3324, 2021.
- [5] M. A. Figliozzi, “The impacts of congestion on commercial vehicle tour characteristics and costs,” *Transportation research part E: logistics and transportation review*, vol. 46, no. 4, pp. 496–506, 2010.
- [6] C. Tan, Y. Cao, X. Ban, and K. Tang, “Connected vehicle data-driven fixed-time traffic signal control considering cyclic time-dependent vehicle arrivals based on cumulative flow diagram,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 8, pp. 8881–8897, 2024.
- [7] P. Varaiya, “Max pressure control of a network of signalized intersections,” *Transportation Research Part C: Emerging Technologies*, vol. 36, pp. 177–195, 2013.
- [8] S. O. SUBAIR, B. A. IBITOYE, and A. T. KURANGA, “Evaluation of traffic congestion in an urban roads: A review,” *ABUAD Journal of Engineering and Applied Sciences*, vol. 2, no. 2, pp. 1–7, 2024.
- [9] J. P. Bharadiya, “Artificial intelligence in transportation systems a critical review,” *American Journal of Computing and Engineering*, vol. 6, no. 1, pp. 35–45, 2023.
- [10] L. Zemmouchi-Ghomari, “Artificial intelligence in intelligent transportation systems,” *Journal of Intelligent Manufacturing and Special Equipment*, vol. 6, no. 1, pp. 26–42, 2025.

- [11] J. Guerrero-Ibañez, J. Contreras-Castillo, and S. Zeadally, “Deep learning support for intelligent transportation systems,” *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 3, p. e4169, 2021.
- [12] R. Abduljabbar, H. Dia, S. Liyanage, and S. A. Bagloee, “Applications of artificial intelligence in transport: An overview,” *Sustainability*, vol. 11, no. 1, p. 189, 2019.
- [13] A. Haydari and Y. Yılmaz, “Deep reinforcement learning for intelligent transportation systems: A survey,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 11–32, 2020.
- [14] K. Shreya, R. Arun, A. Gowtham, and R. Arunkumar, “Safelane: Real-time traffic management and accident detection using yolo and lstm,” in *Proceedings of International Conference on Computer Science and Communication Engineering (ICCSCE 2025)*. Springer Nature, 2025, p. 283.
- [15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.
- [16] R. Sapkota, M. Flores-Calero, R. Qureshi, C. Badgujar, U. Nepal, A. Poullose, P. Zeno, U. B. P. Vaddevolu, S. Khan, M. Shoman *et al.*, “Yolo advances to its genesis: A decadal and comprehensive review of the you only look once (yolo) series,” *Artificial Intelligence Review*, vol. 58, no. 9, p. 274, 2025.
- [17] R. Sapkota, R. Qureshi, M. Flores-Calero, C. Badgujar, U. Nepal, A. Poullose, P. Zeno, U. P. Vaddevolu, H. Yan, and M. Karkee, “Yolov10 to its genesis: A decadal and comprehensive review of the you only look once series,” 2024.
- [18] L. T. Ramos and A. D. Sappa, “A decade of you only look once (yolo) for object detection: A review,” *IEEE Access*, vol. 13, pp. 192 747–192 794, 2025.
- [19] X. Han, J. Chang, and K. Wang, “Real-time object detection based on yolo-v2 for tiny vehicle object,” *Procedia Computer Science*, vol. 183, pp. 61–72, 2021.
- [20] N. M. Alahdal, F. Abukhodair, L. H. Meftah, and A. Cherif, “Real-time object detection in autonomous vehicles with yolo,” *Procedia Computer Science*, vol. 246, pp. 2792–2801, 2024.
- [21] D. P. Carrasco, H. A. Rashwan, M. Á. García, and D. Puig, “T-yolo: Tiny vehicle detection based on yolo and multi-scale convolutional neural networks,” *Ieee Access*, vol. 11, pp. 22 430–22 440, 2021.

- [22] M. J. Shafiee, B. Chywl, F. Li, and A. Wong, “Fast yolo: A fast you only look once system for real-time embedded object detection in video,” *arXiv preprint arXiv:1709.05943*, 2017.
- [23] A. Nazir and M. A. Wani, “You only look once-object detection models: a review,” in *2023 10th International conference on computing for sustainable global development (INDIACom)*. IEEE, 2023, pp. 1088–1095.
- [24] K. Nellore and G. P. Hancke, “A survey on urban traffic management system using wireless sensor networks,” *Sensors*, vol. 16, no. 2, p. 157, 2016.
- [25] M. Eom and B.-I. Kim, “The traffic signal control problem for intersections: a review,” *European transport research review*, vol. 12, no. 1, p. 50, 2020.
- [26] P. Serafini and W. Ukovich, “A mathematical model for the fixed-time traffic control problem,” *European Journal of Operational Research*, vol. 42, no. 2, pp. 152–165, 1989.
- [27] H. Ceylan and M. G. Bell, “Reserve capacity for a road network under optimized fixed time traffic signal control,” *Journal of Intelligent Transportation Systems*, vol. 8, no. 2, pp. 87–99, 2004.
- [28] A. Muralidharan, R. Pedarsani, and P. Varaiya, “Analysis of fixed-time control,” *Transportation Research Part B: Methodological*, vol. 73, pp. 81–90, 2015.
- [29] S. Mahato, “Smart traffic signals: Comparing marl and fixed-time strategies,” *arXiv preprint arXiv:2505.14544*, 2025.
- [30] A. Janota, F. Kalus, R. Pirník, J. Kafková, P. Kuchár, M. Skuba, and P. Holečko, “Reinforcement learning approach to adaptive traffic signal control using sumo-rl,” in *2024 25th International Carpathian Control Conference (ICCC)*. IEEE, 2024, pp. 01–06.
- [31] A. Omari Alaoui, M. Oumoulylte, H. Oubalahcen, M. K. Boutahir, A. Hessane, and A. El Allaoui, “Real-time traffic signal adjustment using yolov8 for improved integration of emergency vehicles in smart traffic systems,” *Signal, Image and Video Processing*, vol. 19, no. 7, p. 576, 2025.
- [32] R. R. Raj, R. Shanker, and M. Maranco, “Dynamic ai traffic signal system for real-time traffic management using pygame and yolo v8,” in *Proceedings of the 1st International Conference on Research and Development in Information, Communication, and Computing Technologies (ICRDICCT 2025)*. SCITEPRESS, 2025, pp. 300–310.

- [33] H. Su, Y. D. Zhong, J. Y. Chow, B. Dey, and L. Jin, “Emvligh: A multi-agent reinforcement learning framework for an emergency vehicle decentralized routing and traffic signal control system,” *Transportation Research Part C: Emerging Technologies*, vol. 146, p. 103955, 2023.
- [34] K. J. Kumar, S. J. Arul, S. Kurse, B. Smitha, M. Vikas, S. Vedanth, C. S. Reddy, and S. Salman, “A review on smart traffic control system using modern approaches,” in *AIP Conference Proceedings*, vol. 2965, no. 1. AIP Publishing LLC, 2024, p. 040012.
- [35] A. S. Bale, V. Narayanaswamy, V. Y. Shanthakumar, P. B. Shyla, S. Balakrishna, V. S. Nagaraja, S. B. Menasinakai, M. K. Hitesh, and E. Esarapu, “Recent advancement in emergency vehicle communication system using iot,” in *IoT and Big Data Analytics for Smart Cities*. Chapman and Hall/CRC, 2022, pp. 121–158.
- [36] J. Kaur, “Priority vehicle management system for urgency based services using rfid technology,” *The Online Journal of Distance Education and e-Learning*, vol. 11, no. 2, 2023.
- [37] M. A. Fera, R. Aswini, M. Santhiya, K. G. Deepa, and M. Thangaprabha, “Health monitoring in emergency vehicles with their authentication by rfid and location tracking by gps,” in *2015 Seventh International Conference on Advanced Computing (ICoAC)*. IEEE, 2015, pp. 1–6.
- [38] A. Dani, D. Rane, S. Bansal, and A. More, “Rfid based automatic pathway for emergency vehicles,” in *2025 IEEE International Students’ Conference on Electrical, Electronics and Computer Science (SCEECS)*. IEEE, 2025, pp. 1–5.
- [39] O. Avatefipour and F. Sadry, “Traffic management system using iot technology—a comparative review,” in *2018 IEEE International Conference on Electro/Information Technology (EIT)*. IEEE, 2018, pp. 1041–1047.
- [40] Y.-L. Lai, Y.-H. Chou, and L.-C. Chang, “An intelligent iot emergency vehicle warning system using rfid and wi-fi technologies for emergency medical services,” *Technology and health care*, vol. 26, no. 1, pp. 43–55, 2018.
- [41] U. Cedric, “Smart school bus monitoring and notification system using rfid and gps,” Master’s thesis, University of Rwanda, 2021.
- [42] A. TESFAYE, “The practice and challenges in implementing gps and rfid technologies: The case of container and cargo handling in erca,” Ph.D. dissertation, St. Mary’s University, 2018.
- [43] F. V. Webster, “Traffic signal settings,” Tech. Rep., 1958.

- [44] P. Michailidis, I. Michailidis, C. R. Lazaridis, and E. Kosmatopoulos, “Traffic signal control via reinforcement learning: A review on applications and innovations,” *Infrastructures*, vol. 10, no. 5, p. 114, 2025.
- [45] L. Zhong and Y. Chen, “A novel real-time traffic signal control strategy for emergency vehicles,” *IEEE Access*, vol. 10, pp. 19 481–19 492, 2022.
- [46] S. Phon-Amnuaisuk, S.-J. Tan, Y.-H. Yap, F. C.-M. Choong, P. Vejjanugraha, K.-C. Khor, and K.-H. Ng, “Multi-agent traffic light controls with sumo,” in *2023 8th International Conference on Business and Industrial Research (ICBIR)*, 2023, pp. 824–830.
- [47] L. Li, Y. Lv, and F.-Y. Wang, “Traffic signal timing via deep reinforcement learning,” *IEEE/CAA Journal of Automatica Sinica*, vol. 3, no. 3, pp. 247–254, 2016.
- [48] Y. Zhu, Y. Lv, S. Lin, and J. Xu, “A stochastic traffic flow model-based reinforcement learning framework for advanced traffic signal control,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 26, no. 1, pp. 714–723, 2025.
- [49] B. Abdulhai, R. Pringle, and G. J. Karakoulas, “Reinforcement learning for true adaptive traffic signal control,” *Journal of transportation engineering*, vol. 129, no. 3, pp. 278–285, 2003.
- [50] H. Wei, G. Zheng, H. Yao, and Z. Li, “Intellilight: A reinforcement learning approach for intelligent traffic light control,” in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 2496–2505.
- [51] H. Wei, C. Chen, G. Zheng, K. Wu, V. Gayah, K. Xu, and Z. Li, “Presslight: Learning max pressure control to coordinate traffic signals in arterial network,” in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 1290–1298.
- [52] S. Wang and S. Wang, “A novel multi-agent deep rl approach for traffic signal control,” *arXiv preprint arXiv:2306.02684*, 2023.
- [53] M. T. Rafique, A. Mustafa, and H. Sajid, “Adaptive traffic signal control using reinforcement learning,” *arXiv preprint arXiv:2408.15751*, 2024.
- [54] D. Kleczyński, J. Drzał, and B. Pawłowicz, “Adaptive traffic signal control using proximal policy optimization transformer and radio frequency identification-based vehicle detection in the simulation of urban mobility environment.” *Advances in Science and Technology Research Journal*, vol. 20, no. 5, pp. 372–385, 2026.

- [55] T. Chu, J. Wang, L. Codecà, and Z. Li, “Multi-agent deep reinforcement learning for large-scale traffic signal control,” *IEEE transactions on intelligent transportation systems*, vol. 21, no. 3, pp. 1086–1095, 2019.
- [56] R. S. Sutton, A. G. Barto *et al.*, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.
- [57] V. Mnih *et al.*, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
- [58] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1, no. 2.
- [59] G. Jocher, A. Chaurasia, and J. Qiu, “Ultralytics yolov8,” <https://github.com/ultralytics/ultralytics>, 2023.
- [60] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “Yolov4: Optimal speed and accuracy of object detection,” *arXiv preprint arXiv:2004.10934*, 2020.
- [61] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, “Sumo—simulation of urban mobility: an overview,” in *Proceedings of SIMUL 2011, the third international conference on advances in system simulation*. ThinkMind, 2011.
- [62] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, “Microscopic traffic simulation using sumo,” in *2018 21st international conference on intelligent transportation systems (ITSC)*. Ieee, 2018, pp. 2575–2582.
- [63] D. Krajzewicz, G. Hertkorn, C. Rössel, and P. Wagner, “Sumo (simulation of urban mobility)-an open-source traffic simulation,” in *Proceedings of the 4th middle East Symposium on Simulation and Modelling (MESM20002)*, 2002, pp. 183–187.
- [64] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, “Sumo—simulation of urban mobility: an overview,” in *Proceedings of SIMUL 2011, the third international conference on advances in system simulation*. ThinkMind, 2011.
- [65] A. Wegener, M. Piórkowski, M. Raya, H. Hellbrück, S. Fischer, and J.-P. Hubaux, “Traci: an interface for coupling road traffic and network simulators,” in *Proceedings of the 11th communications and networking simulation symposium*, 2008, pp. 155–163.
- [66] H. Arbale and D. N. Mutisya, “Book review: ”research methods for business students” (eighth edition) by mark n. k. saunders, philip lewis, and adrian thornhill (pearson education, 2019),” *African Quarterly Social*

- Science Review*, vol. 1, no. 2, p. 8–21, Apr. 2024. [Online]. Available: <https://quarterlyreview.net/ojs/index.php/aqssr/article/view/8>
- [67] J. W. Creswell and J. D. Creswell, *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications, 2017.
- [68] M. Boneh, “Vehicle detection using deep learning and yolo algorithm,” <https://github.com/MaryamBoneh/Vehicle-Detection>, 2021, accessed 11 February 2023.
- [69] Department for Transport, “Road traffic statistics,” 2026, accessed 2026. [Online]. Available: <https://roadtraffic.dft.gov.uk/>
- [70] M. Bakirci, “Enhancing vehicle detection in intelligent transportation systems via autonomous uav platform and yolov8 integration,” *Applied Soft Computing*, vol. 164, p. 112015, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1568494624007890>

Appendix A

Source Code Samples

The purpose of this appendix is to provide representative code examples from the implemented framework. Full source code listings are omitted for brevity; however, the selected examples illustrate the principal components of the proposed intelligent traffic management system.

A.1 YOLOv8 Detection Implementation

The following code fragment demonstrates the loading of a trained YOLOv8 model and the execution of object detection on input images.

```
from ultralytics import YOLO
model = YOLO("best.pt")
results = model.predict( source=image_path, conf=0.25 )
```

A.2 Traffic State Extraction

Traffic-state information is generated using lane detector measurements obtained through the SUMO–TraCI interface.

```
state = [ queue_eb0, queue_eb1, queue_eb2, queue_sb0, queue_sb1,
↪ queue_sb2, current_phase ]
```

A.3 DQN Action Selection

The reinforcement learning controller applies an epsilon-greedy policy when selecting traffic signal actions.

```

if np.random.rand() <= epsilon:

    action = random.randrange(action_size)

else:
    action = np.argmax( model.predict(state)[0] )

```

A.4 Emergency Vehicle Prioritisation

The emergency vehicle module grants signal priority when an emergency vehicle is detected.

```

if vehicle_type == "ambulance":
    activate_priority()

elif vehicle_type == "fire":
    activate_priority()

elif vehicle_type == "police":
    activate_priority()

```