Mohamed Khider University
of Biskra Faculty of Science
and Technology Department
of Electrical Engineering

MASTER THESIS

Science and Technology Electronics
Embedded Systems

Réf.:

Submitted and defended by:
Sahraoui Abdennour
Ben Zin Abdelbaset

On: 19 June 2025

Smart Traffic Light System

Board of Examiners:

Pr.	Hamaizia Zahra	Professor	University of Biskra	President
Dr.	Fedias Meriem	MCB	University of Biskra	Examiner
Dr.	Medouakh Saadia	MCB	University of Biskra	Supervisor

Academic year: 2024 – 2025

Mohamed Khider University of
Biskra Faculty of Science and
Technology Department of
Electrical Engineering

MASTER THESIS

Science and Technology Electronics
Embedded Systems

Réf.:

Submitted and defended by:

Sahraoui Abdennour & Ben Zin Abdelbaset

On: 19 June 2025

Smart Traffic Light System

Presented by:	**Favorable opinion of the supervisor:**
Sahraoui Abdennour	**Dr. MEDOUAKH Saadia**
Ben Zin Abdelbaset	

Favorable opinion of the jury president

………………..

Stamp and signature

Acknowledgment

I thank God for granting me the strength to accomplish this work and move forward.

I would also like to express my sincere thanks and deep gratitude to my beloved family, who have been a constant source of support and strength throughout my studies.

I would like to thank everyone who contributed to the success of my internship and helped me in writing this thesis.

I express my deepest appreciation to my thesis supervisor, Mrs. Saadia Madouakh, for her supervision, guidance, assistance, and valuable advice.

I also extend my heartfelt thanks to all the professors, and to everyone who has guided me through their words, writings, advice, and constructive criticism.

﴿وَقُلْ رَبِّ زِدْنِي عِلْمًا﴾

— سورة طه، الآية 114 —

To the ones whose prayers were the foundation of our success,

whose hands toiled to build our future...

We dedicate this work to our beloved parents, symbols of unconditional love and sacrifice.

To our dear brothers and sisters, our constant source of support and joy this achievement is as much yours as it is ours.

To our friends, who shared with us moments of hardship and triumph, thank you for your shared effort and dedication.

To our esteemed professors, who generously shared their knowledge and guidance,

and to our supervisor your support and insight have meant the world to us.

With deepest gratitude and respect

Abstract:

Rapid urbanization has significantly increased the number of vehicles on city roads, causing persistent issues such as congestion, accidents, and environmental pollution. Traditional traffic management systems, reliant on fixed-time signaling, lack the responsiveness required to effectively handle dynamic urban traffic conditions, resulting in inefficiencies and exacerbating these problems. In this work, we proposed and implemented an Intelligent Traffic Signal System (ITSS) that uses advanced technological solutions, including deep learning (specifically YOLOv8), object tracking methods (SORT), and adaptive decision-making algorithms that leverage real-time traffic data collected through diverse sensors, such as inductive loops, cameras, radar, and LiDAR. By dynamically adjusting signal timings based on vehicle counts, waiting times, and emergency vehicle detection, ITSS demonstrates substantial improvements in traffic flow efficiency, congestion mitigation, and accident prevention. The practical application and evaluation detailed within highlight the advantages of this intelligent system, notably in enhancing environmental sustainability through reduced vehicle emissions, improving economic productivity by minimizing travel delays, and ensuring public safety by prioritizing emergency response. However, research in the field of ITSS acknowledges critical challenges including initial infrastructure investment, sensor reliability, data privacy concerns, and public acceptance. The findings underscore that despite these hurdles, intelligent traffic management represents an essential evolution for sustainable urban development, advocating for continued innovation and implementation of ITSS within urban environments.

Keywords: Intelligent Traffic Signal System (ITSS), object detection, YOLOv8, SORT, SUMO.

الملخص:

أدى التوسع الحضري السريع إلى زيادة ملحوظة في عدد المركبات على طرق المدن، مما تسبب في مشاكل مستمرة مثل الازدحام والحوادث والتلوث البيئي. تفتقر أنظمة إدارة المرور التقليدية، المعتمدة على إشارات زمنية ثابتة، إلى الاستجابة اللازمة للتعامل بفعالية مع ظروف حركة المرور الحضرية الديناميكية، مما يؤدي إلى انخفاض الكفاءة وتفاقم هذه المشاكل. في هذا العمل، اقترحنا وطبقنا نظام إشارات مرور ذكي (ITSS) يستخدم حلولاً تكنولوجية متقدمة، بما في ذلك التعلم العميق (وتحديدًا YOLOv8)، وطرق تتبع الأجسام (SORT)، وخوارزميات اتخاذ القرار التكيفية التي تستفيد من بيانات حركة المرور اللحظية التي يتم جمعها من خلال أجهزة استشعار متنوعة، مثل الحلقات الاستقرائية والكاميرات والرادار والليدار. من خلال ضبط توقيتات الإشارات ديناميكيًا بناءً على عدد المركبات وأوقات الانتظار واكتشاف مركبات الطوارئ، يُظهر نظام ITSS تحسينات كبيرة في كفاءة تدفق حركة المرور وتخفيف الازدحام والوقاية من الحوادث. يُسلّط التطبيق العملي والتقييم المفصلان هنا الضوء على مزايا هذا النظام الذكي، لا سيما في تعزيز الاستدامة البيئية من خلال خفض انبعاثات المركبات، وتحسين الإنتاجية الاقتصادية من خلال تقليل تأخيرات السفر، وضمان السلامة العامة من خلال إعطاء الأولوية للاستجابة للطوارئ. ومع ذلك، تُقرّ الأبحاث في مجال أنظمة دعم الحركة المرورية الذكية بوجود تحديات حرجة، بما في ذلك الاستثمار الأولي في البنية التحتية، وموثوقية أجهزة الاستشعار، ومخاوف خصوصية البيانات، والقبول العام. وتُؤكد النتائج أنه على الرغم من هذه العقبات، تُمثّل إدارة المرور الذكية تطورًا أساسيًا للتنمية الحضرية المستدامة، مما يدعو إلى مواصلة الابتكار وتطبيق أنظمة دعم الحركة المرورية الذكية في البيئات الحضرية.

الكلمات المفتاحية: نظام إشارات المرور الذكي(ITSS) ، كشف الأجسام، YOLOv8، SORT، SUMO.

Summary

List of Figures:

List of tables:

List of abbreviations:

AI: Artificial Intelligence

DL: Deep Learning

RL: Reinforcement Learning

ITS: Intelligent Transportation System

ITSS: Intelligent Traffic Signal System

YOLO: You Only Look Once

CNN: Convolutional Neural Network

RNN: Recurrent Neural Network

LSTM: Long Short-Term Memory

SORT: Simple Online and Realtime Tracking

SUMO: Simulation of Urban Mobility

PLC: Programmable Logic Controller

DSRC: Dedicated Short-Range Communication

V2I: Vehicle-to-Infrastructure

C-V2X: Cellular Vehicle-to-Everything

GAN: Generative Adversarial Network

GPS: Global Positioning System

GUI: Graphical User Interface

API: Application Programming Interface

NB-IoT: Narrowband Internet of Things

IoT: Internet of Things

OID: Open Image Dataset

Wi-Fi: Wireless Fidelity

CV: Computer Vision

ML: Machine Learning

FPS: Frames Per Second

VS Code: Visual Studio Code

Colab: Google Colaboratory

General Introduction:

The rapid urbanization and continuous growth of populations globally have significantly increased the volume of vehicles on urban roadways, leading to persistent challenges such as traffic congestion, higher rates of road accidents, prolonged travel times, and increased environmental pollution. Traditional traffic management systems, characterized by fixed-time and predetermined signaling schemes, often fail to adapt effectively to dynamic real-time road conditions, exacerbating these issues and contributing to inefficiencies and safety hazards. As cities evolve into complex ecosystems, the demand for more intelligent, responsive, and adaptive traffic management solutions has become imperative.

To address these pressing urban challenges, Intelligent Traffic Signal Systems (ITSS) have emerged as innovative solutions within the broader framework of smart cities. These advanced systems integrate cutting-edge technologies such as deep learning algorithms, sensor networks, real-time data analytics, and artificial intelligence-driven decision-making processes. Unlike traditional methods, ITSS dynamically adjusts signal timings and sequences based on real-time traffic data collected through advanced sensing technologies, including inductive loops, cameras, radar, and LiDAR. Furthermore, intelligent decision-making mechanisms utilize sophisticated computational techniques, such as Reinforcement Learning, Deep Learning, and hybrid rule-based algorithms, to analyze current conditions and predict future traffic patterns, enabling proactive management and optimal traffic flow.

Variety techniques in traffic signal control have been developed for improved decision-making and adaptability. Recent developments in this field increasingly rely on reinforcement learning (RL) algorithms, fuzzy logic, rule-based approach and hybrid approaches. In addition, many techniques are proposed to detect traffic density to control traffic signals.

This memorandum aims to propose and implement an Intelligent Traffic Signal System (ITSS) was designed to manage traffic flow at intersections using real-time video cameras placed at each lane to monitor traffic continuously analysis and automated control logic. Using the YOLOv8 deep learning model, our system detects and counts vehicles and real-time tracking with SORT algorithms; based on vehicle count, waiting time, and the presence of emergency vehicles, a decision algorithm selects the appropriate signal phase. It prioritizes urgent cases, dynamically adjusts green light durations, and optimizes traffic flow using a hybrid rule-based logic. This integrated approach allows for efficient, responsive, and intelligent traffic control.

We have chosen to organize our study around three main chapters as follows:

- The first chapter provides an overview of traditional traffic light systems, highlighting the need for their improvement. It introduces smart traffic light systems and the Intelligent Transportation System (ITS), and reviews various artificial intelligence approaches in traffic management. Additionally, it presents notable international case studies on AI-based traffic optimization.
- Chapter Two introduces the concept of an Automatic Traffic Light Control System proposed as a key component in smart cities. It discusses the system architecture, the types of data and sensing technologies used, and the communication methods that enable real-time decision-making.
- Chapter Three presents the implementation and evaluation of our Intelligent Traffic Signal System (ITSS). We begin by describing the training of the YOLOv8 deep learning model for vehicle detection, followed by the integration of our system in a simulated environment using SUMO and real-time detection algorithms.

We will end this dissertation with a general conclusion and the perspectives.

CHAPTER1:

Overview of Smart Traffic Systems

1.1 Introduction:

The increasing population and urbanization have resulted in the rapid growth of vehicles on the roads, leading to an increase in traffic congestion, accidents, and pollution. Traffic congestion not only results in the wastage of time but also causes economic losses. Inefficient traffic management systems are one of the primary contributors to these problems. Traditional traffic light systems operate based on predetermined schedules and do not consider the real-time traffic flow, leading to delays, congestion, and accidents. Therefore, there is a need for a smarter and more efficient traffic management system that can adapt to changing traffic conditions and optimize traffic flow **[1].** The answer to the traffic problem is through technology, mainly the automated traffic light system.

In this chapter, we present an overview of the traditional traffic light systems and the importance of improving them. In sections 2 and 3, we describe the smart traffic light systems and the intelligent transportation system ITS. We review several approaches for artificial intelligence in traffic management, in section 4. Finally, we present case studies of AI-based traffic optimization from around the world, illustrating how AI technologies are transforming urban traffic management.

1.2 Traditional Traffic Light Systems:

1.2.1 Definition:

Traffic lights, also known as traffic signals, are visual signals placed at intersections or pedestrian crossings to control the flow of vehicles and pedestrians (figure 1.1). Their primary purpose is to allocate the right-of-way and prevent conflict between different traffic streams. Traffic lights work on a simple principle of color-coded signals – red, yellow, and green – that appear in a specific sequence. These signals are powered by an electrical system and controlled by a fixed timing mechanism. **[2]**

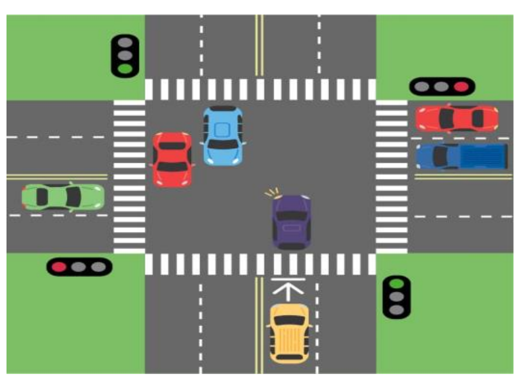


Figure 1.1: Traditional Traffic Light Systems

1.2.2 History:

We encounter traffic lights every day without giving them much thought, yet their origins date back over a century, born out of a need to solve a basic problem: road chaos. Up next, we explore the fascinating evolution of traffic lights through the years **[3]:**

December 10, 1868— The first gas-lit traffic lights were installed outside the Houses of Parliament in London. Proposed by British railway engineer J.P. Knight to control the traffic of horse carriages, gas lights were manually controlled by a police officer using semaphore arms. The lights became a safety hazard as they sometimes exploded and injured police officers (Figure 1.2).

Figure 1.2: The world's first traffic signal installed near the Houses of Parliament.

1912 —A traffic control device was placed on top of a tower in Paris at the Rue Montmartre and Grande Boulevard, with a revolving four-sided metal box on top of a glass showcase where the word "Stop" was painted in red and the word "Go" painted in white.

1914 — As automobile traffic increased, American policeman Lester Wire designed the first electric traffic light. It was first installed in Cleveland, Ohio, on August 5, 1914.

1917 — First interconnected traffic signal system installed in Salt Lake City, with six connected intersections controlled simultaneously from a manual switch.

1920 — William Potts, a Detroit policeman, invented the first four-way and three-colored traffic lights. He introduced yellow lights to indicate the light would change soon. Detroit became the first city to implement the four-way and three-colored traffic lights.

1923 — Garrett Morgan received a patent for an electric traffic signal. The African American inventor owned a sewing machine company in Cleveland and, after witnessing a horrific accident, worked on his automated traffic signal system. GE paid him $40,000 for the invention.

1928 — Charles Adler Jr. developed a sonically actuated traffic light. To operate it, drivers pulled up to a red light and honked their horns to make the light change. Installed in Baltimore, it was the first actuated traffic signal in the United States and served as the basis for modern traffic signals.

1929 — Adler also invented a pedestrian push button, which was installed in Baltimore—the first pedestrian-actuated signal.

1950s — Computerized detection used in traffic lights. A pressure plate was placed at intersections so computers would know that a car was waiting at the red light.

1960s — As computers improved, they could monitor traffic and change lights in an even more efficient way.

1990s — The countdown timer was introduced to traffic lights to help pedestrians know whether they have enough time to cross the road before the signal changes color.

2010s — Connected vehicles can communicate with traffic signals and other vehicles. This can vastly improve speed, timing, and efficiency at intersections—perhaps as much as 40 percent as more vehicles get connected, according to Washington State University research.

1.2.3 Main Components:

Traffic signals are made up of four essential parts—signal heads, controller units, detectors, and timing mechanisms—that together keep traffic moving smoothly and safely. Let’s dive in, beginning with the signal head.

1.2.3.1 Signal Heads:

Signal heads, also known as traffic lights, provide visual representation to drivers by illuminating lights in three different colors: red, yellow, and green. Each color delivers a different message to drivers: red means stop, yellow means prepare to stop and green means proceed **[4].**

1.2.3.2 Controller Unit:

A traffic light controller unit is the core electronic device that manages the operation of traffic signals at intersections. In traditional traffic systems, it controls the timing and sequencing of red, yellow, and green lights to regulate vehicle and pedestrian traffic safely and efficiently. A traffic signal control system consists of different signals directing different routes in and out of an intersection being controlled by a central command controller. The main controller is the brain that takes traffic trends sent by traffic detectors embedded in the road into consideration and then directs the signals. The controller can be programmed to operate on a fixed time or vehicle actuation module **[5].**

- Fixed time traffic signal control system: This involves the traffic lights being programmed to display a certain signal at all roads for the same fixed intervals. For instance, the traffic lights display the green light for the same fixed time despite the traffic volume.
- Dynamic traffic signal control system: Under this module, the traffic signal control system takes into account the vehicle demand volume using the road-embedded detector and adjusts the green light accordingly .in case of jammed roads, it adjusts the timings accordingly to step UP the traffic flows.

1.2.3.3 Detectors:

Traffic light sensors are devices integrated into traffic signal systems that detect the presence, speed, and type of vehicles and pedestrians at intersections. Their primary function is to provide real-time data that allows traffic controllers to adjust signal timings dynamically, ensuring optimal traffic flow. In addition to reducing delays, these sensors also enhance safety by managing the movement of vehicles and pedestrians more efficiently. Several types of sensors are employed in traffic light systems, each with its advantages and limitations. Each sensor type plays a distinct role in enhancing traffic control and is selected based on specific environmental and operational requirements **[6].** The most common types include:

- **Inductive Loop Sensors:** Embedded beneath the roadway, these sensors detect vehicles by measuring changes in magnetic fields.
- **Infrared Sensors:** Use beams of infrared light to detect vehicles by measuring interruptions in the beam.
- **Microwave Sensors:** Utilize radar technology to detect moving objects, offering the advantage of functioning in various weather conditions.
- **Video Detection Systems:** Employ cameras and sophisticated image processing algorithms to monitor traffic and pedestrian movements.

1.2.3.4 Timing Mechanisms

Traditional traffic light systems control the sequence and duration of signal phases (green, yellow, red) using one of three main timing mechanisms **[7]:**

- A fixed-time traffic signal is a signal that uses a timer to change at predetermined intervals
- An actuated traffic signal changes according to traffic movements instead of at fixed intervals of time, using sensors to detect traffic.
- Pre-timed traffic signals are somewhat like fixed-time traffic signals as they have their signal changes set ahead of time.

1.2.4 Signal Sequence and Operation

1.2.4.1 Standard Signal Phases

A traditional traffic light consists of a vertically oriented metal or plastic housing, typically affixed to a pole or overhead structure at intersections. The housing contains a series of colored lights arranged in a vertical configuration with red on top, yellow in the middle, and green on the bottom as it appears in (Figure 1.3). This order is consistent worldwide to avoid confusion. The traffic light follows a standardized color scheme, with three distinct colors, each color conveys a specific message **[8]:**

- **Red Phase**: the red light indicates "stop" to all vehicular traffic approaching the intersection, drivers must come to a complete stop and wait for the light to change.
- **Yellow (Amber):** the yellow light serves as a transitional phase between red and green. It warns drivers that the signal is about to change, signaling them to prepare to stop if they can do so safely or to proceed with caution if they are already within the intersection.
- **Green Phase**: Positioned at the bottom of the signal, the green light signals "go" to vehicular traffic so drivers with a green light can proceed through the intersection with caution.

Figure 1.3: Colors on the traffic light.

Some intersections may also include special signals for:

- **Turn Signals:** Traffic light turn signals are special lights at intersections that control when vehicles can turn left or right safely. They usually appear as arrow-shaped lights in red, yellow (amber), or green. Traffic lights tell drivers and pedestrians what they must do at intersections and along roads. They tell road users when to stop and go, when and how to turn and when to drive with extra caution **[9].**

Figure 1.4: left-turn green arrow is shown with a red light [9]

- **Pedestrian Phases:** Walking Pedestrian ("Walk") shall be displayed only when the corresponding through movement green indications are displayed, or when an all-red period is displayed if special pedestrian phasing is used (such as leading pedestrian intervals or exclusive pedestrian phases). The Walking Pedestrian indication does not necessarily have to be displayed with the green at actuated intersections (where a pushbutton actuation is used) as this approach allows for the use of less vehicular green time during cycles when no pedestrians are waiting to cross **[10].**

Figure 1.5 Pedestrian Signal Phases

1.2.4.2 Fixed-Time Control Systems

A fixed-time traffic signal is a signal that uses a timer to change at predetermined intervals, instead of changing according to traffic movements. They create an organized, predictable traffic pattern in the area they're installed by using an electro-mechanical signal controller that can be adjusted and uses a dial timer to ensure the signal changes at whatever interval a traffic designer decides what is necessary for that intersection **[11].** Some fixed time systems use different preset time intervals for morning rush hour, evening rush hour and other busy times **[12].**

The main advantage of fixed-time traffic signals is their initial costs and maintenance needs are much lower than other traffic signal systems. The downside to them is that they can often end up causing unnecessary delays, leaving vehicles to sit at intersections for long periods of time when no traffic is around. This is why fixed-time signals are mostly used in urban areas where traffic is more constant and heavier **[11].**

1.2.4.3 Actuated Control Systems (Sensor-Based)

An actuated signal is a traffic control signal that makes use of detection to respond to vehicle and/or pedestrian demand. Depending on the number and placement of loops, the operation may be fully-actuated or semi-actuated. Potential advantages include **[10]:**

- Traffic-actuated control may provide maximum efficiency at intersections where fluctuations in traffic cannot be anticipated and programmed using pretimed control.
- Traffic-actuated control may provide maximum efficiency at complex intersections where one or more movements are sporadic or subject to variation in volume.
- Traffic-actuated control may provide maximum efficiency at intersections that are poorly located within progressive pretimed systems. In these locations, interruptions of main road traffic are undesirable and must be held to a minimum frequency and duration.
- Traffic-actuated control may minimize delay during periods of light traffic because no green time is provided to phases where no traffic demand exists.
- Traffic-actuated control may reduce the number of collisions associated with the arbitrary stopping of vehicles.

a. Semi-actuated

In semi-actuated control, detectors are located on the side road approaches and in the left turn lanes of the main road. Semi-actuated control is suitable for use at intersections with heavy traffic volumes on the arterial and relatively light volumes on the side road. The signal rests in green on the main road, changing to the side road only as a result of a vehicle or pedestrian actuation **[10].**

b. Fully-actuated

Fully-actuated control requires detection on all approaches of both the main road and the side road. Fully-actuated operation is suitable for use at **[10]:**

- Intersections where the traffic volumes of the main road and the side road are more or less equal but where the traffic distribution may be sporadic and varying
- Locations where turning volumes are high at times and low at other times
- High speed locations where there is a need to avoid “dilemma zone” problems

1.2.4.4 Signal Timing Parameters

Signal timing parameters are critical settings that determine how traffic signals allocate right-of-way at intersections, directly affecting traffic flow efficiency, safety, and pedestrian movement. According to the FHWA Traffic Signal Timing Manual and other authoritative sources, the main parameters include **[13]:**

1) **Minimum Green Time (MIN):** The shortest duration that a green interval must last for a particular movement or phase. Ensures vehicles and pedestrians have sufficient time to start moving and clear the intersection safely. Consideration must also be given to pedestrian timing. When there are no pedestrian provisions (indications or pushbuttons), the minimum assured green must be equal to the minimum pedestrian timing (walk + pedestrian clearance).
2) **Maximum Green Time (MAX):** The longest allowable green interval for a phase, limiting how long the green can be extended even if vehicles continue to be detected. Prevents excessive delays to conflicting movements and guards against detector failures causing indefinite green extensions. Can have multiple settings for different times of day (e.g., peak vs. off-peak).
3) **Extension (Passage):** The extension (passage) parameter extends the green interval for each vehicle actuation up to the maximum green. The actual passage time parameter (vehicle extension or gap time) is the time that the phase will be extended for each actuation. This setting is the number of seconds required for a vehicle moving at the approach speed to travel from the detector

to the stop line. If the passage interval is too short, quick stops may result as well as terminating the green before the vehicular movement has been adequately served.

4) **Yellow:** The yellow interval follows the green interval at the end of each phase. The yellow interval is also referred to as the "change" interval and controls the duration of the yellow display for that phase. The phase change interval timing advises drivers that their phase has expired and they should: (1) come to a safe stop prior to the stop line, or (2) proceed through the intersection if they are too near the intersection to stop.
5) **Red:** The red clearance interval (also known as the all-red interval) follows the yellow interval of each phase. It must expire before the next phase in sequence can begin. It is normally one to two seconds, but on slower speed approaches, it is not unusual to use a very short duration of 0.0 to 0.5 seconds since the yellow time of 3.0 to 4.0 seconds provides sufficient time to meet both the change and the clearance requirements.

1.2.5 Limitations of Traditional traffic light systems

Traffic lights are an integral part of our transportation systems, providing order and safety at intersections. However, it is essential to acknowledge and address the drawbacks associated with traffic lights. Issues such as traffic congestion, inflexibility, increased travel time, pedestrian safety concerns, and environmental impacts pose challenges that need to be mitigated. Efforts are being made to explore alternative traffic management systems, such as roundabouts, intelligent traffic control systems, and adaptive signal timings, to overcome some of these limitations. By embracing innovative approaches, we can strive for a better balance between efficient traffic flow and improved safety while reducing the negative impacts associated with traditional traffic lights. Ultimately, the goal is to create a harmony between different modes of transportation, promoting sustainable and intelligent systems that meet the needs of a rapidly evolving world **[14].**

1.3 Importance of Improving Traffic Signal Control

Traffic signal control systems play a crucial role in managing the flow of vehicles and pedestrians at intersections. With the increasing number of vehicles on the road, efficient traffic management is essential for ensuring safety, reducing congestion, and improving overall transportation efficiency **[15] [16].**

1.3.1 Improved Traffic Flow

One of the most significant advantages of a traffic signal control system is the improvement in traffic flow. By strategically controlling the timing of red, yellow, and green lights at intersections, these systems can optimize the movement of vehicles and reduce unnecessary delays. Traffic signals help prevent bottlenecks by coordinating the flow of traffic in all directions, allowing for a smoother and more predictable driving experience.

1.3.2 Enhanced Safety

Safety is a primary concern in any traffic management system, and traffic signal control systems are instrumental in reducing accidents at intersections. By clearly indicating when it is safe for vehicles and pedestrians to proceed, traffic signals help prevent collisions that often occur due to confusion or misjudgment.

1.3.3 Reduced Emissions and Fuel Consumption

Another significant advantage of traffic signal control systems is their positive impact on the environment. By optimizing traffic flow and reducing stop-and-go driving, these systems help decrease fuel consumption and lower vehicle emissions. When traffic moves more smoothly, vehicles spend less time idling at intersections, leading to a reduction in the amount of fuel burned and the release of harmful pollutants into the atmosphere.

1.3.4 Time and Cost Savings

Time is a valuable resource, and traffic signal control systems help save time for both drivers and pedestrians by reducing delays at intersections. For commuters, this means shorter travel times and a more predictable journey. For businesses that rely on transportation for goods and services, efficient traffic management can lead to significant cost savings by reducing delivery times and minimizing fuel costs.

1.3.5 Support for Pedestrian and Cyclist Safety

Traffic signal control systems are not only beneficial for vehicle traffic but also play a crucial role in enhancing the safety of pedestrians and cyclists. By providing dedicated signal phases for pedestrians and cyclists, these systems ensure that they can cross intersections safely without the risk of conflicts with vehicles.

1.4 Traffic Light in Smart Cities

In the wake of rapid urbanization and technological advancement, the cities are in need of adaptation with the huge revolution of technology that has touched every field of daily life and daily uses. That's where the concept of smart cities has emerged.

What is a smart city one should ask?

Smart City is a city that utilizes advanced technology and innovation to enhance the efficiency of services and urban planning, to reduce the cost and resources used by the city and target population. This ideal city focuses on a good design and the participation of the business sector and people in developing the city under the concept of a better and modern city where city people will live happily and sustainably, and have a good quality of life. Smart City is divided into several categories: Smart Environment, Smart Economy, Smart Energy, Smart Governance, Smart Mobility and Smart People, as shown in figure 1.6 **[17].**

The smart city that emphasizes the development of smart traffic and transportation system to drive the country by elevating efficiency and connection between transportation systems and various travel, increasing convenience and safety in that connection, and being environmentally friendly.

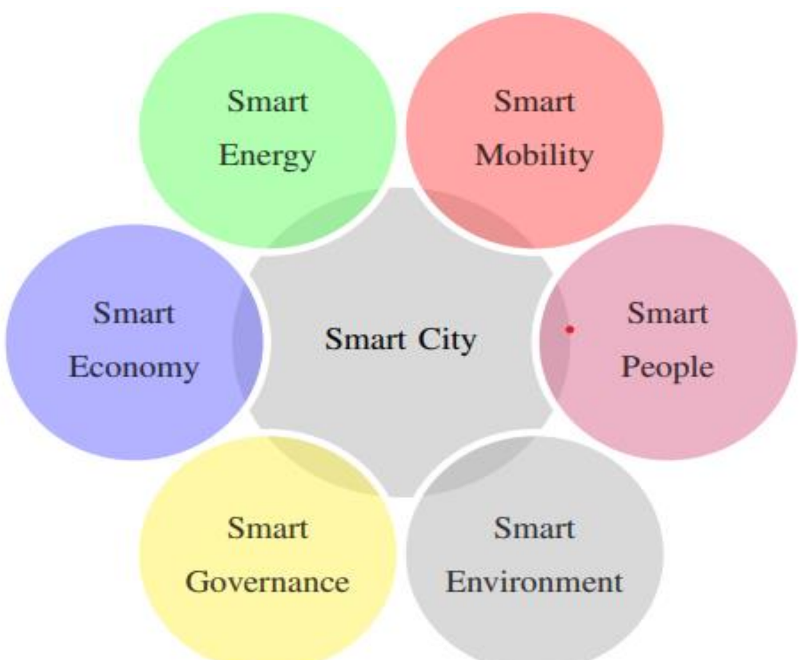


Figure 1.6: Smart city composition model https [8]

1.5 Smart Traffic Light Systems

1.5.1 Definition and Objectives

A smart traffic light in the context of traffic management refers to an automated system that gives different traffic lights depending on the real-time road conditions using data from sensors located at various locations of the roads and adjacent intersections. A smart traffic system is a traffic system that cohere other peripherals such as sensors, detectors, communicators, and such to conduct traffic flow where it is. The smart term means that the traffic signals can adapt and adjust to the current situations on the road through the equipped peripherals, then give out the proper response to the cases. Smart traffic lights curtail inefficient problems such as vehicles delay for a long time at an empty lane or traffic jams at busy hours of the day **[18].**

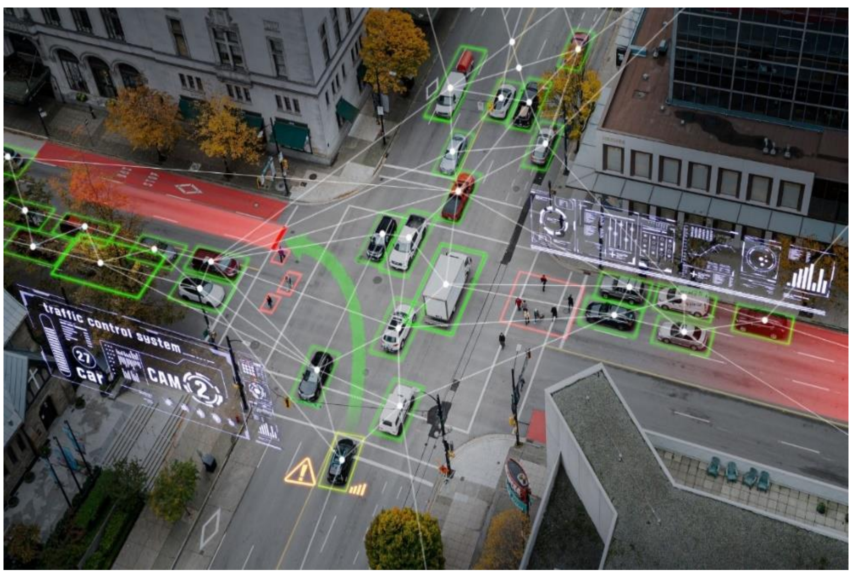


Figure 1.7: Smart Traffic light System

1.5.2 Advantages of Smart Traffic Light Systems

Smart signals represent a step toward cleaner and more efficient cities, providing a range of significant advantages that improve urban mobility, safety, environmental sustainability, and economic efficiency. These systems leverage advanced technologies such as sensors, cameras, real-time data analytics, and AI to dynamically manage traffic flow and pedestrian movement. Here are the key advantages **[16] [19] [20]:**

- **Enhanced Safety:** Smart signals significantly enhance pedestrian safety. The integration of sensors and cameras reduces pedestrian accidents. These technologies detect pedestrians and adjust traffic light timings accordingly. Improved visibility at night addresses the 76 percent of fatalities occurring during these hours. Smart signals prioritize pedestrian crossings, reducing risks associated with dangerous roadway designs.

- **Increased Efficiency:** Smart signals increase efficiency at pedestrian crossings. Real-time data processing reduces waiting times. Pedestrians experience shorter delays, leading to more efficient movement. Traffic flow improves for all road users, not just pedestrians. Vehicles spend less time idling at intersections, which enhances overall traffic management. Smoother traffic flow results from adaptive signal control. Smart signals adjust to real-time conditions, easing congestion.
- **Reduction in congestion and pollution:** Efficient traffic flow minimizes the time vehicles spend idling. Less idling leads to lower emissions and reduced pollution. Urban areas experience improved air quality as a result. Smart signals help untangle congestion caused by various road users. Pedestrians, cars, bicycles, and wheelchairs move more freely.
- **Economic impacts**: Improved traffic flow and reduced congestion translate directly into economic benefits. By minimizing delays, these systems contribute to increased productivity, as less time is wasted in traffic. This efficiency can have a ripple effect on the local economy, impacting everything from individual work productivity to broader economic activities.
- **Environmental Benefits:** Reducing Emissions and Improving Air Quality are an often-overlooked advantage of Intelligent Traffic Light Systems is their positive impact on the environment. By optimizing traffic flow and reducing idle time at intersections, these systems significantly cut down on fuel consumption and vehicular emissions. Efficient traffic management leads to less time spent idling at lights, which in turn reduces the carbon footprint of vehicles, contributing to improved air quality in urban areas, particularly in larger cities where vehicle emissions are a major contributor to air pollution.

1.5.3 Challenges and Limitations

Despite their numerous benefits, the widespread adoption of Smart Traffic Lights is not without challenges. Here are the challenges **[19][21]:**

- **Congestion Management**
  - **Traffic Congestion:** Despite their advanced capabilities, smart traffic lights may still struggle to mitigate heavy traffic during peak hours, leading to congestion on the roads.
  - **Adaptive Challenges:** Implementing adaptive traffic signal control systems can be complex and may require significant infrastructure upgrades, posing challenges for many cities.
- **Technical Limitations**
  - **Sensor Reliability:** The reliance on sensors for real-time data can be a double-edged sword. Sensor malfunctions or inaccuracies can lead to flawed decision-making by traffic lights.
  - **Software Vulnerabilities:** Smart traffic systems are susceptible to software glitches, which could disrupt traffic control and lead to unforeseen issues.
- **Cost of Implementation**
  - **High Initial Costs:** Installing and maintaining smart traffic light systems can be expensive, posing financial challenges for cash-strapped municipalities.
  - **Ongoing Maintenance:** Continuous software updates, sensor maintenance, and system upkeep can strain budgets.
- **Privacy Concerns**
  - **Data Collection:** The collection of real-time data for traffic management raises privacy concerns, as it can potentially track the movements of individuals and vehicles.
  - **Data Security:** Ensuring the security of collected data is a challenge, as it must be protected from unauthorized access and cyber threats.
- **Energy Consumption**
  - **Energy Usage:** Smart traffic lights require continuous power for sensors, communication equipment, and real-time control, potentially increasing energy consumption.
  - **Environmental Impact:** Higher energy consumption can contribute to environmental concerns, especially in cities striving for sustainability.

- **Integration and Standardization**
  - **Interoperability:** Ensuring seamless integration of smart traffic systems with existing infrastructure and vehicles can be a challenging task.
  - **Lack of Standards:** The absence of uniform standards for smart traffic technology can hinder widespread adoption and compatibility.
- **Public Acceptance and Education**
  - **Public Awareness:** Educating the public about the benefits and functioning of smart traffic lights is crucial for their successful implementation.
  - **Resistance to Change:** Resistance from drivers and pedestrians to new traffic management systems can pose challenges.
- **Vulnerability to Disasters**
  - **Natural Disasters:** Smart traffic systems may be vulnerable to natural disasters, disrupting traffic control during emergencies.
  - **Emergency Response:** Ensuring that smart traffic lights respond appropriately during emergency situations is essential for public safety.
- **Bias and Fairness**
  - **Algorithmic Bias:** The algorithms used in smart traffic systems can inadvertently introduce bias, affecting certain groups of road users unfairly.
  - **Equity Concerns:** Ensuring fair and equitable traffic management for all demographics is a challenge.
- **Unforeseen Circumstances**
  - **Unpredictable Events:** Smart traffic lights may struggle to adapt to rare or unexpected events on the road, such as accidents or protests.

1.5.4 How do Smart Traffic Light work?

Smart traffic signals are equipped with sensing, video capture, and connectivity technologies to collect real-time data from the environment. The obtained data is either pre-processed on the device or transmitted to a cloud-based transport management system, where it's processed by a predictive traffic light algorithm that generates instructions for signal adjustments, as shown in figure 1.6. Smart Traffic Lights has the ability to collect and exchange data with connected cars, onboard vehicle computers, telematics systems, cloud-based traffic platforms, and mobile travel or driving apps. A smart traffic light unit still has the familiar red/yellow/green three-light interface **[22] [23].**

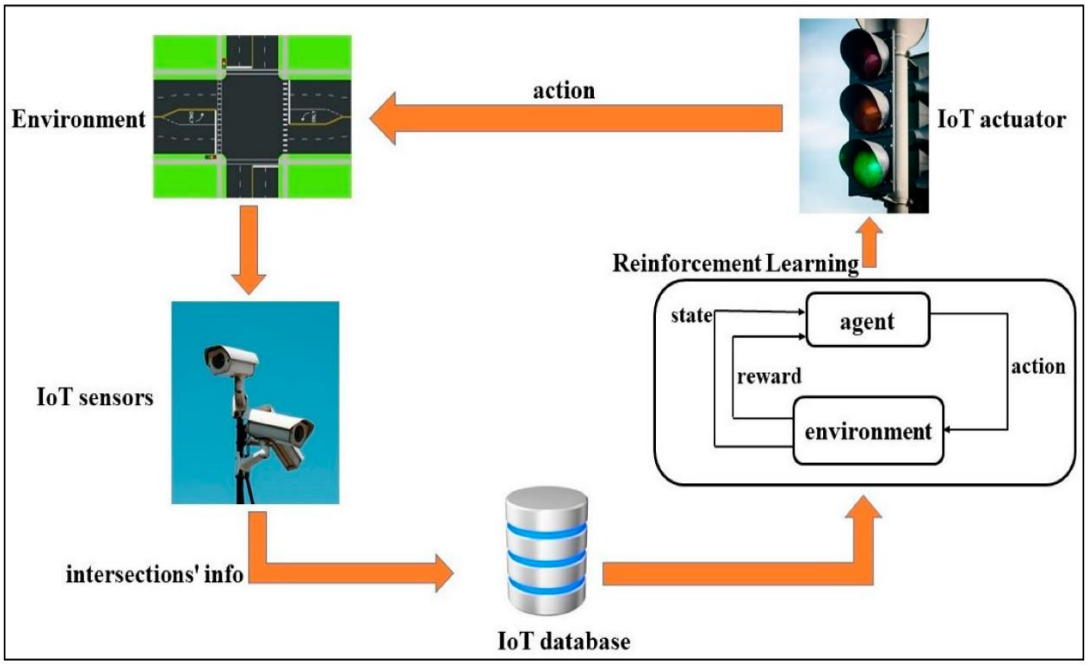


Figure 1.8: Smart Traffic light management System [24]

The basic components of a smart traffic light system include **[23][25]:**

1.5.4.1 Sensors and Cameras:

Smart traffic lights rely on a network of sensors and cameras to collect real-time traffic data. Then, the data collected is transmitted to a central traffic management center or cloud-based platform for processing and analysis. These sensors can include inductive loop detectors, video cameras, radar and Lidar sensors and Bluetooth and Wi-Fi sensors, etc.

1.5.4.2 Controller Unit

The controller unit is the brain of the traffic signal control system. It processes data from detectors and applies programmed algorithms to determine the optimal signal timing for each intersection. These algorithms can use various techniques:

- **Adaptive signal control:** This involves adjusting the timing of traffic signals based on real-time traffic volumes and patterns, using optimization models and feedback loops to minimize delays and stops.
- **Traffic prediction and forecasting:** This involves using historical traffic data and machine learning models to predict future traffic conditions and adjust the signals accordingly, taking into account factors such as time of day, weather, and special events.

- **Multi-modal optimization:** This involves considering the needs and behaviors of different types of road users, such as pedestrians, cyclists, and transit vehicles, and optimizing the signals to balance their competing demands and ensure safe and efficient movement for all.

The specific algorithms and models used can vary depending on the vendor, location, and objectives of the smart traffic light system, but they all aim to provide more responsive, adaptive, and intelligent traffic management than traditional fixed-time or manual methods.

1.5.4.3 Communication and Integration

To function effectively, smart traffic lights need to communicate and integrate with other systems and devices, such as:

- **Traffic management centers:** These are centralized facilities where traffic engineers and operators monitor and control traffic signals and other devices, using software and tools to analyze data, detect incidents, and make decisions.
- **Connected vehicles**: Smart traffic lights can communicate with vehicles equipped with dedicated short-range communication (DSRC) or cellular vehicle-to-everything (C-V2X) technologies, exchanging information about traffic conditions, signal timings, and speed advisories to optimize traffic flow and safety.
- **Smart city platforms:** Smart traffic lights can be integrated with other smart city systems and applications, such as public transportation, parking, and emergency response, to provide a more holistic and coordinated approach to urban mobility and operations.

1.5.5 Comparison with Traditional Systems

When comparing smart traffic light control systems and traditional methods, the evidence is clear: smart systems generally offer superior outcomes in terms of traffic flow, safety, and environmental impact. While the costs may be higher upfront, the long-term benefits—decreased congestion, lowered emissions, and reduced accidents—make a compelling case for their adoption. As cities continue to innovate, smart traffic systems stand out as a pivotal component in the future of urban transportation **[26].**

1.6 Intelligent transportation system ITS

1.6.1 Overview of ITS

ITS (Intelligent Transportation Systems) has been defined by WSP, as a combination of leading-edge information and communication technologies used in transportation and traffic management systems to improve the safety, efficiency, and sustainability of transportation networks, to reduce traffic congestion and to enhance drivers' experiences. ITS is an advanced application which aims to provide innovative services relating to different modes of transport and traffic management thus enabling users to be better informed and make safer, more coordinated, and 'smarter' use of transport networks. Additionally, ITS is an integrated set of technologies, communication systems, and management strategies designed to improve the efficiency, effectiveness, safety, and sustainability of transportation networks. It encompasses a wide range of applications and solutions aimed at addressing the contemporary challenges associated with transportation in urban and rural areas **[27].**

1.6.2 ITS Applications

The transportation landscape is undergoing a significant transformation with the emergence of Intelligent Transport Systems (ITS). ITS has offered a comprehensive solution to many of the challenges faced by today's urban environments, from traffic management to environmental conservation and safety enhancement. The importance of ITS lies in its various applications and innovative services that makes driving more fun, easy and safer. Here's a survey of the most important applications **[28]:**

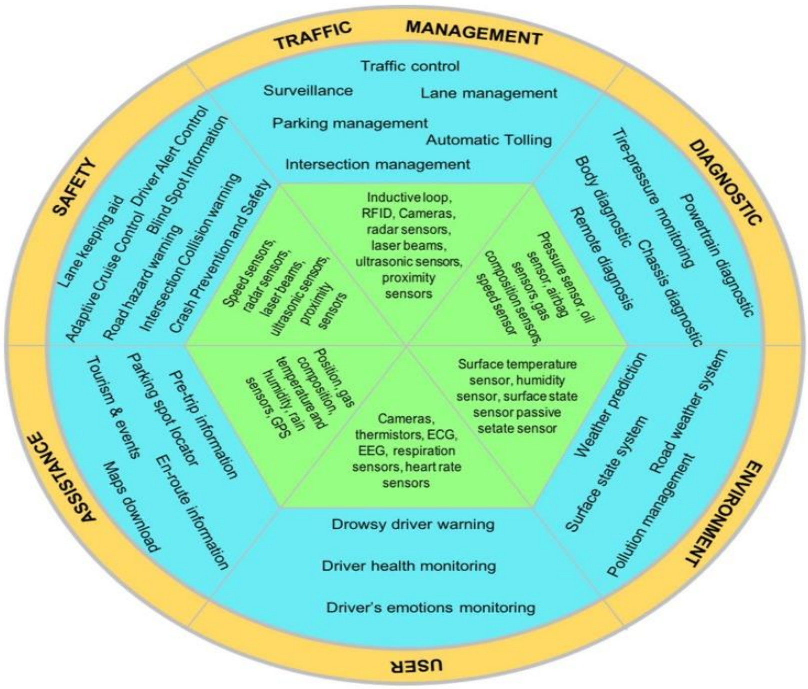


Figure 1.9: Figure ITS Applications [28]

1.6.2.1 Safety Category

In the safety category, applications focus on improving the safety of drivers and passengers (as shown in Figure1.10) thus reducing the number of accidents, injuries, and fatalities.

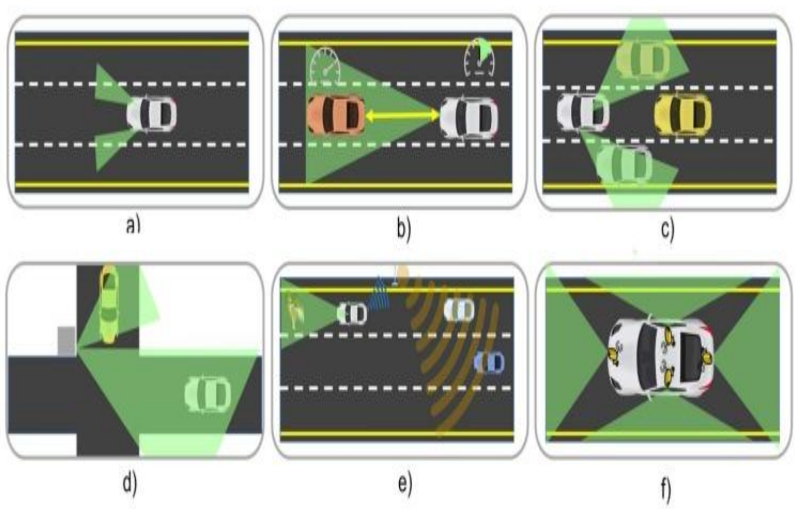


Figure 1.10: Safety Category: a) lane keeping aid b) adaptive cruise control c) blind spot information d) intersection collision warning e) road hazard warning f) surround view monitoring [28].

1.6.2.2 Traffic Management Category

ITS applications in this category improve the traffic flow in roads and urban zones (as shown in Figure 1.11). Surveillance applications can be divided into two categories: fixed surveillance systems which consist of fixed stations which use cameras and sensors that are installed on the roads to monitor road conditions. The second category, called surveillance-on-the-road, uses sensors and cameras embedded in vehicles to support surveillance.

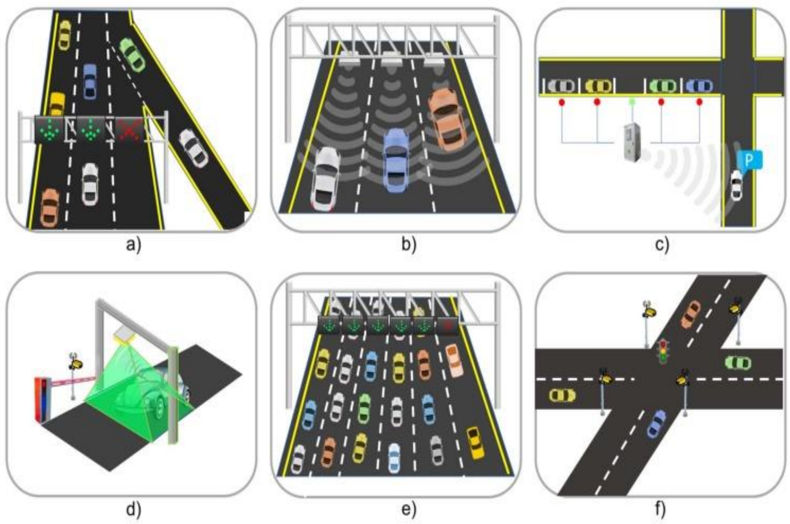


Figure 1.11: Traffic Management Category: a) lane management, b) surveillance c) parking management d) automatic tolling) special event transportation f) intersection management [28].

1.6.2.3 Diagnostic Category

This category focuses on providing diagnostic services that allow the detection of component failures that could lead to a breakdown by using different types of sensors. This category can be improved using communication technologies to send information directly to the cloud and to the service and maintenance area of the vehicle. Using a personalized vehicle registry, it is possible to identify and prevent possible car breakdowns by keeping a record and status of each vehicle part.

1.6.2.4 Environment Category

In the environment category information is collected from sensors deployed in or above the pavement to determine road conditions through the measurement of parameters such as road temperature, road conditions, number of chemical elements and friction or grip of the surface (Figure 1.12).

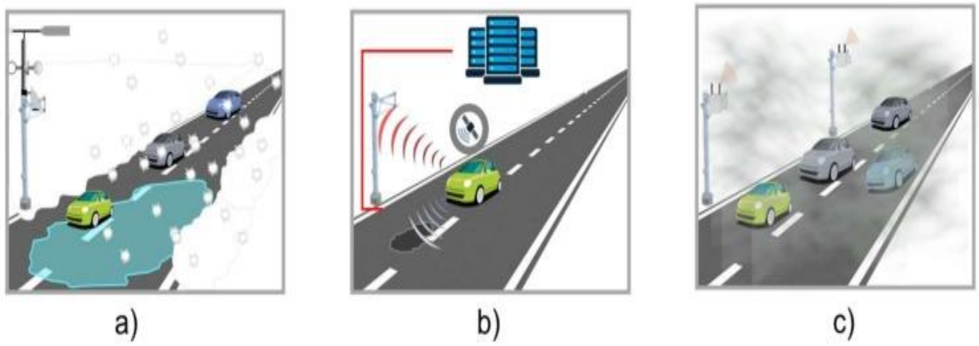


Figure 1.12: Environment Category: a) road weather condition b) surface state c) pollution management [28].

1.6.2.5 User Category

In the user category, sensors monitor the drivers' performance and behavior, which are essential for traffic safety and reducing accidents (Figure 1.13), using conditions such as: fatigue, alcohol levels and emotional state disorders.

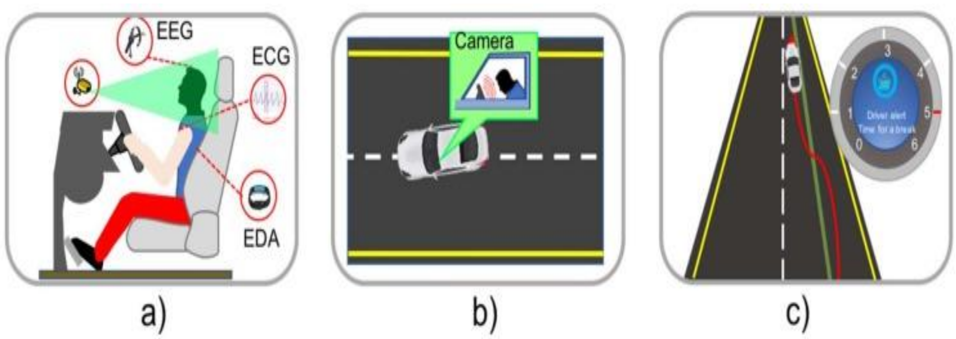


Figure 1.13: User Category: a) driver's health and emotions monitoring b) drowsy driver Warning c) driver alert control [28].

1.6.2.6 Assistance Category

Pre-trip information applications collect information about different road conditions, producing several trip options for various driving routes. Parking spot locator applications allow drivers to find available parking places at locations such as streets, garages, or parking lots. Magnetometers, RFID technologies and GPS are used to collect data from different parking spots and can offer drivers a wide variety of opportunities to park their vehicles **[28].**

1.6.3 Role of Smart Traffic Lights within the ITS Framework

the Role of an Intelligent Traffic Management System in modern cities, Smart Traffic Lights concept, is transformative. An ITS Framework contributes to the creation of smarter, more responsive urban environments by constantly analyzing traffic patterns, monitoring emerging trends, and embracing evolving vehicle technologies. As cities continue to grow, the integration of ITMS remains a cornerstone for ensuring sustainable and efficient urban mobility.

1.7 Artificiel Intelligence in Traffic Management techniques

Traffic congestion is a new phenomenon that is overwhelming large cities around the world. It is due to a static management of traffic lights. Therefore, the most feasible option in the short term is to improve the use of existing roads by providing a new control in intersections to allow vehicles to cross faster and minimize waiting times in the roadways **[29].** A lot of techniques are proposed to detect traffic density to control traffic signals.

1.7.1 Reinforcement Learning (RL)

Recent developments in traffic signal control increasingly rely on reinforcement learning (RL) algorithms for improved decision-making and adaptability.

Reinforcement learning is a machine learning paradigm where an agent seeks to maximize cumulative reward by developing a state-action policy through repeated interaction with its environment. Reinforcement learning agents achieve optimal control with respect to a defined reward by developing an optimal state-action policy.

Function approximators, such as artificial neural networks, have been used in reinforcement learning to approximate value functions when the agent's representation of the environment, or state space, becomes too large. Convolutional neural networks, a specific type of network architecture, are inspired by biological research on the animal visual cortex and have displayed impressive performance. They apply the mathematical convolution operation between various filters and the layer input to produce feature maps. Convolutional networks are advantageous because minimal input pre-processing is required and they can develop their own features **[30].**

Various techniques and algorithms have been developed to solve RL problems; each suited to different scenarios and assumptions about the environment. There are three main types of reinforcement learning algorithms: dynamic programming, Monte Carlo methods, and temporal difference learning. Each one works differently and is suited for certain kinds of problems **[31].**

- **Dynamic programming** methods use a complete model of the environment to find the best actions. They work well when we know everything about how the world behaves but require significant computing power for large problems.
- **Monte Carlo** methods learn from complete episodes of experience. They don't need to know how the world works beforehand. These methods are simple to understand and use, but they can be slow because they wait until the end of an episode to learn.
- **Temporal difference** learning combines ideas from dynamic programming and Monte Carlo methods. It learns from parts of experiences and updates its estimates frequently, making it faster than Monte Carlo methods in many cases. It also doesn't need a complete model of the environment like dynamic programming does.

Some popular algorithms that use these methods include:

- **Q-learning:** A type of temporal difference learning that learns the value of actions.
- **SARSA:** Another temporal difference method that learns while following a specific policy.
- **Policy gradient methods:** These directly learn the best policy without using a value function.
- **Actor-critic methods:** These combine policy learning with value function estimation

1.7.2 Deep Learning (DL)

Traffic prediction is the use of machine learning models to predict traffic patterns in a given area. This technology has been gaining popularity in recent years, with many cities using it to manage traffic and reduce congestion. Deep learning models have proven to be particularly effective in traffic prediction due to their ability to analyze large datasets and make accurate predictions. Deep learning models are a type of machine learning model that uses neural networks to analyze large datasets. These models are particularly effective for traffic prediction because they can analyze historical traffic data and identify patterns that are not immediately apparent to human analysts. This makes it possible for traffic managers to anticipate congestion and adjust traffic signal timings accordingly **[32].**

Deep learning employs a variety of specialized neural network architectures and algorithms designed to automatically learn hierarchical representations from data. Here are the most widely used deep learning techniques and models **[33]:**

- **Classic Neural Networks:** Classic neural networks, also called Fully Connected Neural Networks, are used image processing and detecting objects Classic Neural Networks consist of multilayer perceptron's which neurons are connected to a continuous network.
- **Convolutional Neural Networks:** Convolutional Neural Network or CNN is a classical artificial neural network model with a high potential for solving complex tasks and analyzing image and non-image data. It is based on the same principle as the arranged neurons in the cortex of the animal brain.
- **Recurrent Neural Networks:** RNNs are used to predict sequences, and in that process, they use knowledge from the previous state as input for future predictions. RNs come in two forms:
  - LSTM: Predicts data in time sequences using acquired memory. It has three gates: entrance, exit, and oblivion.
  - Gated RNNs: This form also predicts time sequence data through memory but has
- **Transfer Learning:** Transfer learning is the process of refining a model that has already been learned to carry out new and more precise tasks. It can only be successful if the model's features from the initial study are generic. One of the most popular deep learning approaches is transfer learning, which has a lower data requirement than others and hence requires less time for data processing.
- **Generative Adversarial Networks:** GAN consist of two components: a generator and a discriminator. The generator learns how to generates false data, while the discriminator Heps distinguish true from false data.
- **Deep Reinforcement Learning:** Deep Reinforcement Learning (DRL) combines multiple layers of artificial neural networks with reinforcement learning to train machines to replicate the way human brain works and solve problems by trial and error.

1.7.3 Fuzzy Logic Controllers

A fuzzy logic control system provides a better optimal solution for the fluctuating traffic system. Controlling the traffic flow system using fuzzy technology has the ability to convert human thinking process into an algorithm using some mathematical models. Implementation of real rules which are similar to the way that traffic policemen would think to manage traffic signal lights can be done by fuzzy if-then rules. The traffic signal controllers are supposed to adjust the cycle time of greenlight signal depending upon the amount of vehicles arrival which would maximize the traffic flow and control the regular waiting time. The inputs of fuzzy signal control system are generated by the help of an experience. Fuzzy rule-based system derives actions from given inputs by constructing if-then rules which represent the relation among the linguistic variables. In general, a fuzzy traffic signal controller will improve the traffic protection in the junction, usage of junction at its maximum level and minimize the delays **[34].**

1.7.4 Hybrid rule-based approaches:

Hybrid rule-based approaches combine traditional rule-based logic with data-driven or AI techniques to improve traffic management systems. These methods leverage the interpretability and structured decision-making of rule-based systems alongside the adaptability and learning capabilities of data-driven models or metaheuristic algorithms. The concept of hybrid approaches has emerged as a compelling solution. By combining the strengths of rule-based systems and AI models, hybrid approaches offer the potential to enhance the process of data extraction and uncover meaningful insights from diverse data sources **[35].**

Here are key types of hybrid rule-based approaches:

- Rule-Based & Machine Learning
- Rule-Based & Fuzzy Logic
- Rule-Based & Statistical Methods
- Rule-Based & Ontology/Knowledge Graphs
- Rule-Based & Neural Networks

1.8 Case Studies of AI-Based Traffic Optimization

1.8.1 Pittsburgh, USA (Surtrac System)

Surtrac is the name of an artificial intelligence system developed by a company called Rapid Flow Technologies, and it lives inside of traffic lights. It's been programmed to understand traffic theory, is connected to radar sensors and cameras, and can "talk" to other Surtrac lights. What's interesting about the Surtrac system is that it's decentralized. Each Surtrac light makes its own decision whether to display red, yellow or green based on what it senses in its own intersection as well as after parsing data transmitted from nearby Surtrac lights. It then beams data it has generated back to the other lights, building a real-time knowledge base **[36].**

The city of Pittsburg, Pennsylvania (USA) is pioneering with smart traffic technology in combination with Artificial Intelligence. Pittsburgh cuts down travel time by a quarter and traffic jams by 40 percent using radar sensors and cameras at every light to recognize traffic activity. The data coming from the sensors is used by Artificial Intelligence to streamline the traffic in the most intelligent and optimal way by reacting to the traffic conditions in real time **[37].**

1.8.2 Hangzhou, China (Alibaba City Brain)

City Brain is an AI-powered urban management system developed by Alibaba Cloud, first launched in 2016 in Hangzhou, China—Alibaba's home city. The system uses big data, artificial intelligence, and real-time data collection to optimize city operations, with an initial focus on traffic management. The system connects and controls over 1,300 traffic lights across the city, using real-time data and intelligent algorithms to optimize traffic flow and improve urban mobility. Over 200 traffic officers are connected via mobile phones, enabling them receive real-time alerts on any traffic-related emergencies **[38].**

As a result of City Brain's ability to predict traffic flow, detect accidents and instant feedback, Hangzhou has dropped to the 57th spot from 5th on the list of China's worst-congested cities, he added. "Hangzhou is probably the only city that can tell you how many cars are on the street at any given time," he said. In the case of medical emergencies, City Brain is able to change traffic lights so emergency vehicles like ambulances and fire trucks can head to the scene without interruption, accelerating their arrival time by 49%.

1.8.3 Barcelona, Spain

Barcelona's smart traffic lights are a key part of the city's advanced urban mobility and smart city initiatives. These traffic lights leverage AI, sensors, and real-time data to dynamically manage traffic flow, improve safety, and reduce congestion. Barcelona's smart traffic lights are a key part of the city's advanced urban mobility and smart city initiatives. These traffic lights leverage AI, sensors, and real-time data to dynamically manage traffic flow, improve safety, and reduce congestion. The use of LED technology in traffic lights in the city will cut energy consumption by 85%, saving 18 tones in CO_2 emissions and 65,000 euros in monthly power bills. The measure forms part of the Climate Plan, moving towards the de carbonization of the city **[39].**

Barcelona has introduced a sensor system to help drivers locate open parking spots. These sensors, embedded under the asphalt, detect available spaces and notify drivers, reducing emissions and congestion by directing them to vacant spots. Within the first year of implementation, the city issued 4000 parking permits daily. Additionally, there is an online payment option for parking fees. The Transports Metropolitan de Barcelona (TMB), the city's transport system, has implemented a new orthogonal bus network consisting of diagonal, vertical, and horizontal lines. This network is designed to be more frequent, user-friendly, and faster. The goal is for travelers to make only one transfer between any two points in the city for 95% of their journeys **[40].**

1.9 Conclusion

This chapter has shown that the future of urban traffic management lies in the adoption of smart, adaptive systems. Smart traffic lights, powered by AI and integrated into broader ITS networks, are already proving their value in cities around the world. They represent a major step forward in building more efficient, sustainable, and intelligent urban environments. By leveraging advanced technologies, these systems can adapt to real-time traffic conditions and prioritize emergency vehicles leading to safer roads and reduced environmental the proposed.

CHAPTER 2:

Automatic Traffic Light Control System

2.1 Introduction:

With the rapid advancement of intelligent urban planning, conventional traffic systems are no longer sufficient to meet modern demands. The limitations of fixed-time signal plans and reactive mechanisms have prompted the emergence of intelligent traffic control solutions. This chapter introduces the concept of an Automatic Traffic Light Control System as a key component in smart cities. It discusses the system architecture, the types of data and sensing technologies used, and the communication methods that enable real-time decision-making. This forms the foundation for the experimental implementation presented in the next chapter, where we detail our Intelligent Traffic Signal System (ITSS) and evaluate its performance.

2.2 Intelligent Control of Traffic Signal Lights Systems

Intelligent Traffic Signal Systems (ITSS) are smart solutions designed to manage traffic more efficiently. They use live data, sensors, and smart algorithms to improve traffic flow, cut down congestion, and make roads safer. These systems work by combining tools like vehicle sensors, communication networks, and central control units to automatically adjust traffic lights based on what's happening on the road in real time. **[42]**

One of the primary advantages of Intelligent Traffic Light Systems is their ability to significantly improve traffic flow and reduce congestion. By utilizing real-time data and adaptive signal timings, these systems ensure that traffic lights operate not on a fixed schedule, but in response to actual traffic conditions. **[43]**

2.2.1 Data Acquisition

In smart traffic systems, data is gathered accurate in real time using sensors and devices placed on roads and vehicles. These tools help track how traffic moves, how fast cars are going, and even things like weather or road conditions. This information helps cities manage traffic better, reduce accidents, and make transportation more efficient. **[44]**

2.2.1.1 Types of Data Collected

Smart traffic light systems rely on real-time information to make better decisions. To do this, they collect different types of data Below are the main types of data these systems collect **[45] [46]:**

- **Vehicle count and classification:** Number of vehicles on each road and the type of vehicles example: Car, Bus, Truck, Motorcycle, Ambulance.

- **Waiting time:** How long vehicles have been waiting at a red light.
- **Vehicle speed and direction:** How fast vehicles are moving on each road and which direction they are going.
- **Pedestrian movement:** Detection of pedestrians waiting to cross.
- **Weather and road conditions:** Data about rain, fog, or snow that may affect driving.
- **Time of day:** To help apply different timing during peak and off-peak hours.
- **Traffic density:** How crowded a road segment.

2.2.1.2 Sensing Technologies

Traffic light sensors are devices integrated into traffic signal systems that detect the presence, speed, and type of vehicles and pedestrians at intersections. Their primary function is to provide real-time data that allows traffic controllers to adjust signal timings dynamically, ensuring optimal traffic flow. In addition to reducing delays, these sensors also enhance safety by managing the movement of vehicles and pedestrians more efficiently, several types of sensors are employed in traffic light systems, each with its advantages and limitations. The most common types include **[47] [48]:**

- **Inductive loop detectors:** Inductive loop sensors consist of wire loops Embedded beneath the roadway; these sensors detect vehicles by measuring changes in magnetic fields.

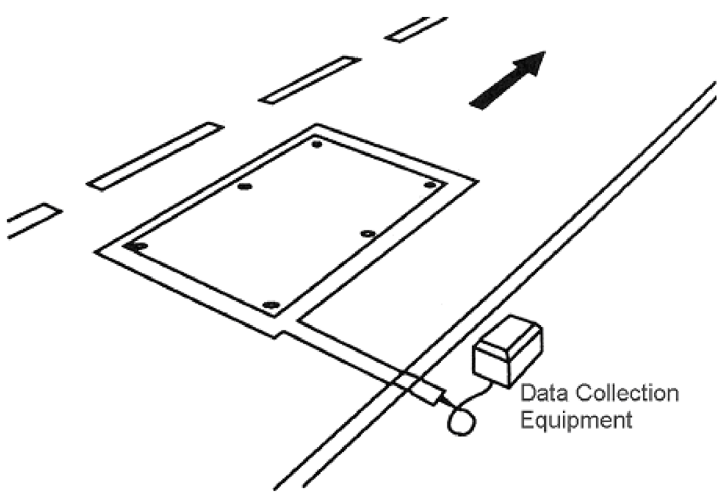


Figure 2.1: Inductive Loop detectors

- **Video cameras with computer vision:** Employ cameras and sophisticated image processing algorithms to monitor traffic and pedestrian movements These cameras provide real-time video feeds of intersections and surrounding areas. The video is analyzed to determine vehicle count, speed, classification, occupancy, turning movements, and length.

Figure 2.2: Video cameras

- **Infrared and ultrasonic sensors:** Use beams of infrared light to detect vehicles by measuring interruptions in the beam while ultrasonic sensors emit sound waves to detect objects.

Figure 2.3: Infrared and ultrasonic sensors

- **GPS/mobile data from connected vehicles:** Data from GPS-enabled devices and connected vehicles provide insights into vehicle locations, speeds, and routes. This information enhances traffic flow analysis and signal timing optimization**.**

- **Radar and LIDAR systems:** Radar sensors emit radio waves and analyze the reflected signals to determine a vehicle's presence, speed, and distance. LiDAR: Like radar, which uses radio waves, LiDAR (light detection and ranging) sensors on the other hand use laser beams to measure the distance to objects. This allows LiDAR to create a detailed 3D point cloud of the surrounding environment.

2.2.2 Data Transmission and Communication

Once collected, traffic data is transmitted to control centers or edge computing devices. This layer ensures continuous and reliable data flow to the decision-making systems.

Among Communication Technologies

2.2.2.1 5G / LTE / NB-IoT for fast, low-latency communication:

a) **5G** is the next evolution from the 4G LTE wireless networks that we use today. It is designed to connect more devices and advanced applications it enables significantly faster data speeds and increased network capacity, providing a more seamless and efficient internet experience and offers ultra-low latency, which is the delay in transmitting data between devices and the network **[49].**

b) **LTE** stands for “Long Term Evolution” is a type of high-speed wireless communication used mostly in 4G networks. It lets devices like phones or sensors connect to the internet quickly and send or receive data in real time. It’s known for being fast, stable, and having low delay when transferring information **[50]**.

c) **NB-IoT** Narrowband IoT (NB-IoT) is a low-power, wide-area network (LPWAN) technology designed specifically for the Internet of Things, it is optimized for low data rates, extended battery life, and deep coverage **[51]**.

2.2.2.2 Dedicated Short-Range Communication (DSRC) for V2I (Vehicle-to-Infrastructure) communication:

Dedicated Short-Range Communications, or DSRC, is a wireless technology made for safe and fast communication between vehicles (V2V) and between vehicles and roadside units (V2I). It’s designed to work in specific areas with low delay, using the 5.85 to 5.925 GHz frequency band. DSRC supports data speeds between 6 and 27 Mb/s and can reach up to 1,500 meters, even when cars are moving at speeds of 140 km/h. V2V is used when vehicles need to share information directly with each other, while V2I connects vehicles to infrastructure like traffic signs or control units. Some systems send updates every 100 milliseconds, while others only send data when something important happens **[52]**.

2.2.2.3 Wi-Fi / Ethernet in local installations:

a) **Wi-Fi** is a wireless technology that lets devices like phones, laptops, and cameras connect to each other and to the internet without cables. It works through a wireless router, which sends the internet signal to your device, allowing you to browse, stream, or share data within a network. In smart traffic light systems **[53]**, Wi-Fi is used to wirelessly connect traffic signals with the central control center. This connection allows real-time sharing of traffic data, helping the system adjust signal timings based on actual traffic flow and conditions.

b) **Ethernet** is a network technology primarily used for local area networks (LANs), it operates by transmitting data packets over a physical medium, typically using twisted pair cables, coaxial cables, or fiber optic cables. It uses both hardware components, such as network interface cards (NICs) and switches, and software protocols to manage the flow of data, ensuring reliable and orderly communication between devices **[54]**.

Ethernet is used in smart traffic systems to create stable wired links between signals and control centers. It ensures secure, reliable communication for coordinating traffic lights across intersections.

2.2.2.4 Fog computing nodes for near-sensor processing:

Fog computing helps connect the cloud and nearby devices, like sensors or IoT tools, by handling some of the work—like processing data or making decisions—closer to where the data is created. Instead of sending everything straight to the cloud, fog computing allows things like storage, networking, and quick decision-making to happen on local devices along the way. For example, in smart transportation systems, GPS data can be filtered or compressed near the source before it’s sent to the cloud, saving time and reducing traffic on the network **[55]**.

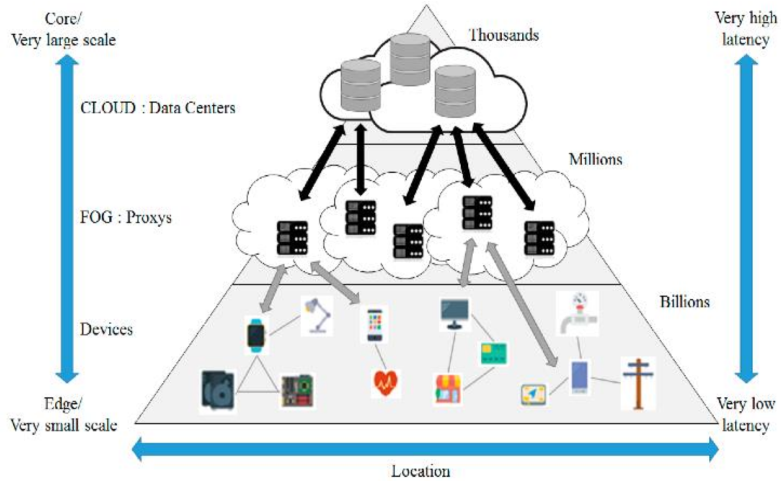


Figure 2.4: The Structure of Fog Computing. [56]

2.2.3 Data Pre-processing

Before decision-making, the system must clean and organize the data. This essential pre-processing step prepares the raw inputs for accurate and efficient analysis in the following section we present the key pre-processing steps involved in this process.

2.2.3.1 Noise filtering from sensor errors:

Data filtering plays a key role in industrial systems by cleaning input signals through the removal of noise, spikes, or errors before they are used for control or monitoring. This helps improve the accuracy and stability of automation systems. There is several filtering methods used in practice, each with specific advantages depending on the application **[57]**.

2.2.3.2 Data fusion from multiple sources (e.g., camera + radar):

Data fusion from multiple sources, such as combining camera and radar inputs, is a technique used in intelligent transportation systems to enhance the accuracy and reliability of environmental perception. By integrating data from these diverse sensors, the system can compensate for the limitations of individual sensors, leading to improved object detection and tracking capabilities **[58]**.

2.2.3.3 Anomaly detection (e.g., stalled vehicles):

Anomaly detection leverages artificial intelligence (AI) and machine learning (ML) to automatically identify unexpected changes in a data set's normal behavior, In the context of smart traffic systems, this involves detecting unusual traffic conditions, such as stalled vehicles, that disrupt normal traffic flow **[59]**.

2.2.3.4 Temporal and spatial alignment of data points:

Temporal and spatial alignment means making sure that data from different sensors matches up correctly in both time and location. This step is vital to ensure all information reflects the real situation accurately, allowing the system to make smart and reliable decisions.

2.2.3.5 Temporal Alignment

When data comes from different sensors like cameras or loop detectors, it needs to be lined up on the same timeline. This makes it easier to understand how traffic events unfold and respond to them correctly. Accurate timing is especially important for real-time monitoring and predicting what's likely to happen next **[60]**.

2.2.3.6 Spatial Alignment:

To make full use of data from different sources, each piece of information must be linked to its exact location. This helps create a clear picture of what's happening on the roads, making it easier to manage traffic flow and improve routes **[60]**.

2.2.4 Traffic Analysis and Prediction

Using machine learning and analytics, the system analyzes current conditions and forecasts traffic patterns. These insights help anticipate peak loads, lane blockages, and pedestrian surges. The techniques used are:

2.2.4.1 Short-Term Traffic Prediction using LSTM (Long Short-Term Memory) networks:

LSTM-based short-term traffic prediction is a deep learning technique developed to estimate upcoming traffic conditions like flow rates, vehicle speed, or congestion levels by processing past traffic records. What sets LSTM networks apart is their strength in handling time-dependent data—they can effectively learn and retain both recent changes and long-term trends in traffic, making them highly suitable for modeling complex traffic behaviors **[61]**.

2.2.4.2 Traffic flow modeling using Kalman Filters or ARIMA models:

a) **The Kalman filter** is a state-space model that was first introduced by Kalman. It can be applied to model systems with multi-input and multi-output and can be used for both stationary and non-stationary situations. This feature of the Kalman filter makes it an appropriate choice for modeling the traffic states. Kalman filter updates the prediction of state variables based on the observation in the previous step. Therefore, it only needs to store the previous estimate information. The Kalman filter has two distinct features **[62]:**

- It does not require any additional space to store the entire previously observed data.
- It is computationally efficient since it does not need to utilize all the previous estimated/measured data in each step of the prediction process.

b) ARIMA stands for Autoregressive Integrated Moving Average and it's a technique for time series analysis and for forecasting possible future values of a time series. Autoregressive modeling and Moving Average modeling are two different approaches to forecasting time series data. ARIMA integrates these two approaches, hence the name. Forecasting is a branch of machine learning using the past behavior of a time series to predict the one or more future values of that time series **[63]**. ARIMA works in traffic flow modeling by first transforming the traffic flow time series into a stationary sequence through appropriate differencing operations, which removes trends and seasonality in the data. The model is represented as ARIMA (p, d, q), where p is the autoregressive order, d is the differencing order, and q is the moving average order. After differencing, the structure of the sequence is modeled using the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots to determine suitable values for p and q. The ARIMA model then fits the historical traffic flow data and is commonly used for short-term traffic flow forecasting due to its ability to handle data with trend and seasonality structures. This approach allows the model to capture temporal dependencies in traffic flow and make predictions about future traffic volumes **[64]**.

2.2.4.3 Congestion detection through clustering and heatmaps:

Congestion detection through clustering and heatmaps involves using data clustering techniques to identify patterns of traffic congestion and visualizing these patterns with heatmaps for better interpretation and analysis.

a) Clustering is an unsupervised method used to detect natural groupings in traffic data, especially for identifying congestion patterns from speed and flow information. Feature extraction is done in two ways: point-based using computer vision, and area-based using segmentation methods like Watershed. These features help group similar congestion events through hierarchical clustering. The result is automatic congestion classification, reducing manual work and enhancing traffic behavior analysis **[65]**.

b) **Heatmaps** represent data values as colors in a matrix format, making it easier to visualize complex traffic patterns. In congestion detection, heatmaps display traffic speed or congestion intensity over space and time, with each pixel's color intensity corresponding to speed values. This visual representation helps in identifying congestion hotspots and temporal changes in traffic flow **[66]**.

2.2.5 Decision-Making and Optimization

Decision-making and optimization are core functions in smart traffic systems, enabling adaptive responses to real-time traffic data. Below, we explore methods that enhance traffic flow through intelligent control strategies.

2.2.5.1 Rule-Based Algorithms:

These are predefined thresholds for signal changes. Rule-based algorithms in traffic signal control utilize predefined thresholds to determine when to change signal phases. These thresholds are based on specific traffic parameters such as queue length, vehicle count, or waiting time.

2.2.5.2 Optimization Algorithms:

Among these algorithms are Genetic Algorithm, Ant Colony Optimization and Particle Swarm Optimization.

a) **The Genetic Algorithm (GA)** is a heuristic function for optimization, where the extreme of the function (i.e., minimal or maximal) cannot be established analytically. A population of potential solutions is refined iteratively by employing a strategy inspired by Darwinist evolution or natural selection. Genetic Algorithms promote "survival of the fittest This is a stochastic search technique to look for optimal solution.

   GA technology in the traffic control system is to provide intelligent green interval responses based on dynamic traffic load inputs, thereby overcoming the inefficiencies of the conventional fixed traffic controllers **[67]**.

b) **The Ant Colony Optimization (ACO)** is a meta-heuristic algorithm based on the behavior of ant colonies searching for food. ACO has successfully been employed to solve many complicated combinatorial optimization problems and its stochastic and decentralized nature fits well with traffic networks. **ACO** can be used to dynamically adjust green signal durations in intersections by modeling cars as ants, choosing the least congested route based on past 'pheromone' (traffic flow history) **[68]**.

c) **The Particle Swarm Optimization** is an evolutionary computation technique motivated by the social cooperative and competitive behavior of birds flocking or fish schooling. In the PSO algorithm, a swarm is defined as a population of interacting elements and the particle is a member in the swarm, indicating a potential solution in the optimization process. Each member in the swarm adjusts the search patterns according to its experience and to other members experiences. **PSO** is adopted to optimize the multiphase traffic signal timing and phase sequence at the intersection. The optimization objective is to minimize the average vehicle delay of every cycle at the intersection **[69]**.

2.2.5.3 Reinforcement Learning (RL):

Reinforcement Learning (RL) is a machine learning paradigm where agents learn optimal behaviors through interactions with their environment, receiving feedback in the form of rewards or penalties. In the context of traffic signal control, RL enables adaptive decision-making to optimize traffic flow. Unlike traditional traffic signal systems, RL-based approaches do not require a perfect model of the environment. Instead, they learn optimal strategies through interaction with the environment, leveraging a trial-and-error approach to maximize long-term rewards **[70]**.

2.2.5.4 Objectives:

Objectives of these algorithms are:

- Minimize average waiting time.
- Minimize average vehicle delay
- Reduce vehicle queue length.
- Prevent congestion and gridlocks.
- Improve traffic flow efficiency.
- Prioritize emergency or public transport vehicles.
- Balance throughput across multiple intersections.
- Ensure fairness among all directions.
- Decrease the number of vehicles stops.

2.2.6 Traffic Signal Lights & Actuators

Once the optimal signal timing is determined, the control system proceeds to adjust the traffic light phases through various actuation hardware components. These include:

2.2.6.1 Programmable Logic Controllers (PLCs):

A programmable Logic Controller (PLC) is a specialized digital computer employed in industrial settings for automation and control. Acting as the central brain of machinery and processes, PLCs receive input from sensors, process the data through programmed logic, and generate output signals to control devices like motors and valves. They use a programming language, often ladder logic, resembling electrical relay diagrams. PLCs are ruggedized for harsh industrial environments and play a crucial role in automating tasks and reducing manufacturing and energy. Their modular design allows scalability, making them versatile components in modern industrial control systems **[71]**.

In intelligent traffic light PLC uses as control core, using a sensor module for receiving real-time information of vehicles, traffic control mode for information to select the traffic lights **[72]**.

There are mainly 6 major parts of a PLC that are:

- Processor
- Memory (RAM/ ROM)
- Input device
- Output device
- Power supply
- Programming device

Figure 2.5: Programmable Logic Controllers

2.2.6.2 Networked traffic light controllers:

Networked traffic signal controllers are designed to be integrated into centralized or distributed traffic management systems. These controllers communicate with each other and with central control units via wired (e.g., Ethernet) or wireless (e.g., Wi-Fi, LTE) networks, enabling real-time monitoring, coordination, and optimization of traffic signal timing across intersections **[73]**.

2.2.6.3 LED-based dynamic signage:

LED-based dynamic signage plays a pivotal role in modern traffic management systems by providing real-time information to drivers, enhancing road safety, and optimizing traffic flow.
LED road sign screens can clearly display traffic information regardless of day or night, sunny or rainy. This is because they use high-brightness LED lamp beads, which can remain 'bright' even under direct sunlight **[74]**.

- **Examples of LED-Based Dynamic Signage:**
  - **Lane control signs:** LED signs above lanes display dynamic arrows or "X" symbols to open/close specific lanes based on traffic flow.

Figure 2.6: Lane control signs

- **Variable speed limit signs (VSLS):** LED panels display changing speed limits based on congestion, weather, or incidents.

Figure 2.7 Variable speed limit signs

- **Pedestrian countdown timers:** LED displays show remaining time for pedestrians to cross safely, improving safety and compliance.

Figure 2.8: Pedestrian countdown timers

- **Emergency vehicle priority display:** Dynamic LED signs notify nearby drivers to yield when an emergency vehicle is approaching an intersection.

These systems must act within milliseconds to reflect decisions in real-time.

2.3 Our ITSS proposed

Our Intelligent Traffic Signal System (ITSS) was designed to manage traffic flow at intersections using real-time video cameras placed at each lane to monitor traffic continuously analysis and automated control logic. The system captures live camera feeds from each lane, detects and counts vehicles using a YOLOv8 deep learning model, and tracks them across frames using the SORT algorithm. Based on the number of vehicles, their waiting time, and the presence of emergency vehicles, a decision algorithm determines the most appropriate signal phase. The system prioritizes urgent cases, dynamically adjusts green light durations, and ensures optimal flow using hybrid rule-based logic. Finally, the selected signal phase is applied and visualized through a real-time interface. This integrated approach allows for efficient, responsive, and intelligent traffic control.

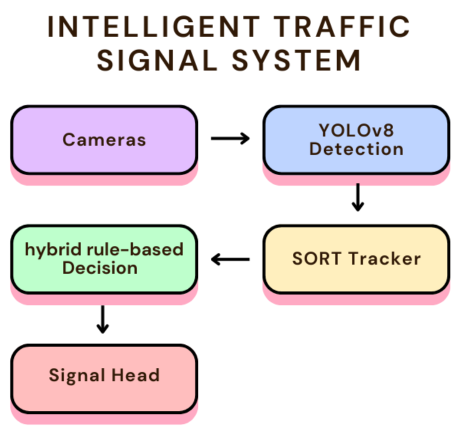


Figure 2.9: ITSS proposed

2.3.1 Object Detection by deep learning

2.3.1.1 What is deep learning?

Deep learning is a subfield of machine learning that relies on deep neural networks—structures made up of many layers to model complex decision-making, much like the human brain, DL algorithms can automatically learn representations from data such as images, video, or text, without introducing human domain knowledge. It is behind many of the AI systems used in everyday applications today **[75]**.

2.3.1.2 How deep learning works?

Neural networks, also known as artificial neural networks, are designed to imitate how the human brain functions by using data inputs, weights, and biases—components that act like digital neurons.

These elements work together to help the system identify, classify, and interpret patterns or objects within data.

Deep neural networks are built with several interconnected layers of nodes. Each layer processes and refines the information from the previous one, improving prediction or classification accuracy—a process known as forward propagation. The input and output layers are referred to as visible layers. The input layer receives the raw data, while the output layer produces the final result or classification.

To improve accuracy, the network also relies on a method called backpropagation. This process uses optimization algorithms like gradient descent to evaluate prediction errors and then adjusts the weights and biases by moving backward through the layers. By combining forward propagation and backpropagation, the model gradually improves its predictions over time.

Training deep learning models demands significant computational power. High-performance GPUs are commonly used due to their ability to perform parallel processing across many cores with large memory capacity. In some cases, distributed cloud computing is also used to meet these demands. However, running and managing multiple GPUs on-site can be resource-intensive and expensive to scale. On the software side, most deep learning systems are developed using frameworks like JAX, PyTorch, or TensorFlow.

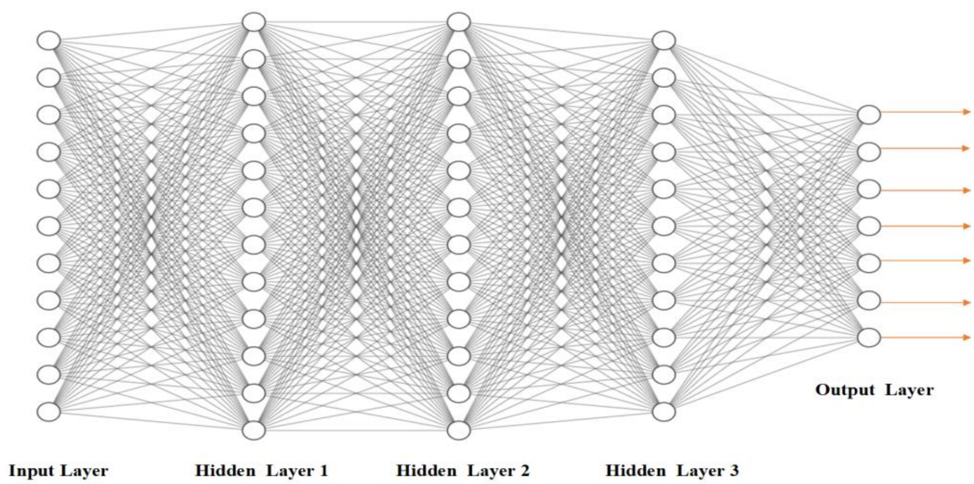


Figure 2.10: Deep Nets Architecture. [76].

2.3.1.3 Types of Deep Learning Models

Deep learning models for vehicle detection are grouped into two types: one-stage and two-stage. Below, we briefly explain how each type works and where they are typically used

a) Two-Stage Models:

Two-stage object detectors are a type of object detection model known for their high accuracy, especially in complex visual scenes. They work in two steps **[77] [78]:**

- **Step 1:** The model begins by extracting important features from the input image using a Convolutional Neural Network (CNN).
- **Step 2:** It then generates several regions of interest, known as object proposals. Only these selected regions are used for object classification and precise localization.

Examples of two-stage object detection models:

- **R-CNN:** It was the first to mix region proposals with CNNs but ran slowly since it processed each region one by one.
- **Fast R-CNN:** Made things faster by reusing features across regions using RoI pooling.
- **Faster R-CNN:** Improved speed by adding region proposal generation directly inside the network.
- **Mask R-CNN:** Built on Faster R-CNN by adding a part that draws object masks for segmentation.

b) One-Stage Models:

In computer vision, one-stage object detectors are designed for both speed and accuracy. Unlike two-stage models, they detect and classify objects in one step, making them much faster and ideal for real-time use **[79]**.

Examples of One-stage object detection models **[80]**:

- **YOLO (You Only Look Once):** A real-time object detection model that predicts bounding boxes and class probabilities in a single step, it is widely used for its speed and efficiency in applications like traffic monitoring.
- **SSD (Single Shot Multi Box Detector):** Detects objects by applying a set of default bounding boxes at multiple scales and aspect ratios across feature maps.

- **Retina Net:** Improves accuracy by using Focal Loss to focus on hard examples and handle class imbalance.
- **Efficient Det:** A compact and scalable model that adjusts depth, width, and resolution uniformly.

It achieves strong accuracy with fewer parameters, ideal for real-time edge deployment.

2.3.2 Object Detection using YOLO:

2.3.2.1 What is YOLO (You Only Look Once)?

YOLO (You Only Look Once), a popular object detection and image segmentation model, was developed by Joseph Redmon and Ali Farhadi at the University of Washington. Launched in 2015, YOLO gained popularity for its high speed and accuracy **[81]**.

- **YOLOv2**, released in 2016, improved the original model by incorporating batch normalization, anchor boxes, and dimension clusters.
- **YOLOv3**, launched in 2018, further enhanced the model's performance using a more efficient backbone network, multiple anchors, and spatial pyramid pooling.
- **YOLOv4** was released in 2020, introducing innovations like Mosaic data augmentation, a new anchor-free detection head, and a new loss function.
- **YOLOv5** further improved the model's performance and added new features such as hyperparameter optimization, integrated experiment tracking, and automatic export to popular export formats.
- **YOLOv6** was open-sourced by Meituan in 2022 and is used in many of the company's autonomous delivery robots.
- **YOLOv7** added additional tasks such as pose estimation on the COCO key points dataset.
- **YOLOv8** released in 2023 by Ultralytics, introduced new features and improvements for enhanced performance, flexibility, and efficiency, supporting a full range of vision AI tasks.
- **YOLOv9** introduces innovative methods like Programmable Gradient Information (PGI) and the Generalized Efficient Layer Aggregation Network (GELAN).

- **YOLOv10** created by researchers from Tsinghua University using the Ultralytics Python package, provides real-time object detection advancements by introducing an End-to-End head that eliminates Non-Maximum Suppression (NMS) requirements.
- **YOLO11** 🚀 NEW: Ultralytics' latest YOLO models, deliver state-of-the-art (SOTA) performance across multiple tasks, including object detection, segmentation, pose estimation, tracking, and classification, leveraging **capabilities across diverse AI applications and domains.**

2.3.3 Vehicle Detection using YOLOv8:

In our project, we trained a YOLOv8 model to detect five types of vehicles: cars, buses, trucks, motorcycles, and ambulances. The pretrained YOLOv8n model was based on the COCO dataset, which does not include ambulances, so we trained a custom model to add this missing class.

Below, we explain the Architecture of YOLOv8 model works **[82][83][84]:**

Table 2-1: YOLOv8 Variants [83]

Model variant	d (depth_multiple)	w (width_multiple)	mc (max_channels)
n	0.33	0.25	1024
s	0.33	0.50	1024
m	0.67	0.75	768
l	1.00	1.00	512
xl	1.00	1.25	512

- **n**: smallest model, fastest inference but lowest accuracy
- **s**: small model, good balance of speed and accuracy
- **m**: medium model, higher accuracy than small models with moderate inference speed
- **l**: large model, highest accuracy but slowest inference
- **xl**: extra-large model, best accuracy for resource-intensive applications

2.3.3.1 YOLOv8 Variants:

All of these models belong to the YOLOv8 family, each variant offers different trade-offs between accuracy, speed, and model size. The variants are divided based on the difference in the value of the parameters like depth_multiple (d), width_multiple (w), and max_channel (mc).

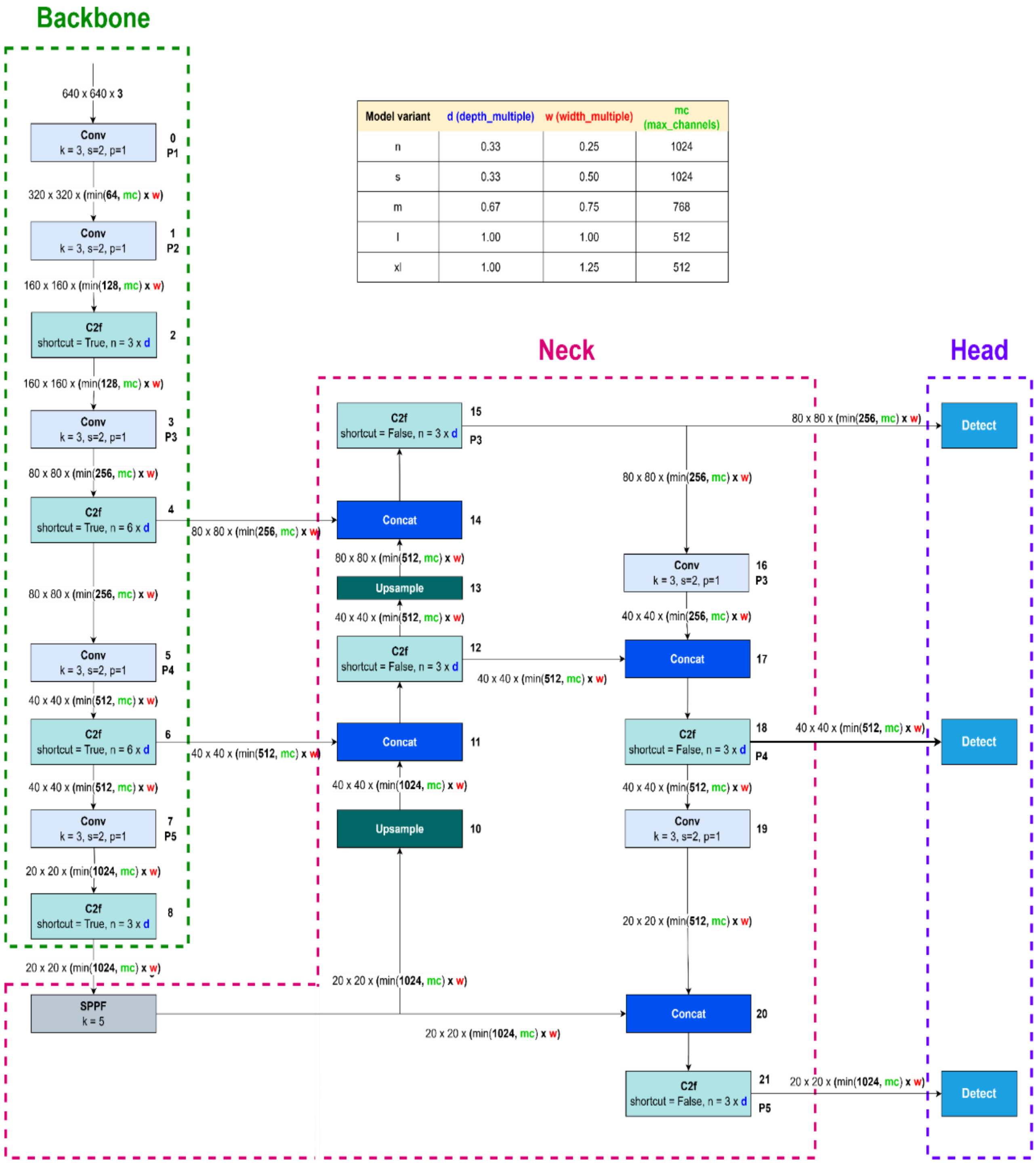
Types of YOLOv8:

- **depth_multiple (d):** depth_multiple parameters determine the number of Bottleneck Blocks are used in the C2f block. This scales the number of layers in the network. A value less than 1 reduces the depth (fewer layers), making the model smaller and faster but potentially less accurate. Conversely, a value greater than 1 increases the depth (more layers), leading to a larger and potentially more accurate model but slower to run.
- **width_multiple (w):** This scales the number of channels in the convolutional layers. A value less than 1 thins the network (fewer channels), resulting in a smaller and faster model but potentially sacrificing some accuracy. On the other hand, a value greater than 1 widens the network (more channels), creating a larger and potentially more accurate model but requiring more processing power.
- **max_channels (mc):** This parameter sets an upper limit on the number of channels allowed in the network. It is a safety measure to prevent the model from becoming too wide (too many channels) especially when width_multiple is set high. This can help control the model size and prevent overfitting.

2.3.3.2 Architecture of YOLOv8 model:

YOLOv8 Architecture consists of three main sections (figure *):

- **Backbone** is the deep learning architecture that acts as a feature extractor of the inputted image.
- **Neck** combines the features acquired from the various layers of the Backbone module.
- **Head** predicts the classes and the bounding box of the objects which is the final output produced by the object detection model.



Model variant	d (depth_multiple)	w (width_multiple)	mc (max_channels)
n	0.33	0.25	1024
s	0.33	0.50	1024
m	0.67	0.75	768
l	1.00	1.00	512
xl	1.00	1.25	512

Figure 2.11: YOLOv8 Architecture [83]

2.3.3.3 Blocks used in YOLOv8 Architecture:

1) Convolutional Block (Conv Block):

The basic building block of the architecture includes three main parts: a Conv2D layer, a BatchNorm2D layer, and a SiLU activation function.

- **Conv2D Layer:** This layer scans the input image using small filters to pick out important features like edges or patterns. The number of filters (k) decides how many features it can detect. The stride (s) controls how far the filter moves each time — bigger strides mean a smaller output. Padding (p) adds extra space around the image to keep its size. And channels (c) refer to how many color layers the input has, like 3 for an RGB image.
- **BatchNorm2D Layer:** This helps keep the data stable as it moves through the network by normalizing values. It speeds up training and makes the process more reliable.
- **SiLU Activation Function:** Also known as Swish, this function lets the network learn more smoothly by allowing gentle gradients. It's defined as SiLU(x) = x * sigmoid(x). This smooth behavior helps the model train better, especially in deep layers.

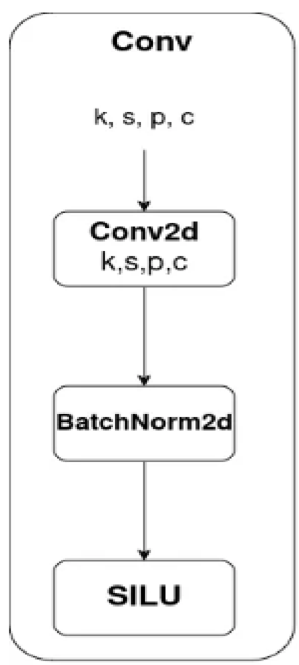


Figure 2.12: Convolutional Block [82]

2) Bottleneck Block:

The bottleneck block is made up of one or two Conv blocks and may include a shortcut connection depending on the setting.

- If shortcut = true, the block includes a shortcut (or skip) connection, which allows the input to bypass the Conv blocks and be added directly to the output.
- If shortcut = false, the input simply goes through two Conv blocks in a row.

3) Shortcut Connection:

This is like a fast lane that skips certain layers. It helps the model train better by allowing gradients to flow through the network more smoothly. This is useful in deep networks because it reduces the risk of the vanishing gradient problem, where the updates become too small for earlier layers to learn anything meaningful.

4) Vanishing Gradient Problem:

This happens when gradients shrink too much as they move backward through a deep network, making it hard for the early layers to learn and slowing down training.

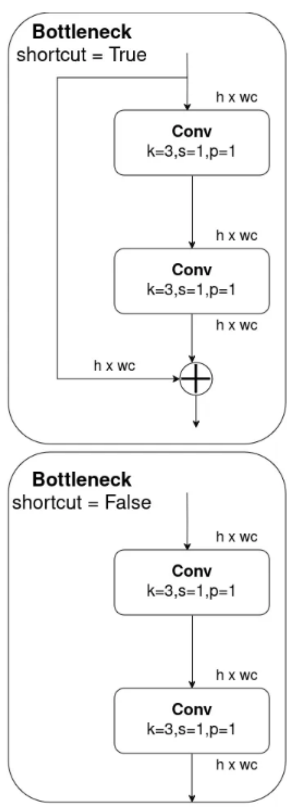


Figure 2.13: Bottleneck Block [82]

5) C2f Block:

The C2f block begins with a convolutional layer that produces a feature map. This feature map is then split into two parts. One part is sent through several bottleneck blocks, with the number of these blocks determined by the model's depth_multiple parameters. The other part skips the bottlenecks entirely and goes straight to the concatenation stage. After the bottleneck and shortcut branches finish processing, their outputs are merged together using a concat operation and then passed into one final convolutional layer. This design helps the model combine deep, learned features with more direct, unaltered ones for improved performance.

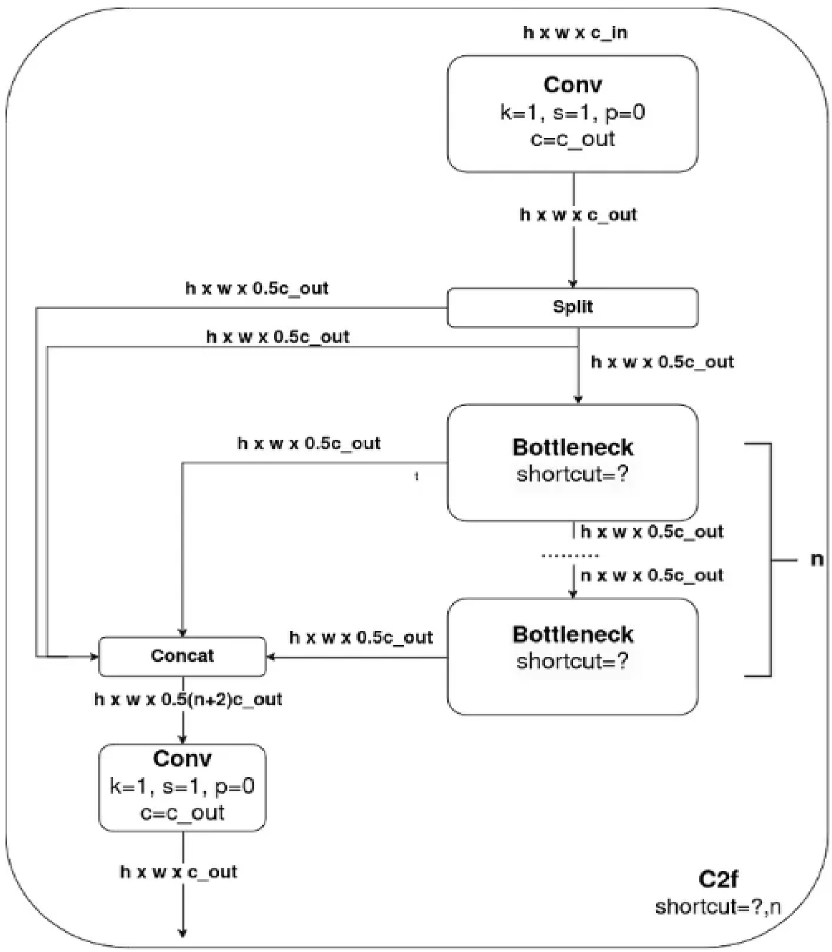


Figure 2.14: C2f Block [82]

6) Spatial Pyramid Pooling Fast (SPPF) Block:

The SPPF block starts with a convolutional layer, followed by three MaxPool2d layers. Each MaxPool layer processes the same input but at different positions, producing separate feature maps. These outputs are then combined (concatenated) and passed through a final convolutional layer.

The concept behind this design comes from Spatial Pyramid Pooling, which breaks the image into grids and pools features from each grid separately. This helps the model handle images of different sizes and capture features at multiple scales, which is important for recognizing objects that may appear larger or smaller in different images.

While traditional SPP can be slow due to its multiple layers and varying kernel sizes, the SPPF version simplifies the process using fixed-size pooling kernels. This speeds things up while still capturing essential multi-scale information.

MaxPool2d plays a key role by reducing the size of the feature map while keeping the most important information. It works by taking the highest value from each region of the input, which helps highlight strong features and reduces the amount of data the network has to process.

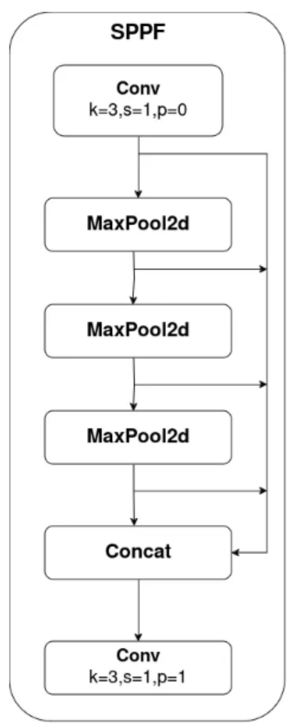


Figure 2.15: SPPF Block [82]

7) Detect Block:

The Detect block is the part of the model that identifies and locates objects in an image. In YOLOv8, unlike earlier versions, detection is anchor-free, meaning the model directly predicts the centre of each object rather than calculating its position based on predefined anchor boxes. This approach simplifies the process and speeds up post-processing by reducing the number of predicted boxes to evaluate.

Inside the Detect block, there are two parallel paths. One path is used to predict the bounding boxes, while the other is used to predict the object classes. Each path includes two convolutional layers, followed by a final Conv2d layer that outputs either the bounding box loss or the class loss, depending on the track. This design helps the model detect and classify objects efficiently and accurately.

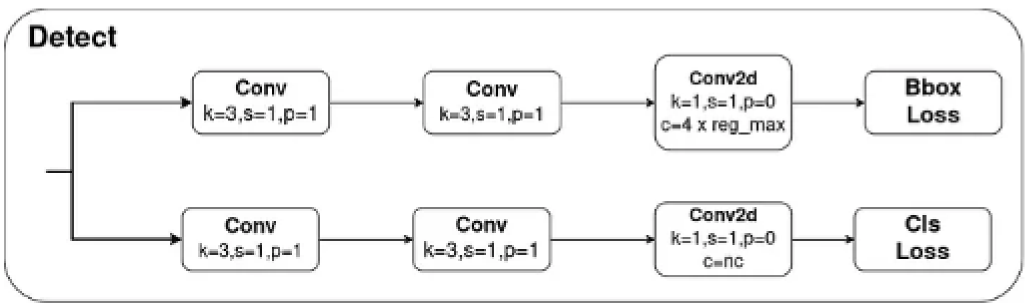


Figure 2.16: Detect Block [82]

2.3.4 Vehicle Tracking and Counting using SORT:

In our project, SORT (Simple Online and Realtime Tracking) is used to track detected vehicles across video frames. It assigns unique IDs to each vehicle, enabling accurate counting and movement monitoring in real time.

2.3.4.1 What is SORT:

SORT is a lightweight, real-time algorithm designed for multiple objects tracking in video sequences. It operates by associating object detections across consecutive frames to maintain consistent identities for each object over time.

2.3.4.2 How SORT works:

The core idea behind SORT is straightforward: it uses a Kalman Filter to predict where an object will move next, and then compares that prediction with the actual detections to track the object over time.

This tracking process is organized into four main stages **[85][86]**:

- **Detection**: An object detection model is first used to identify and localize objects in the current frame via bounding boxes.
- **Prediction**: The Kalman Filter predicts the next positions of the previously tracked objects based on their motion history.
- **Association**: The predicted positions are matched with new detections using an association algorithm (often based on the Intersection-over-Union metric or Hungarian algorithm) to maintain object identity.
- **Correction**: Once matched, the Kalman Filter corrects its predictions using the actual observed detections, improving the accuracy of the tracking.

2.3.4.3 Kalman Filters:

The Kalman Filter is a method used to estimate the state of a changing system — in our case, the position of a moving object. It uses equations that describe how the object is expected to behave, and then improves that estimate using real but noisy measurements (like object detections from a camera).

Kalman Filters work best with systems that follow a straight-line motion (linear systems). In SORT, we assume that the object moves with a constant speed, meaning its velocity doesn't change much — only small changes caused by random noise. Using this assumption, the filter can predict where the object will be next. This part is called the prediction step. The state of the object at each time step includes:

- The object's position (u and v, which are its coordinates on the screen).
- The size of the object (called scale), and the shape of the bounding box (aspect ratio).
- The object's speed in both horizontal and vertical directions.
- And how the scale might change slightly over time.

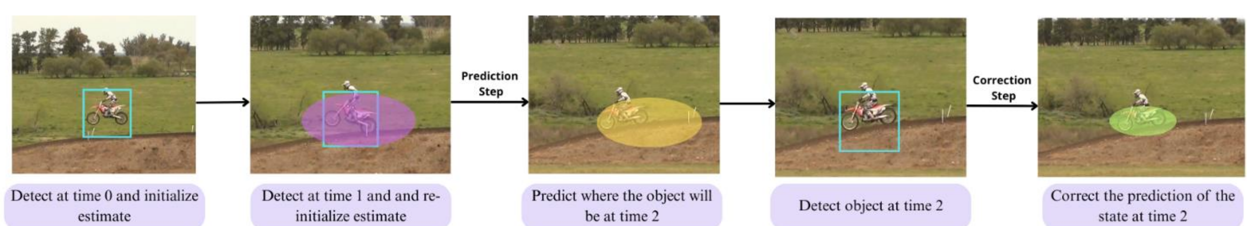


Figure 2.17: Overview of the Kalman filter functioning in SORT [85].

2.3.5 Intelligent Decision-Making in Our Control System:

In our smart traffic system, the "brain" decides which lane gets the green light. It uses live data like how many cars are waiting and if there's an emergency vehicle. The system follows clear rules to make fair and quick decisions. Below the step-by-step explanation of our decision-making (Brain) (see figure 2.18):

- **Receive Inputs**

  The system first takes in real-time traffic data:

  - The number of vehicles waiting in each direction (North-South and East-West).
  - The total waiting time of those vehicles.

- **Emergency Vehicle Check**

  If an emergency vehicle is detected in any direction, that lane is immediately given the green light. This ensures fast response to critical situations.

- **Empty Lane Check**

  If one direction has no vehicles at all, the system skips it and gives the green light to the other active direction to save time.

- **Starvation Condition**

  If a direction has not been served for 50 seconds or more and still has waiting vehicles, it is prioritized to prevent it from being ignored for too long.

- **Scoring Both Directions**

  When both directions have traffic, the system calculates a score for each one:

  $$Score = number\ of\ cars + (0.5 \times total\ waiting\ time) \quad \ldots\ldots\ldots\ldots(2.1)$$

  This helps balance urgency and traffic load.

- **Fairness Rule**

  To avoid favoring the same direction repeatedly, the score of the last served direction is slightly reduced. This promotes equal treatment over time.

- **Select Direction with Higher Score**

  The direction with the highest adjusted score is chosen to receive the green light next.

- **Calculate Green Time**

The system then calculates how long the green light should last using:

$$Green\ Time\ =\ car\ count\ \times\ 1.2\ seconds\ \ldots\ldots\ldots..(2.2)$$

The value is limited to a minimum of 6 seconds and a maximum of 40 seconds.

– **Apply Green Signal**

Finally, the chosen direction is given the green light for the calculated duration. Then the cycle repeats with updated data.

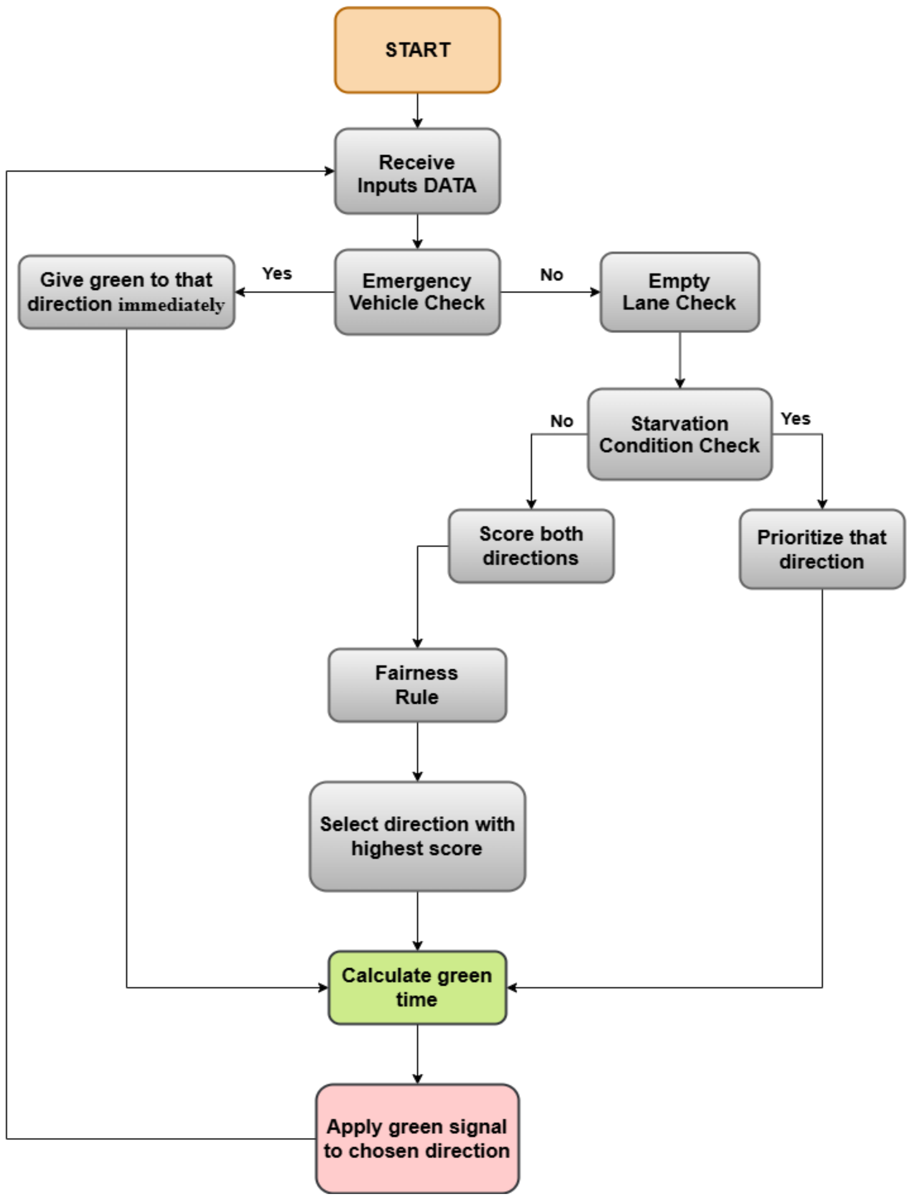


Figure 2.18 Architecture of The Proposed Method

2.4 Conclusion

The growing complexity of urban traffic requires smarter, data-driven approaches to manage the increasing number of vehicles on the roads. Automatic Traffic Light Control Systems represent a fundamental shift from traditional time-based signals to adaptive, real-time solutions that respond dynamically to road conditions. These systems rely on integrated sensing technologies, communication networks, and intelligent algorithms to make timely and effective decisions.

This chapter provided a detailed overview of automatic traffic light systems, emphasizing the role of intelligent sensors, real-time data acquisition, and adaptive decision-making techniques in modern urban mobility. From communication technologies to predictive algorithms and actuation mechanisms, we explored the fundamental elements that drive automated control. These insights lay the groundwork for the practical realization and testing of our proposed Intelligent Traffic Signal System, which will be explored in Chapter 3, focusing on experimental setup and performance evaluation.

CHAPTER 3:

System Implementation and Evaluation

3.1 INTRODUCTION:

Following the theoretical foundation outlined in the previous chapter, this chapter presents the implementation and evaluation of our Intelligent Traffic Signal System (ITSS). We begin by describing the training of the YOLOv8 deep learning model for vehicle detection, followed by the integration of our system in a simulated environment using SUMO and real-time detection algorithms. This chapter also outlines the metrics used for performance evaluation and compares our method with existing techniques to demonstrate its effectiveness in traffic optimization.

3.2 DEVELOPMENT TOOLS:

This section highlights the main software and hardware tools used to develop the system. Each tool played a critical role in coding, testing, and ensuring system performance.

3.2.1 Hardware used:

My PC which has the following features was used to complete this work:

• CPU: Ryzen 5 5600G

• RAM: 16

• GPU: VEGA 7

3.2.2 Software:

In this section, we present the tools and software used to build and run the system, together with the development environment and other auxiliary applications.

3.2.2.1 Operating System:

For AI and image and video processing applications, Windows10 64-bit is a suitable option. due to its strong compatibility with development environments like Visual Studio Code and its extensive support for programming libraries that rely on GPU processing, such as Python, PyTorch, CUDA and Sumo.

3.2.2.2 Visual Studio Code (VS Code):

Visual Studio Code is a contemporary tool for developing and editing code. Coding is made simpler and more effective with this tool's many features that are employed in the software development process. open-source and cost-free. Windows, Linux, and macOS are among the operating systems with which it is compatible. Visual Studio Code requires little space and is simple to install. Python, Java, C++, and other programming languages are supported. Programmers can write understandable code thanks to Visual Studio Code's intuitive interface. It makes programming easier as well. The user can download libraries from the internet and incorporate them into the code as needed, allowing them to customize the editor to suit their needs **[87].**

Figure 3.1 Logo of Visual Studio Code

3.2.2.3 Python:

Python is a programming language that is frequently used in data analysis, machine learning (ML), software development, and online apps. Python, developed by Guido van Rossum and published in 1991, is used by developers due to its efficiency, ease of learning, and compatibility across a wide range of platforms. Python surpasses C, C++, Java, and JavaScript in numerous domains, making it one of the most popular and extensively used programming languages worldwide **[88].**

Figure 3.2 Logo of Visual Studio Code [89].

3.2.2.4 CUDA Toolkit and CUDNN:

1) **CUDA Toolkit:** Compute Unified Device Architecture, a.k.a. CUDA is a parallel computing platform developed by NVIDIA with an initial release date of 23 June 2007. NVIDIA CUDA revolutionized how GPUs are used for general-purpose computing (GPGPU). NVIDIA CUDA made it possible to use GPUs for various applications, including scientific research, engineering simulations, and, eventually, AI and deep learning. By around 2015, the development of CUDA's focus shifted towards neural networks and AI **[90].** Through the use of specialized NVIDIA technology, CUDA enabled us to execute computations more quickly, which helped to shorten the training period and enhance the model's overall performance. Since training huge models on the

CPU takes a lot of time, we chose PyTorch because it is a library that supports CUDA to get the most out of it and speed up our model's training time.

2) **CuDNN:** CUDNN (CUDA Deep Neural Network library) is a specialized, GPU-accelerated library that provides essential building blocks for deep neural networks. It is intended to supply high-performance parts for recurrent neural networks (RNNs), convolutional neural networks, and other intricate deep learning methods. Frameworks like TensorFlow and PyTorch can benefit from enhanced GPU performance by integrating CUDNN. While CUDNN offers specific tools for deep learning, NVIDIA's CUDA installation establishes the foundation for GPU computation. For jobs that a conventional CPU may otherwise take days or weeks to finish, this combination allows for amazing GPU acceleration **[91].**

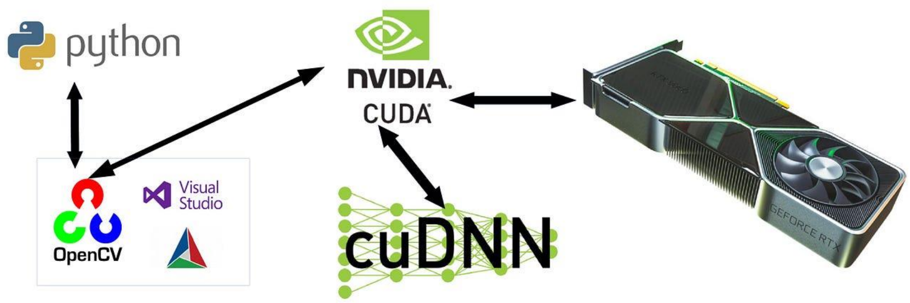


Figure 3.3 Accelerating Neural Networks Using CUDA and CUDNN on NVIDIA GPUs

3.2.2.5 Google Colaboratory (Google Colab):

Google Colaboratory, or Collab, is an as-a-service version of Jupyter Notebook that enables you to write and execute Python code through your browser. Jupyter Notebook is a free, open source creation from the Jupyter Project. A Jupyter notebook is like an interactive laboratory notebook that includes not just notes and data, but also code that can manipulate the data. The code can be executed within the notebook, which, in turn, can capture the code output **[92].**

Enterprise data analysts and analytics developers can use Colab to work through data analytics and manipulation problems in collaboration. They can write, execute and revise core code in a tight loop, developing the documentation in Markdown format, LaTeX or HTML as they go.

Notebooks can include embedded images as part of the documentation or as generated output. In addition, you can copy finished analytics code, with documentation, into other platforms for production use once sufficiently tested and debugged.

Google Colab eliminates the need for complex configuration setup and installation, as it runs right in the browser. It also includes pre-installed Python libraries that require no setup to use.

Figure 3.4 Logo of Google Colaboratory

3.2.2.6 **SUMO**:

Simulation of Urban Mobility", or "SUMO" for short, is an open source, microscopic, multi-modal traffic simulation. It allows to simulate how a given traffic demand which consists of single vehicles moves through a given road network. The simulation allows to address a large set of traffic management topics. It is purely microscopic: each vehicle is modelled explicitly, has an own route, and moves individually through the network. Simulations are deterministic by default but there are various options for introducing randomness **[93].**

Figure 3.5 Logo of sumo

3.2.3 Libraries Used:

3.2.3.1 Ultralytics:

Ultralytics is an open-source Python library that provides state-of-the-art deep learning models and tools for computer vision, including object detection, segmentation, and classification. It is best known for developing the YOLO (You Only Look Once) series, especially the YOLOv5 and YOLOv8 models, and offers a high-level API for training, validating, and deploying these models easily **[94].**

Figure 3.6 Logo of Ultralytics

3.2.3.2 PyTorch:

PyTorch is an open-source machine learning (ML) framework based on the Python programming language and the Torch library. Torch is an open-source ML library used for creating deep neural networks and is written in the Lua scripting language. It's one of the preferred platforms for deep learning research **[95].** PyTorch provides a Python package for high-level features like tensor computation (like NumPy) with strong GPU acceleration and Torch Script for an easy transition between eager mode and graph mode. With the latest release of PyTorch, the framework provides graph-based execution, distributed training, mobile deployment, and quantization.

Figure 3.7 Logo of PyTorch

3.2.3.3 OpenCV:

Open-Source Computer Vision, or OpenCV for short, took form in 1999 by Intel. It is a free cross-platform Computer Vision library for real-time image processing. The purpose? It is used for building Deep Learning and Machine Learning applications, predominantly for classical computer vision applications. With more than 2,500 optimized algorithms, including classic and state-of-the-art Computer Vision and Machine Learning algorithms, it is used in object detection, facial detection, 3D model extraction, and the list goes on. Although it was initially developed in C/C++, it is actively developed for Python, MATLAB, Ruby, and other languages **[96].**

Figure 3.8 Logo of OpenCV

3.2.3.4 TraCI:

TraCI (Traffic Control Interface) is a TCP-based client-server architecture that enables bidirectional communication between the traffic simulation software SUMO and external applications, allowing for the control and manipulation of simulation elements (e.g., traffic lights, vehicles) during runtime **[97].**

Figure 3.9 Logo of TraCI

3.2.3.5 OIDv4 Toolkit:

The OIDv4 Toolkit is an open-source Python library designed to simplify the process of downloading and organizing images and annotations from the Open Images Dataset v4 (OIDv4). This toolkit is particularly useful for machine learning practitioners working on object detection and image classification tasks, as it allows for selective downloading of specific classes, thereby avoiding the need to handle the entire, massive dataset **[98].**

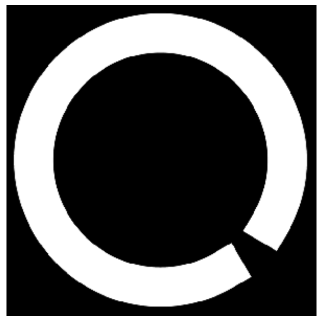


Figure 3.10 Logo of OIDv4 Toolkit

3.3 Training Our Model Using YOLOv8:

To train a custom YOLOv8 model, we used Google Colab, a cloud-based environment that offers GPU acceleration for deep learning. Below is a summary of the training process:

3.3.1 Dataset Preparation

Open Images V4 is a large-scale dataset of ~9 million images that are annotated with image-level labels, object bounding boxes, and visual relationships. It contains 15.4 million bounding boxes for 600 object classes on 1.9 million images, making it one of the largest publicly available datasets for object detection **[99].** We used the OIDv4 Toolkit to download 1000 image for each class of the five target classes: cars, buses, trucks, motorcycles, and ambulances. The dataset was then organized following YOLO format, with annotated labels saved as .txt files corresponding to each image.

3.3.2 Model selection

We chose YOLOv8n (the nano version) for its speed and lightweight structure, making it suitable for real-time applications. Since the pre-trained model was trained on the COCO dataset, which does not include an "ambulance" class, we re-trained the model from scratch to include this additional category.

3.3.3 Training setup in Colab:

We uploaded the dataset and the data.yaml then we configured our training environment with the following hyperparameters:

- Epochs: 200
- Image size: 640
- Batch size: 16
- Optimizer: AdamW
- Learning rate: 0.001

```yaml
train: images
val: images

nc: 5
names: ["Car", "Truck", "Bus", "Ambulance", "Motorcycle"]

```

Figure 3.11: Data.yaml

3.3.4 Model Training

Using the Ultralytics YOLOv8 library, we executed the training command:

```python
from ultralytics import YOLO

model = YOLO('yolov8n.pt')  # or path to custom checkpoint
model.train(
    data='/content/dataset/data.yaml',
    epochs=200,
    imgsz=640,
    batch=16,
    optimizer='AdamW',
    lr0=0.001,
    augment=True,
    device=0,
    project='runs/train',
    name='custom_yolov8_training'
)
```

Figure 3.12: Training command

3.3.5 Output and Model Saving

After training, the model weights (best.pt and last.pt) were saved in the specified project folder. These were later used for vehicle detection in our Intelligent Traffic Signal System (ITSS).

3.4 Implementation of the Proposed Smart Traffic Light System

We built smart traffic light system proposed step by step, adding one part at a time to make sure it worked correctly and responded to real traffic situations. Below is a clear explanation of how we created the system by two methods.

In the first approach, data acquisition was carried out using cameras installed in all four directions at the intersection (see figure 3.13.a). In contrast, the second method employed the Simulation of Urban Mobility (SUMO) tool to generate synthetic traffic data through a custom-designed simulation environment (see figure. 3.13.b).

Figure 3.13 data acquisition, (left) using cameras, (right) using SUMO.

3.4.1 Step-by-Step using cameras:

The steps of the smart traffic light system proposed using cameras for data acquisition and our proposed hybrid rule-based method for decision making.

1. **Setting Up the Video Input**
   First, we added a system to load video footage from four directions: north, south, east, and west. We used the Video Feed module to read the videos and keep them in sync, just like having real-time cameras at an intersection.
2. **Adding Vehicle Detection**
   Next, we used a deep learning model called YOLOv8 to detect vehicles in each video frame. This was handled by the Vehicle Detector class. We trained the model to detect five types of vehicles: cars, buses, trucks, motorcycles, and ambulances. This allowed the system to see what types of vehicles were on the road.
3. **Tracking Vehicles in Real Time**
   Then, we added a tracking system using an algorithm called SORT. This helped us follow each vehicle from one frame to the next, giving each one a unique ID. It made it possible to count vehicles and understand traffic flow correctly.
4. **Organizing Tracking by Direction**
   We created the Multilane Tracker to keep track of vehicles separately in each direction (north, south, east, west). This also helped the system notice if an ambulance was present in any lane, which is important for priority handling.
5. **Collecting Traffic Statistics**
   After that, we used the Traffic State Tracker to collect important data such as how many vehicles were waiting in each direction and for how long. It also kept track of whether there was an ambulance. This information was needed for smart decision-making.
6. **Making Smart Traffic Decisions**
   We then created the system's "brain" using the Traffic Brain module. It used the statistics to decide which direction should get the green light and for how long. The decision was based on how many vehicles were waiting, how long they had waited, and if an ambulance needed to pass.
7. **Controlling the Traffic Lights**
   We added a controller called Light Controller that actually changed the traffic lights based on the decisions from the brain. It handled the switch between red, yellow, and green lights.

8. **Showing Visual Results in Real Time**

   To make it easier to see what was happening, we added a display system using the Visualizer. This showed the vehicles, traffic lights, and decisions on the video in real time. It helped us test and show the system clearly.

9. **Saving System Activity in a Log**

   We created a logging system using TrafficLogger to save everything the system did—such as which direction got green, how long vehicles waited, and how many were detected. This data was saved in a CSV file for later analysis.

10. **Putting Everything Together**

    Finally, we connected all the modules inside the main.py script. This made everything work together—from reading video input to detecting vehicles, tracking them, making decisions, changing lights, and showing the results on screen.

Figure 3.14 the Organizational chart of the proposed automatic smart traffic light system.

3.4.2 Step-by-Step Using SUMO and TraCI:

The steps of the smart traffic light system proposed using SUMO for data acquisition and our proposed hybrid rule-based method for decision making.

1. We built our traffic simulation using SUMO. We created a road network using .net.xml and added vehicle routes using files like balanced_high.rou.xml. Then, we used a .sumocfg file to run the simulation with different traffic conditions.
2. To connect our own logic to SUMO, we used TraCI (Traffic Control Interface). It allowed our Python code to control the simulation in real time — reading traffic data (like vehicle count and wait time) and changing the traffic lights directly from our program.
3. We made the system read traffic data every second. At each simulation step, our program read how many vehicles were waiting in each direction (North-South and East-West) and how long they had been waiting. This data came from the modules simulator.py and traffic_metrics.py.
4. We wrote our own smart decision system. We created a hybrid rule-based Methode inside the brain.py file. This part of the code used the traffic data to decide which direction should get the green light and its duration.
5. Finally, we changed the traffic lights using TraCI. Once the decision was made, we used TraCI to send the green light to the chosen direction inside SUMO, and the red light to the other. This happened automatically at every simulation step.

3.5 Monitoring and Evaluation

In addition, evaluating the performance of algorithms is crucial to determine their effectiveness and potential for improvement.

3.5.1 Evaluation metrics of vehicle detection

- **Precision:** Measures how many of the detected objects are actually correct.
  formula:

$$precision = TP / (TP + FP) \qquad (3.1)$$

  - **TP (True Positive):** The model correctly detects an object that is really there.
  - **FP (False Positive):** The model wrongly detects something that isn't there.

- **Recall:** Measures how many of the actual objects were detected.

  formula:

$$recall = TP\ /\ (TP\ +\ FN) \quad \ldots\ldots\ldots\ldots\ldots\ldots..(3.2)$$

  - **TP (True Positive):** The model correctly detects an object that is really there.
  - **FN (False Negative)**: The model **misses** an object that is really there.
- **mAP@0.5:** Average Precision at IoU threshold 0.5 This tells how good the model is at detecting objects when the match only needs to be 50% accurate.
- **mAP@0.5:0.95:** Averaged mAP across IoU thresholds from 0.5 to 0.95 (step 0.05 These checks how good the model is across different levels of strictness, from easy (50%) to very strict (95%).

3.5.2 Evaluation metrics of smart traffic systems

An essential part of smart systems is continuous monitoring and learning from past decisions.

The Performance Metrics are:

- **Average Delay Time:** Calculates the difference between actual travel time and expected (free-flow) travel time **[100]**.

  Formula:

$$DT = \sum_{i=1}^{N}(ActualTime_i - FreeFlowTime_i) \quad \ldots\ldots\ldots\ldots\ldots\ldots..(3.3)$$

  - Actual Time *i* : Actual travel time of vehicle *i*
  - Free Flow Time *i*: Travel time under free-flow conditions for vehicle *I*
- **Average Waiting Time:** Measures how long vehicles wait at a red light. Lower values indicate better flow **[101]**.

  Formula:

$$AWT = \frac{\sum_{i=1}^{N} WaitTime_i}{N} \ldots\ldots\ldots\ldots\ldots\ldots..(3.4)$$

  - Wait Time *I* : Waiting time of vehicle *i*
  - N: Total number of vehicles
- **Phase Utilization Efficiency:** Monitors how well green phases are used (e.g., if green time is wasted with no vehicles passing) **[102]**.

  Formula:

$$PUE = \frac{Vehicles\ Passed\ During\ Green}{Max\ Capacity\ During\ Green} \times 100\% \quad \ldots\ldots\ldots\ldots\ldots\ldots..(3.5)$$

  - Vehicles Passed During Green: Number of vehicles that passed during the green phase

- Max Capacity During Green: Maximum number of vehicles that could pass during the green phase

– **Intersection throughput:** Number of vehicles that successfully pass **[102]**.

Formula:

$$\boldsymbol{TP} = \frac{\boldsymbol{N_{out}}}{\boldsymbol{T}} \ldots\ldots\ldots\ldots\ldots\ldots (3.6)$$

- Nout: Number of vehicles that exited the intersection
- T: Total time interval

– **Fairness Index:** Ensures all directions are served fairly without long waiting. Higher is more balanced **[103]**.

Formula:

$$\boldsymbol{F1} = \frac{(\sum_{i=1}^{n} x_i)^2}{n \sum_{i=1}^{n} {x_i}^2} \ldots\ldots\ldots\ldots\ldots\ldots (3.7)$$

- xi: Average delay or wait for direction *i*
- n: Number of directions

– **Fuel consumption and CO_2 emissions:** Estimates environmental impact due to idling and congestion **[104]**.

Formula:

$$VSP = v(1.1 \cdot a + 0.132) + 0.000302 \cdot v^3 \ldots\ldots\ldots\ldots\ldots\ldots (3.8)$$

- v: Instantaneous speed (m/s)
- a: Instantaneous acceleration (m/s^2)

3.6 Experimental Results

3.6.1 Evaluation of The YOLOv8 Model:

This section presents the evaluation results of YOLOv8 vehicle detection model, based on the training and validation metrics collected throughout the training process. The performance is assessed using standard object detection metrics including precision, recall, mAP@0.5, and mAP@0.5:0.95, as well as visual outputs from the training process.

At the final training epoch, the YOLOv8 model achieved the following results (Table 3.1):

Table 3-1Results of vehicle detection using YOLOv8 model

Precision	**Recall**	**mAP@0.5**	**mAP@0.5:0.95**
94.29%	88.41%	94.52%	80.13%

Figure 3.15 shows the Precision-Confidence Curve. This graph visualizes how the model's precision changes with different confidence thresholds for each class (Car, Truck, Bus, Ambulance, Motorcycle), along with the average across all classes (bold blue line). where the X-axis: Confidence threshold (from 0 to 1) and the Y-axis: Precision (how many of the predicted objects were actually correct).

We note that, This Precision-Confidence Curve shows that all object classes achieve high precision as confidence increases. The overall model performs well, reaching a perfect precision of **1.00** at a confidence threshold of **0.941**.

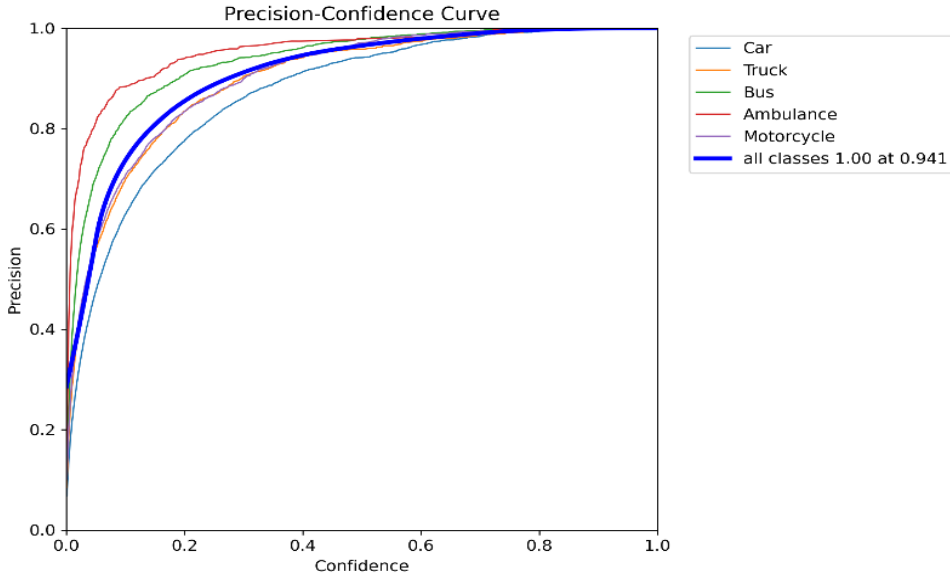


Figure 3.15 Precision vs Confidence Threshold

Figure 3.16 shows the Recall-Confidence Curve. This graph visualizes how the model's precision changes with different confidence thresholds for each class (Car, Truck, Bus, Ambulance, Motorcycle), along with the average across all classes (bold blue line). Where the X-axis: Confidence threshold and Y-axis: Recall (how many of the actual objects the model successfully found).

We note that, This Recall-Confidence Curve shows that recall starts high for all classes but decreases sharply as confidence increases. The overall recall reaches **0.97** at a very low confidence threshold, indicating strong sensitivity at low thresholds.

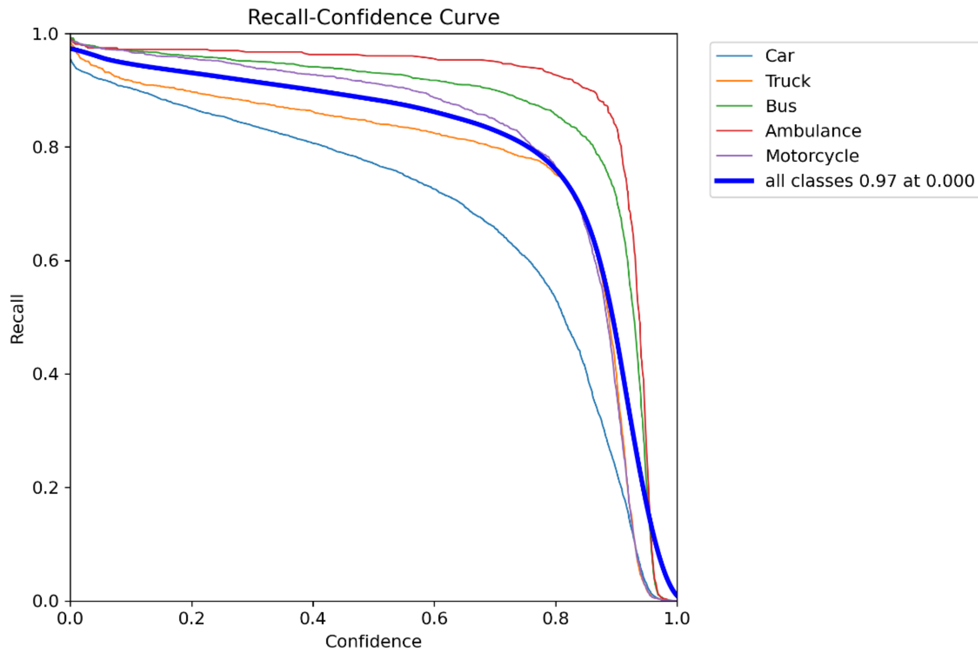


Figure 3.16 Recall vs Confidence Threshold

Figure 3.17 shows the Precision-Recall Curve. This graph displays the trade-off between precision and recall for each class (Car, Truck, Bus, Ambulance, Motorcycle). The bold blue line represents the average performance across all classes. Where the X-axis: Recall – how many real objects the model detected and the Y-axis: Precision – how many of the detections were correct.

We note that, This Precision-Recall Curve shows that all classes maintain high precision across most recall values, especially ambulance and bus. The overall model achieves a strong **mAP@0.5 of 0.953**, indicating excellent detection accuracy.

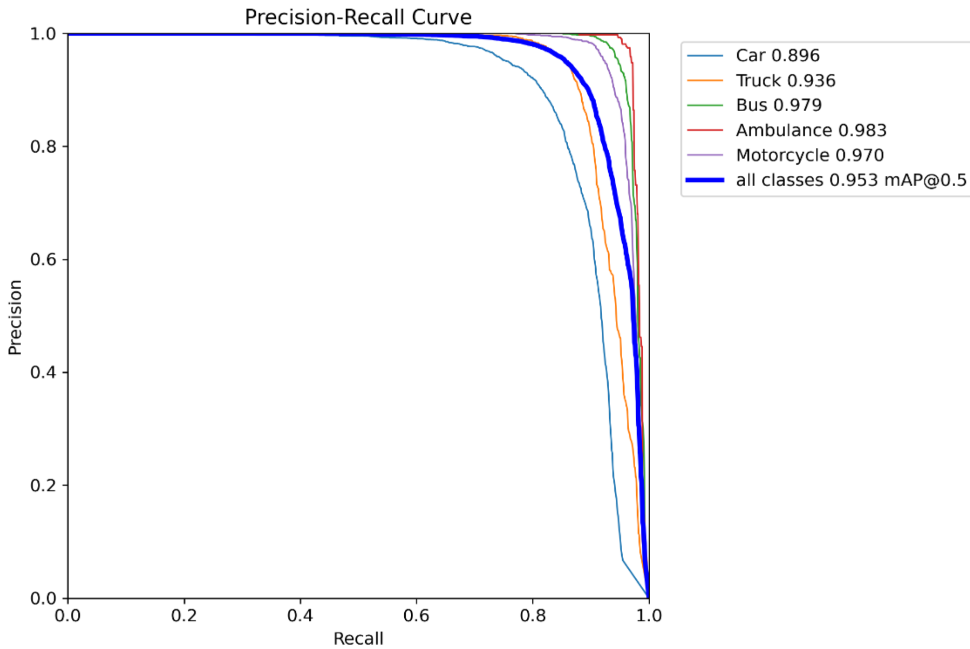


Figure 3.17 Precision vs Recall Curve

Figure 3.18 presents the Training and Validation Metrics Summary. Where these curves illustrate how the YOLOv8 model improved from epoch 156 to 200 across several important metrics. The training and validation losses (box, cls, dfl) consistently decrease, indicating effective model learning. Precision and recall metrics gradually improve, with slight fluctuations in recall. Both mAP@0.5 and mAP@0.5:0.95 show steady growth, confirming improved detection accuracy. Overall, the model demonstrates stable convergence and enhanced performance by epoch 200.

- **Top Row (Training Metrics):**

  1. **train/box_loss:** How accurately bounding boxes were predicted during training.
  2. **train/cls_loss:** Measures how well the model classified each object.
  3. **train/dfl_loss:** Distribution Focal Loss: helps refine box localization precision.
  4. **metrics/precision(B):** Measures how many predicted objects were correct (true positives).
  5. **metrics/recall(B):** Measures how many actual objects were detected.

- **Bottom Row (Validation Metrics):**

  6. **val/box_loss**, **val/cls_loss**, and **val/dfl_loss:** Similar to the training losses but on unseen validation data. These steadily decrease, meaning the model is generalizing well.

7. **metrics/mAP50(B):** Mean Average Precision at 50% IoU: a key accuracy score.
8. **metrics/mAP50-95(B):** Average precision over multiple IoU thresholds (stricter), showing strong performance.

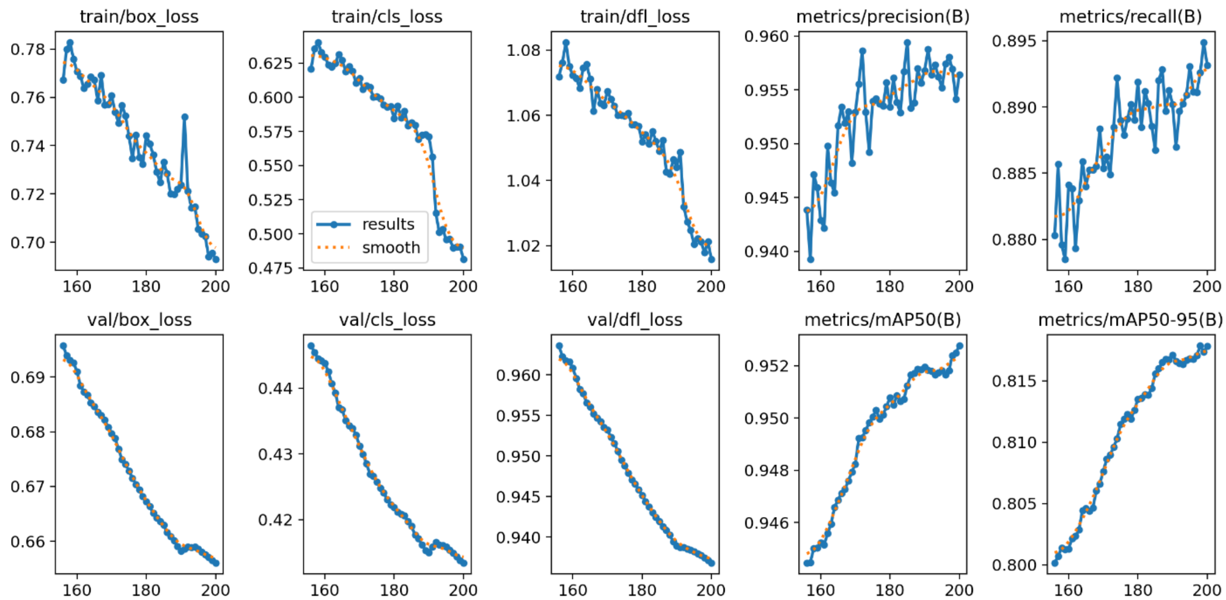


Figure 3.18 Evolution of Losses and Performance Metrics During Training

Figure 3.19 presents the Normalized Confusion Matrix of YOLOv8 Model Predictions. This figure displays the accuracy of class predictions made by the YOLOv8 model, using a normalized confusion matrix. Each row corresponds to the actual (true) class, and each column to the predicted class.

- The darker the square, the better the model performed for that pair.
- The diagonal values show how well each class was correctly predicted.
- The off-diagonal values show misclassifications or confusion between classes.

in Figure 3.19 we observe that the model demonstrates high classification accuracy for most classes, with **Bus (0.96)**, **Ambulance (0.96)**, and **Motorcycle (0.95)** achieving the best results. The **Car** class shows slightly lower accuracy (**0.85**) and significant confusion with the **background** class, where **17%** of background instances were incorrectly predicted as cars. The **Truck** class also exhibits solid performance (**0.87**), with minor misclassifications into **Car (1%)** and **background (15%)**. Overall, the model performs reliably, though further refinement may be needed to better distinguish between **vehicles** and **background**, especially in complex scenes.

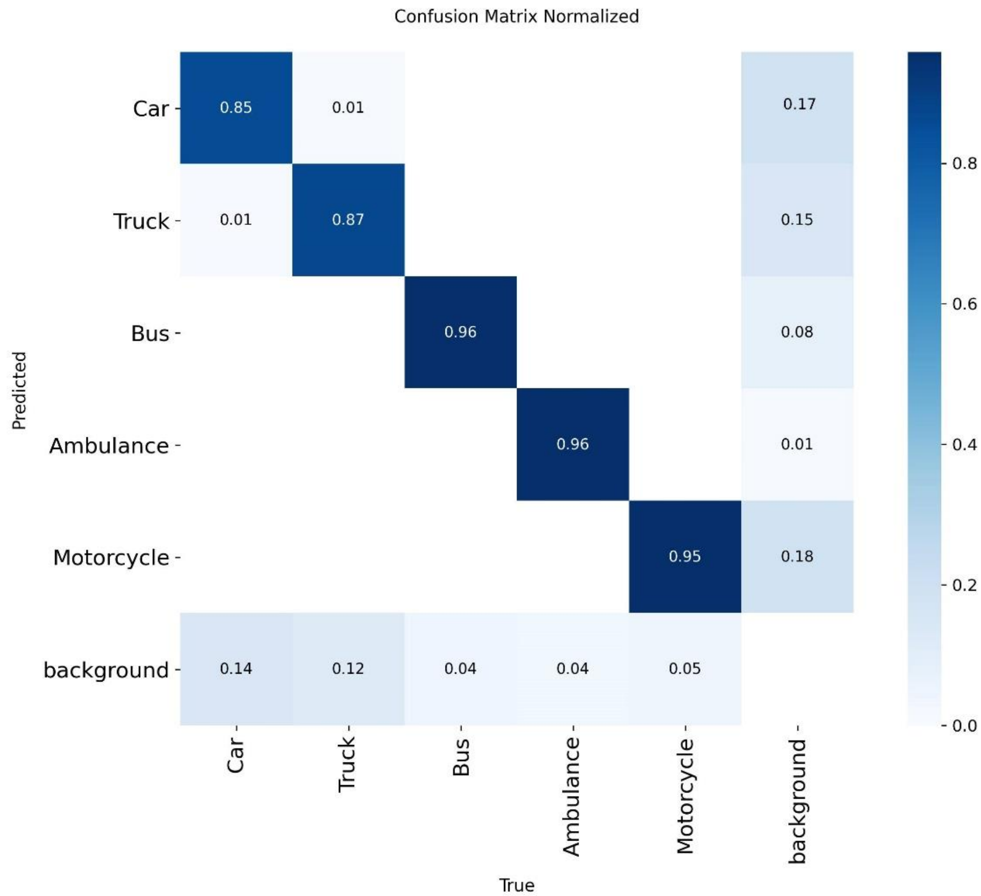


Figure 3.19 Normalized Confusion Matrix of YOLOv8 Model Predictions

3.6.2 Results of Smart Traffic Light System Proposed Using Cameras

On the Smart Traffic System Interface figure: The interface displays real-time vehicle detection across four different road scenes, using bounding boxes to identify and track each car. Vehicles with green bounding boxes indicate that the lane is currently on a green signal, while those with red bounding boxes represent lanes under a red light. The system also shows real-time indicators for each lane, including the signal status, number of detected vehicles, and their waiting time.

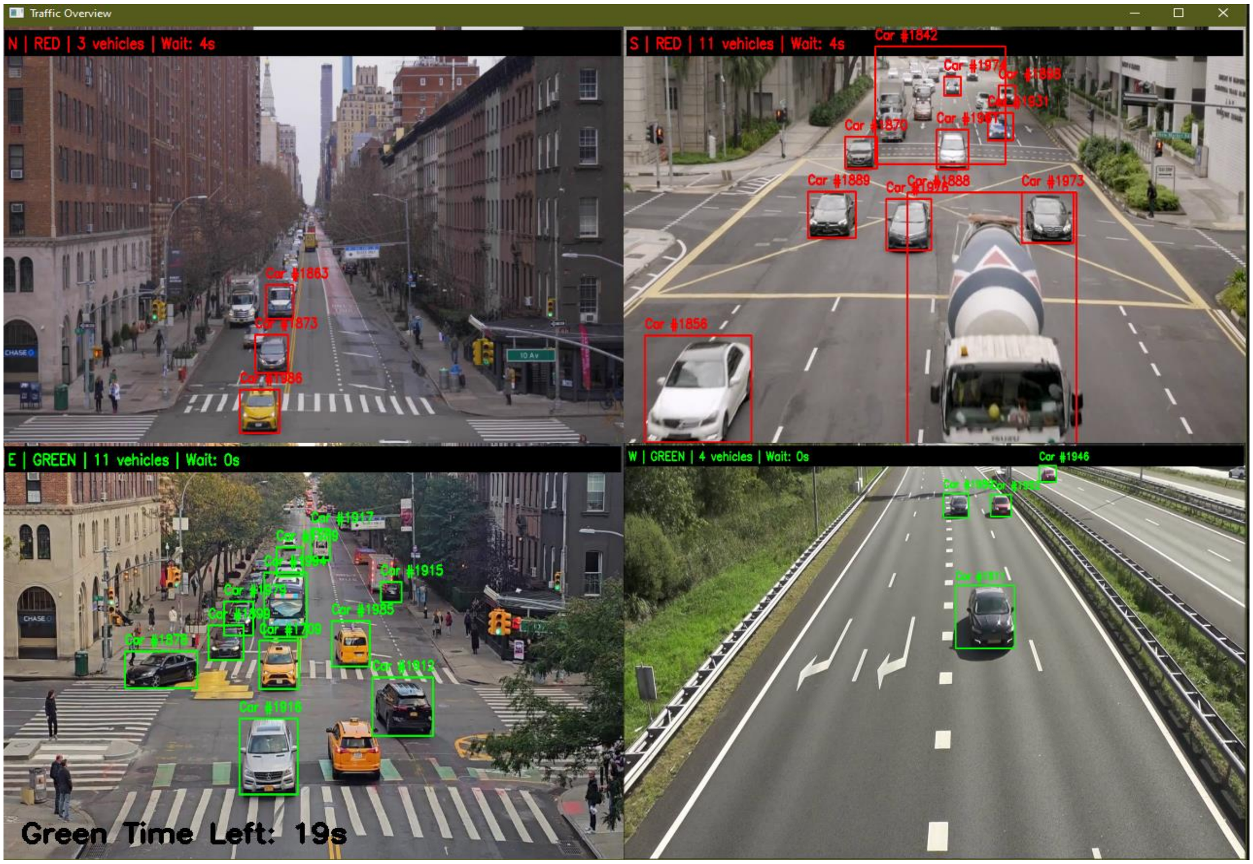


Figure 3.20 Traffic Control interface

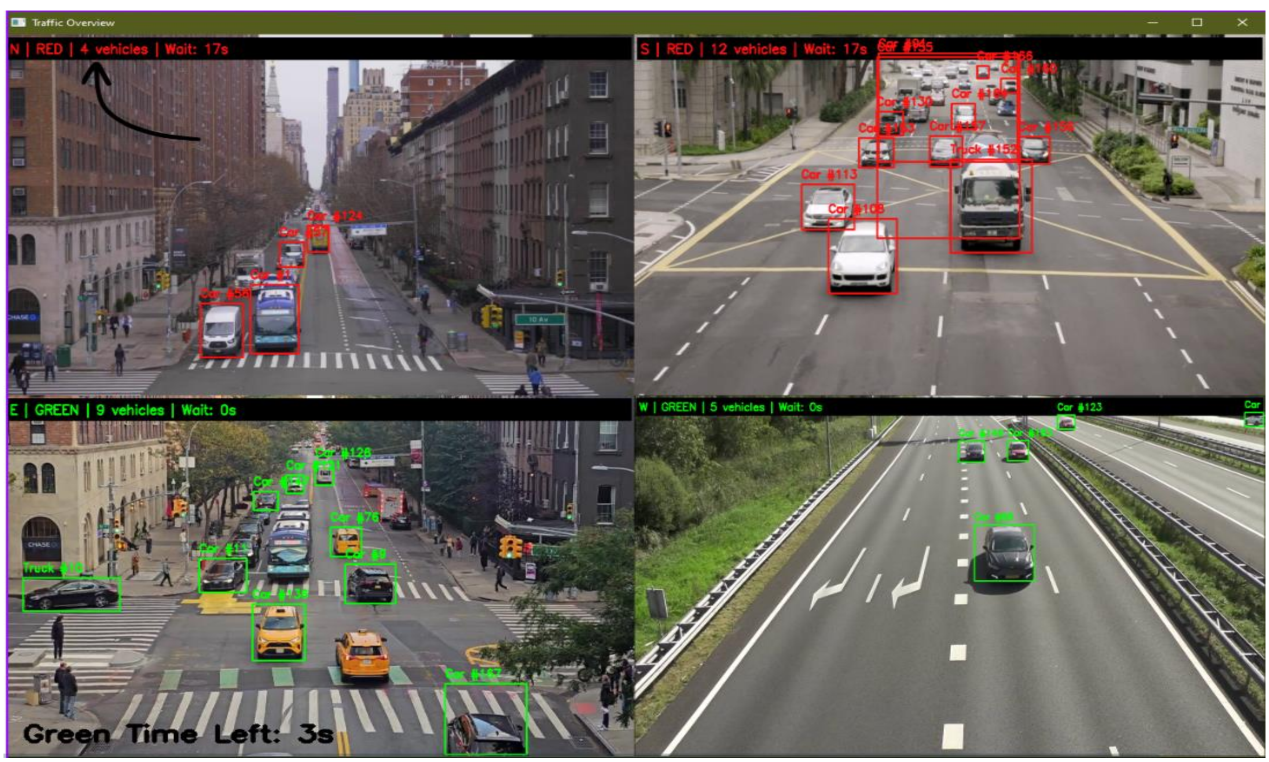


Figure 3.21 Traffic Control interface results

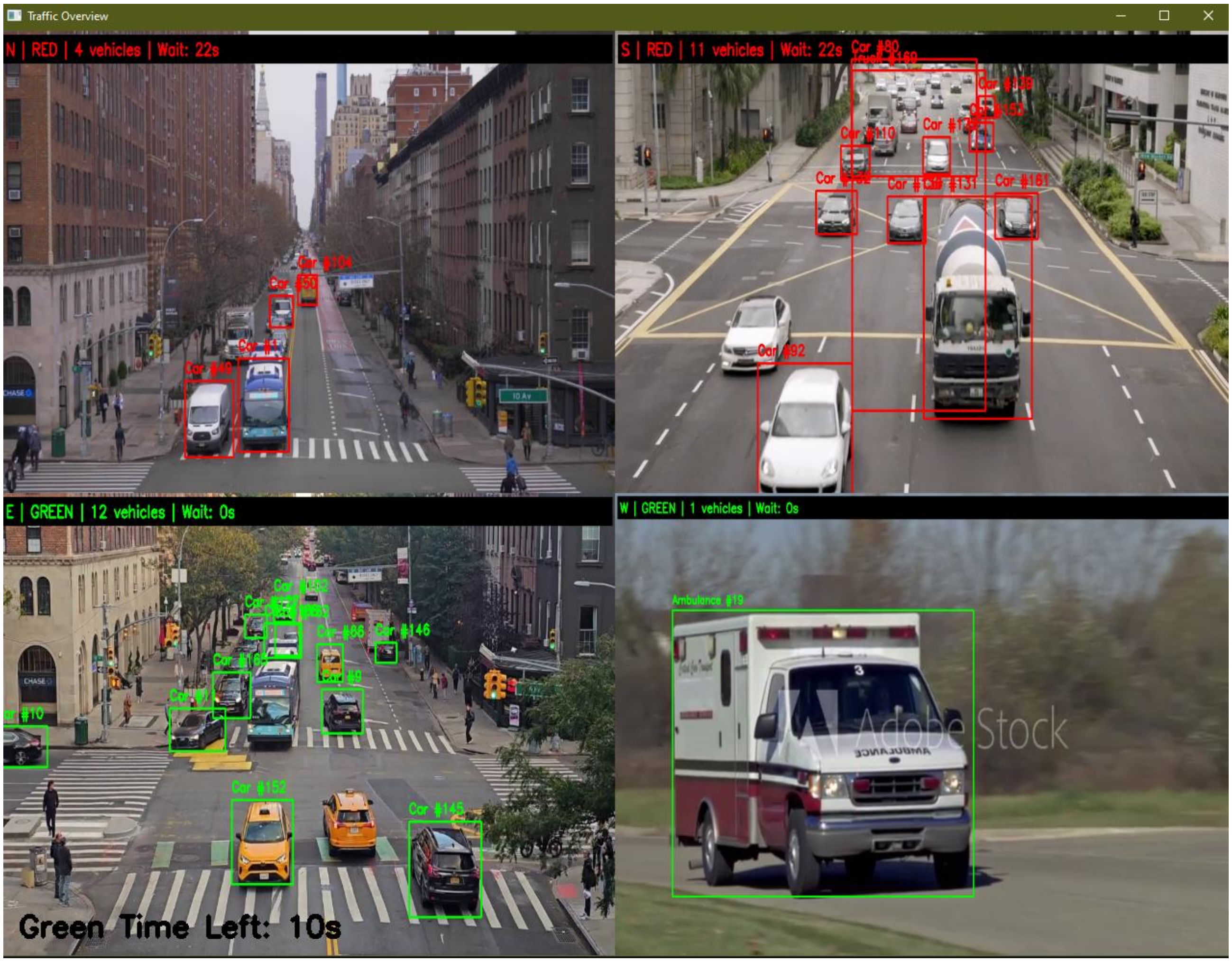


Figure 3.22 Traffic Control in case of Ambulance detected

3.6.3 Evaluation of Smart Traffic Light System Proposed Using SUMO:

To test how well our smart traffic light system performs in a realistic environment, we integrated it with SUMO (Simulation of Urban Mobility). SUMO allowed us to simulate different traffic conditions and analyze how our decision logic handles them in real-time. Our system's performance evaluation is conducted in comparison to the default fixed-time signal logic used in SUMO. This allows us to measure improvements in traffic flow, vehicle waiting time, and responsiveness under dynamic traffic conditions.

Table 3.2 illustrates the Performance of Smart Traffic Control proposed using the evaluation metrics of smart traffic light. We note that, the hybrid rule-based method shows strong performance across key traffic metrics. The average wait time and delay are significantly reduced to 38.3% and 34.5%, respectively, indicating improved traffic flow. The throughput is high at 93%, reflecting efficient vehicle movement through intersections. Green time utilization reaches 70%, while phase efficiency and fairness are also strong at 90% and 96.3%, ensuring balanced and effective signal distribution. Additionally, the system achieves notable environmental benefits with a 25% fuel reduction and a 31% CO_2 reduction, demonstrating both traffic and ecological improvements.

Table 3-2 Performance of Smart Traffic Control proposed

Metric	Hybrid Rule-Based Methode (%)
Average Wait	38.3%
Delay	34.5%
Throughput	93%
Green Time	70%
Phase Efficiency	90%
Fairness	96.3%
Fuel Reduction	25%
CO_2 Reduction	31%

Figure 3.23 presents the performance of the proposed hybrid rule-based method, illustrating our system's results in a clear plot-based format across key traffic and environmental metrics.

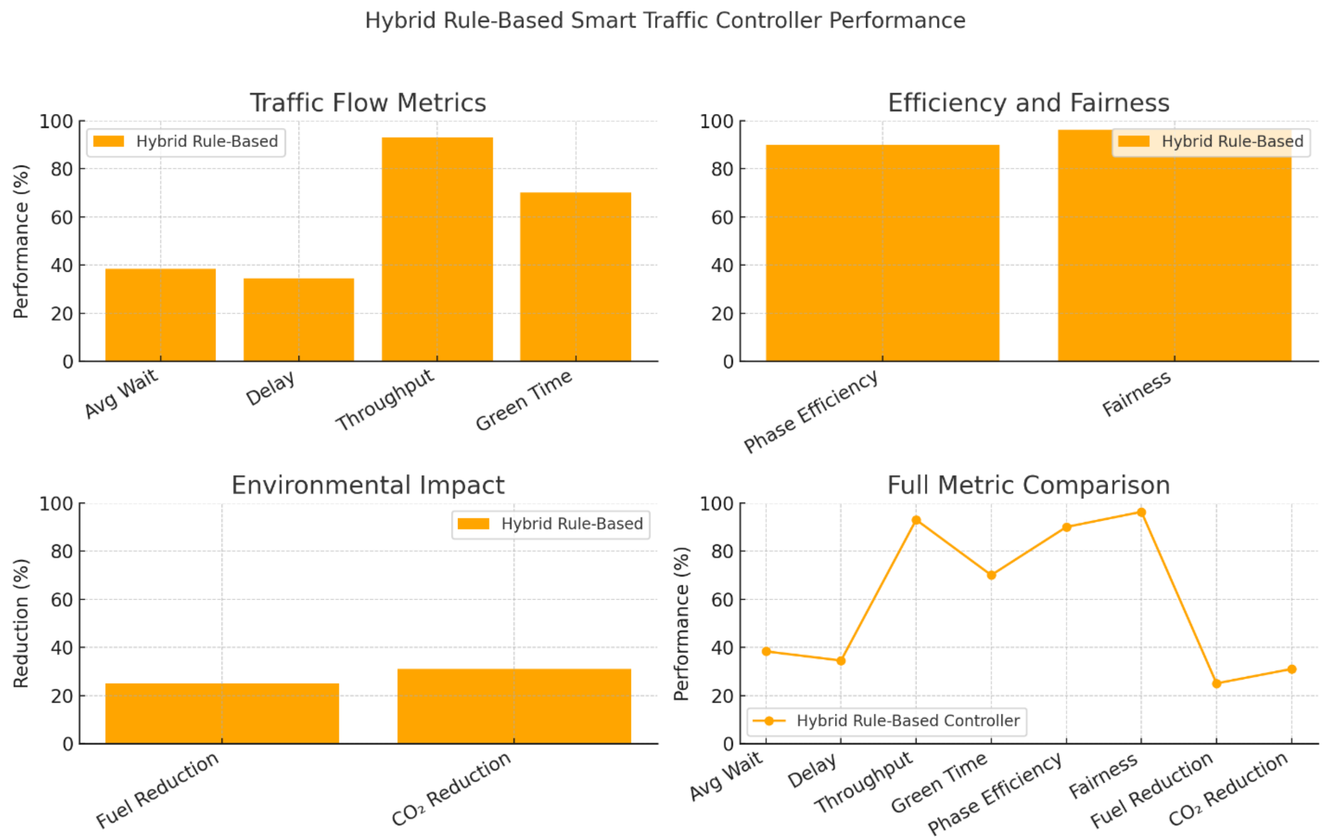


Figure 3.23 Performance of Hybrid Rule-Based Method proposed

The presentation of traffic light using SUMO in figure below.

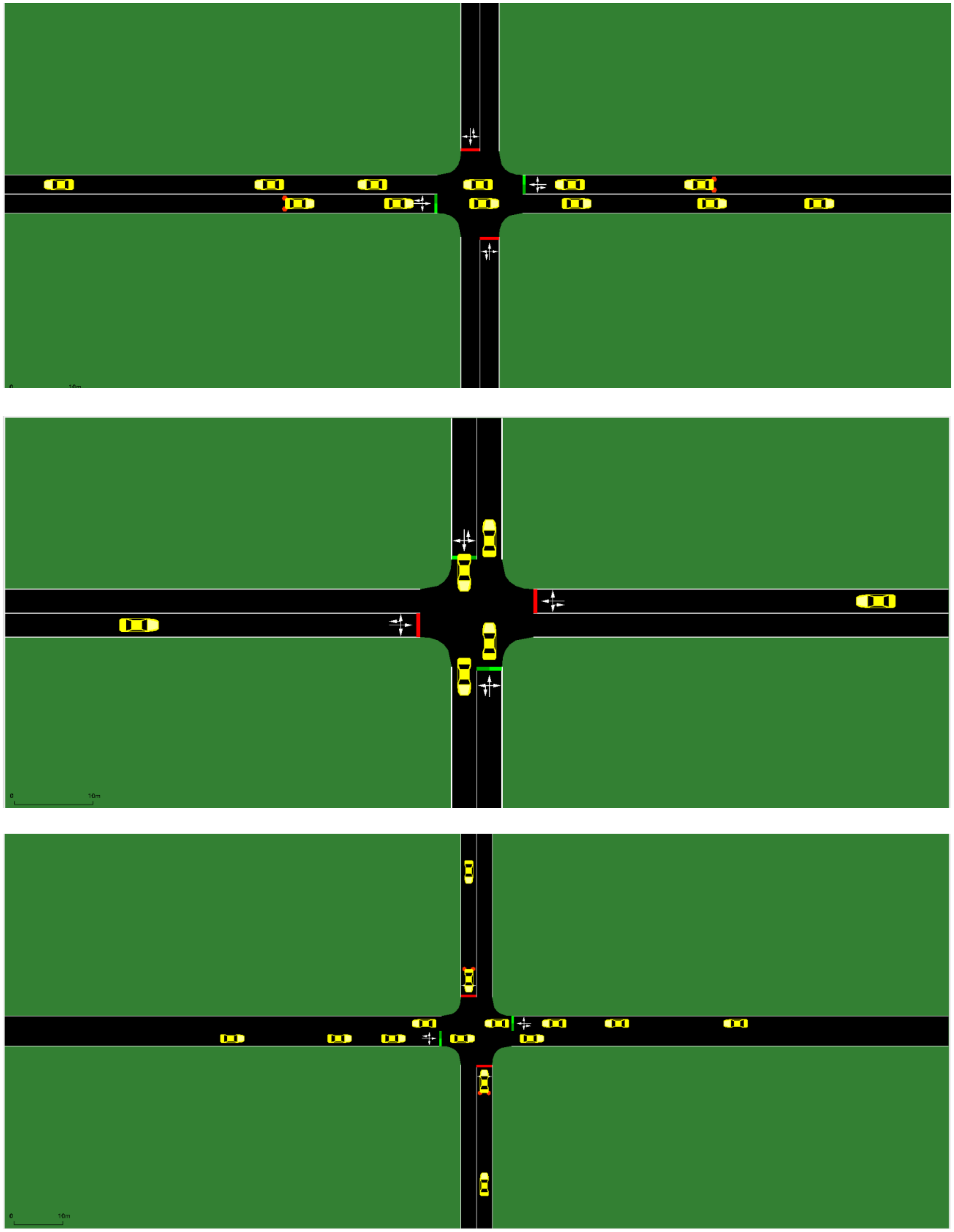


Figure 3.24 Simulation Using Sumo

- **Results using table log:**

Table 3-3 Table of logs

Step	Direction	Green Duration	Cars NS	Cars EW	Wait NS	Wait EW
0	NS	6	2	2	2	2
10	EW	6	4	4	3	3
20	EW	9	6	8	7	10
33	NS	12	10	10	15	15
49	EW	19	8	16	5	26
72	NS	16	14	10	30	10
92	EW	18	8	15	9	32
114	NS	16	14	12	40	15
134	EW	21	7	18	0	55

Table 3-3 displays the performance of the traffic signal system over multiple simulation steps, showing how it responds to varying traffic conditions in both the North-South (NS) and East-West (EW) directions.

- Step indicates the simulation time at which each signal phase is recorded.
- Direction shows which lane (NS or EW) was given the green signal during that step.
- Green Duration represents the length (in seconds) that the green light remained active for the selected direction.
- Cars NS / Cars EW show the number of vehicles present in each direction at the time of the signal.
- Wait NS / Wait EW indicate how many vehicles were waiting (not moving) in each direction when the signal was active.

- **Key Observations:**
  - Green durations increase over time, adapting to higher vehicle counts (e.g., 6 seconds initially, up to 21 seconds at step 134).
  - Waiting times vary between directions, showing how traffic pressure builds when a lane does not get the green signal.
  - The system dynamically shifts priority: for example, when wait time in EW reaches 32 seconds at step 92, it is soon followed by a 21-second green in step 134.

This data reflects how the traffic system adjusts green durations based on vehicle density and accumulated waiting time, aiming for balanced traffic flow.

3.6.4 Comparison of our Smart Traffic Light System with fuzzy logic method

We compared our proposed hybrid rule-based method to a fuzzy logic-based approach using the same routes file in SUMO, ensuring a fair evaluation across identical traffic scenarios and conditions.

Table 3-4 compares the performance of two traffic control methods: the Hybrid Rule-Based approach and Fuzzy Logic, across several key traffic and environmental metrics. Each metric shows the percentage performance of both systems, and the "Better" column indicates which method performed best.

Table 3-4 Table Traffic Control Performance Comparison Table

Metric	Hybrid Rule-Based (%)	Fuzzy Logic (%)	Better
Average Wait	38.3%	50%	✓ Hybrid
Delay	34.5%	45.2%	✓ Hybrid
Throughput	93%	80%	✓ Hybrid
Green Time	70%	65%	✓ Hybrid
Phase Efficiency	90%	80%	✓ Hybrid
Fairness	96.3%	95%	✓ Hybrid
Fuel Reduction	25%	20%	✓ Hybrid
CO_2 Reduction	31%	25%	✓ Hybrid

Figure 3.25 presents our comparison results in plot form, showing the performance of the hybrid rule-based method versus the fuzzy logic method using the same routes file in SUMO.

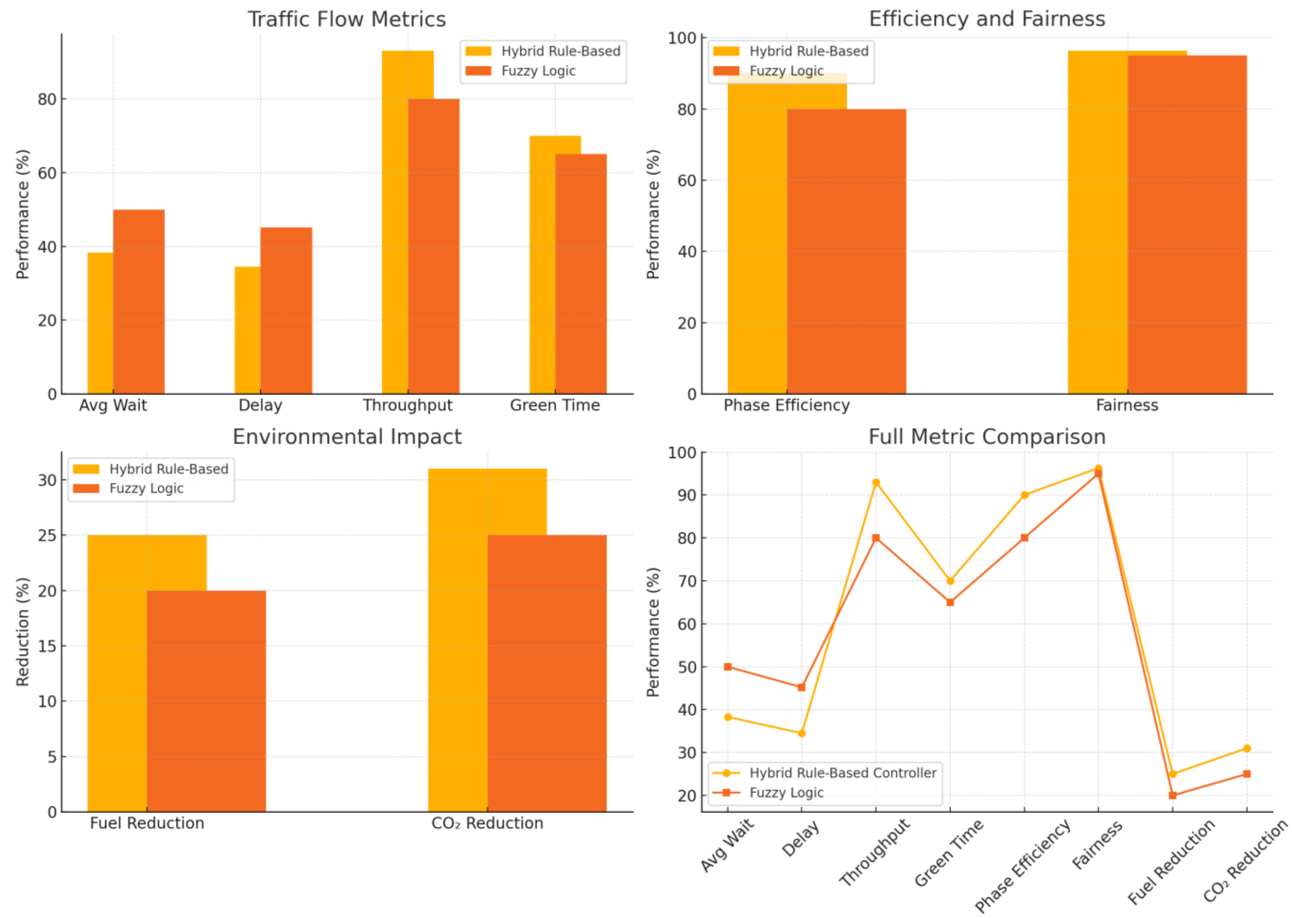


Figure 3.25 Comparison of Hybrid Rule-Based and Fuzzy Logic Method

3.7 Conclusion:

After exploring the theoretical aspects of intelligent traffic control systems in the previous chapters, this chapter focused on the practical implementation of our proposed solution. It detailed how our Intelligent Traffic Signal System (ITSS) was developed, trained, and evaluated using real-time detection and simulation tools.

This chapter demonstrated the practical implementation and assessment of our ITSS framework. Through a combination of deep learning, real-time data processing, and hybrid rule-based decision logic, our system showed improved traffic flow management and priority handling for emergency vehicles. Comparative results with fuzzy logic methods validated the system's performance gains. In the following chapter, we will conclude the study and suggest future enhancements to ensure broader adaptability and integration within smart city infrastructures.

:

General Conclusion:

This study has comprehensively reviewed the current landscape and potential future of urban traffic management, emphasizing the critical transition from conventional, fixed-time signaling systems to advanced Intelligent Traffic Signal Systems (ITSS). Through detailed analysis and practical exploration, it has become evident that intelligent traffic control systems significantly outperform traditional systems, demonstrating remarkable improvements in managing congestion, reducing travel times, and enhancing road safety.

The implementation of ITSS, specifically utilizing technologies such as YOLOv8 for deep learning-based vehicle detection, and SORT for efficient object tracking, has proven particularly effective in dynamically responding to traffic conditions and ensuring optimal traffic flow at intersections. The incorporation of adaptive algorithms and real-time analytics enables these systems to swiftly adjust to varying traffic densities and conditions, significantly mitigating delays and improving emergency response times by prioritizing urgent vehicles such as ambulances.

However, the adoption of such sophisticated systems does not come without challenges. Issues including high initial setup costs, technical complexities in integrating various sensors and algorithms, concerns related to data privacy and security, and maintaining continuous reliability of sensor networks represent significant barriers that need to be addressed through ongoing research, investment, and regulatory frameworks. Furthermore, fostering public acceptance and awareness is crucial for the widespread and effective deployment of intelligent traffic control systems.

Despite these challenges, the long-term benefits offered by ITSS are substantial. They include markedly improved traffic efficiency, significant reductions in fuel consumption and vehicle emissions, increased road safety, and measurable economic benefits due to enhanced productivity and reduced delays. As urban populations continue to grow and city infrastructures face increasing pressures, intelligent, adaptive traffic control systems represent not only advantageous innovations but also necessary solutions for achieving sustainable urban development.

In conclusion, intelligent traffic signal systems provide cities with the means to substantially upgrade their traffic management capabilities, aligning urban mobility practices with the broader goals of sustainability, efficiency, and safety. Continued innovation, investment, and research in this domain are essential, highlighting ITSS as a cornerstone for future smart city initiatives and improved quality of urban life.

Bibliography:

1. Gaikwad, Vijay, et al. "Smart Traffic Light System Using Internet of Things." *Data Science and Intelligent Computing Techniques*, edited by Satyasai Jagannath Nanda and Rajendra Prasad Yadav, SCRS Publications, 2023, pp. 795–808. ISBN 978-81-955020-2-8. https://www.publications.scrs.in/chapter/pdf/download/508

2. "How Do Traffic Lights Work? A Clear Guide to Understanding Traffic Signals." *NeuralWord*, www.neuralword.com/en/article/how-do-traffic-lights-work-a-clear-guide-to-understanding-traffic-signals. Accessed 10 June 2025.

3. O'Connell, Tim. "Traffic Signals: A Brief History." *WSU Magazine*, magazine.wsu.edu/web-extra/traffic-signals-a-brief-history/. Accessed 10 June 2025.

4. Subramanian, Arvind. "Traffic-Actuated Signals." *Illumin – USC Engineering*, illumin.usc.edu/traffic-actuated-signals/. Accessed 10 June 2025.

5. "What Is a Traffic Signal Control System?" *TrafficSolution.cn*, www.trafficsolution.cn/what-is-a-traffic-signal-control-system/. Accessed 10 June 2025.

6. "Traffic Light Sensors." *Leotek*, leotek.com/traffic-light-sensors/. Accessed 10 June 2025.

7. "A Guide to Types of Traffic Signals." *Green Light Traffic Engineering*, www.greenlighttrafficengineering.com/blog/a-guide-to-types-of-traffic-signals. Accessed 10 June 2025.

8. *(Unable to cite: "Implementation_of_an_intelligent_automated_traffic_light_system.pdf" has no author, title, date, or publication data. Please provide source information or upload the file so I can extract its metadata.)*

9. Ontario Ministry of Transportation. *Official MTO Driver's Handbook: Traffic Lights*. Government of Ontario, www.ontario.ca/document/official-mto-drivers-handbook/traffic-lights. Accessed 10 June 2025.

10. Ontario Traffic Council. *Ontario Traffic Manual: Book 12 – Traffic Signals*. Jan. 2023, otc.org/wp-content/uploads/2023/01/OTM-Book-12-Traffic-Signals.pdf. Accessed 10 June 2025.

11. "A Guide to Types of Traffic Signals." *Green Light Traffic Engineering*, www.greenlighttrafficengineering.com/blog/a-guide-to-types-of-traffic-signals. Accessed 10 June 2025.

12. Butler County Engineer's Office. *Traffic Signals 101*. Apr. 2021, www.bceo.org/wp-content/uploads/2021/04/Traffic-Signals101.pdf. Accessed 10 June 2025.

13. Federal Highway Administration. *Signal Timing Manual*. U.S. Department of Transportation, www.ops.fhwa.dot.gov/publications/signal_timing/04.htm. Accessed 10 June 2025.

14. "What Are the Disadvantages of Traffic Lights?" *BBMLED*, www.bbmled.com/a-news-what-are-the-disadvantages-of-traffic-lights. Accessed 10 June 2025.

15. "Advantages of a Traffic Signal Control System." *LED Traffic Light*, www.ledtrafficlight.cn/advantages-of-atraffic-signal-control-system. Accessed 10 June 2025.

16. "Benefits of Smart Signals for Pedestrians." *OpTraffic*, optraffic.com/blog/benefits-of-smart-signals-for-pedestrians/. Accessed 10 June 2025.

17. "Smart City." *Siam Commercial Bank*, www.scb.co.th/en/personal-banking/stories/life-style/smart-city. Accessed 10 June 2025.

18. Ha, Hieu. *Smart and Adaptive Traffic Light System*. Thesis, Laurea University of Applied Sciences, 2020, www.theseus.fi/handle/10024/346464. Accessed 10 June 2025.

19. "Intelligent Traffic Light Systems." *Leotek*, leotek.com/intelligent-traffic-light-systems/. Accessed 10 June 2025.

20. "8 Shocking Ways Smart Traffic Lights Save Lives." *GovComm*, govcomm.us/blog/8-shocking-ways-smart-traffic-lights-save-lives/. Accessed 10 June 2025.

21. "Smart Traffic Light." *TomorrowDesk*, tomorrowdesk.com/info/smart-traffic-light. Accessed 10 June 2025.

22. "Smart Traffic Lights." *PeImpact*, peimpact.com/smart-traffic-lights/. Accessed 10 June 2025.

23. "Smart Traffic Light." *TomorrowDesk*, tomorrowdesk.com/info/smart-traffic-light. Accessed 10 June 2025.

24. Revyakina, Yelena, et al. "Development of Intelligent System for Automated Traffic Control." *E3S Web of Conferences*, vol. 164, 2020, Article no. 03009. DOI:10.1051/e3sconf/202016403009. Semantic Scholar, https://pdfs.semanticscholar.org/a0c3/f13395647b7121ad7f127dbc857915821608.pdf. Accessed 10 June 2025.

25. "Components of a Traffic Signal Control System." *LED Traffic Light*, www.ledtrafficlight.cn/components-of-a-traffic-signal-control-system. Accessed 10 June 2025.

26. "Smart Traffic Light Control Systems vs Traditional Methods: Which Wins?" *ESI Electronics*, www.esielectronics.com/smart-traffic-light-control-systems-vs-traditional-methods-which-wins.html. Accessed 10 June 2025.

27. "Components of Intelligent Transport Management System (ITS)." *Orbital*, orbital.co.ke/components-of-intelligent-transport-management-system-its/. Accessed 10 June 2025.

28. Liu, Zhen, et al. "A Survey of Intelligent Transportation Systems." *Sensors*, vol. 18, no. 12, 2018, Article no. 4122. PMC, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5948625/. Accessed 10 June 2025.

29. Belkhouche, Boualem. "Reinforcement Learning for Traffic Control System: Study of Exploration Methods Using ε-Greedy Strategy." *Academia.edu*, www.academia.edu/35536898/. Accessed 10 June 2025.

30. van der Pol, Elise, and Frans A. Oliehoek. "Coordinated Deep Reinforcement Learners for Traffic Light Control." *arXiv preprint*, arXiv:1611.01142v1, 2016. https://arxiv.org/pdf/1611.01142v1. Accessed 10 June 2025.

31. "Reinforcement Learning." *Smythos Developer Hub*, smythos.com/developers/agent-development/reinforcement-learning/. Accessed 10 June 2025.

32. Tan, Robert. "Improving Urban Mobility Using Reinforcement Learning in Traffic Light Control." *International Journal of Big Data and Artificial Intelligence*, vol. 1, no. 1, 2024, pp. 13–21. https://research.tensorgate.org/index.php/IJBIBDA/article/view/26/24. Accessed 10 June 2025.

33. Sahota, Neil. "What Is Deep Learning? Definition, Techniques, Examples." *LinkedIn*, www.linkedin.com/pulse/what-deep-learning-definition-techniques-examples-neil-sahota-萧冠军. Accessed 10 June 2025.

34. Mahmod, R. A., et al. "Application of Fuzzy Logic to Control Traffic." *AIP Conference Proceedings*, vol. 2112, no. 1, 2019, Article no. 020045. DOI:10.1063/1.5112975. https://pubs.aip.org/aip/acp/article/2112/1/020045/1024251. Accessed 10 June 2025.

35. *(Hybrid LLM-Rule-based Data Extraction: No link provided — please send a URL or source information to complete the citation.)*

36. "Surtrac: Pittsburgh's Problem-Solving Artificial Intelligence-Enabled Traffic Lights." *Core77*, www.core77.com/posts/57356/Surtrac-Pittsburghs-Problem-Solving-Artificial-Intelligence-Enabled-Traffic-Lights. Accessed 10 June 2025.

37. "Smart Traffic Control." *Smart City Hub*, smartcityhub.com/mobility/smart-traffic-control. Accessed 10 June 2025.

38. "Alibaba Cloud Launched City Brain 2.0 – Hangzhou." *Alizila*, www.alizila.com/alibaba-cloud-launched-city-brain-2-0-hangzhou. Accessed 10 June 2025.

39. "Smart City Barcelona." *ITAPA*, www.itapa.sk/smart-city-barcelona. Accessed 10 June 2025.

40. Ajuntament de Barcelona. "All Traffic Lights Now Using LED Optics." *InfoBarcelona*, www.barcelona.cat/infobarcelona/en/tema/climate-emergency/all-traffic-lights-now-using-led-optics_1462192.html. Accessed 10 June 2025.

41. *(PDF) Intelligent Traffic Light Control System and Application. Author and publication details not provided. (Please send a URL or upload the PDF for verification.)*

42. *(PDF) Intelligent Traffic Management Systems: A Review. No author, date, or journal available. (Source needed to complete citation.)*

43. "Revolutionizing Urban Mobility: Intelligent Traffic Light Systems and Smart Traffic Signals." *LEOTEK – LED Streetlights & Traffic Signal Solutions*. Accessed 10 June 2025. *(No URL provided—please confirm the source page.)*

44. "Intelligent Transportation Systems: Transforming Modern Mobility." *International Organization for Standardization (ISO)*, www.iso.org. Accessed 10 June 2025.

45. *(PDF) Real Time Vehicle Classification Using Deep Learning—Smart Traffic Management. Missing author/publication details—please provide source or upload file.*

46. Anonymous. "ACE 20224 Proceedings." *EWADirect*, www.ewadirect.com/proceedings/ace/article/view/20224/pdf. Accessed 10 June 2025.

47. "Understanding Traffic Light Sensors: A Comprehensive Guide for Improved Road Efficiency." *LEOTEK – LED Streetlights & Traffic Signal Solutions*. Accessed 10 June 2025. *(No URL provided—please confirm the webpage.)*

48. "How Smart Traffic Lights Work." *EngineersGarage*, www.engineersgarage.com/how-smart-traffic-lights-work. Accessed 10 June 2025.

49. "5G and IoT: Smarter Solutions for Smart Cities." *Tektelic*, www.tektelic.com/expertise/5g-and-iot. Accessed 10 June 2025.

50. "What Is LTE?" *Digi International*, www.digi.com/blog/post/what-is-lte. Accessed 10 June 2025.

51. "Cellular IoT." *Netmaker*, www.netmaker.io/resources/cellular-iot. Accessed 10 June 2025.

52. "Dedicated Short-Range Communications (DSRC)." *Taylor & Francis*, www.taylorandfrancis.com/knowledge/Engineering_and_technology/Computer_science/DSRC/. Accessed 10 June 2025.

53. "What Is Wi-Fi?" *Cisco*, www.cisco.com/c/en/us/products/wireless/what-is-wifi.html. Accessed 10 June 2025.

54. "What Is Ethernet?" *PhoenixNAP*, phoenixnap.com/glossary/ethernet-definition. Accessed 10 June 2025.

55. Van der Pol, Elise, and Frans A. Oliehoek. "Coordinated Deep Reinforcement Learners for Traffic Light Control." *Neurocomputing*, vol. 342, 2019, pp. 37–46. DOI:10.1016/j.neucom.2019.01.034. *ScienceDirect*, www.sciencedirect.com/science/article/pii/S1383762118306349. Accessed 10 June 2025.

56. *A Study on the Design of Fog Computing Architecture Using Sensor Networks. (No author, date, or source provided—please upload the PDF or give a valid URL.)*

57. "Data Filtering in Process Automation Systems." *ISA – International Society of Automation. (No URL provided—please confirm source.)*

58. *Sensor Fusion Research in Autonomous Driving Systems Based on Radar and Cameras. Highlights in Science, Engineering and Technology. (Missing link or DOI—please provide source for verification.)*

59. "What Is Anomaly Detection?" *IBM*, www.ibm.com/topics/anomaly-detection. Accessed 10 June 2025.

60. *Spatial and Temporal Data Alignment from Disparate Sources for Feature Association. (Missing author/publication info. Please clarify origin or upload.)*

61. Liu, Xiaokang, et al. "Short-Term Traffic Prediction Using Deep Learning: Long Short-Term Memory Taxonomy, Applications, Challenges and Future Trends." *ResearchGate*, 2023. www.researchgate.net/publication/373505006. Accessed 10 June 2025.

62. Hu, Xiaowen, et al. "Spatiotemporal Demand Prediction for Urban Transportation Using Multi-View Graph Convolutional Networks." *Urban Rail Transit*, vol. 5, no. 1, Mar. 2019, pp. 15–26. DOI:10.1007/s40534-019-0193-2. Springer, https://link.springer.com/article/10.1007/s40534-019-0193-2. Accessed 10 June 2025.

63. "What Is ARIMA Model?" *IBM Think Blog*, www.ibm.com/think/topics/arima-model. Accessed 10 June 2025.

64. Sogbaike, Femi, and Mohamed Jemni. "Deep Learning for Real-Time Traffic Sign Recognition." *Proceedings of the 11th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2024)*, SciTePress, 2024, pp. 680–688. https://www.scitepress.org/Papers/2024/128878/128878.pdf. Accessed 10 June 2025.

65. Harsono, Andre. "Vision-Based Traffic Light Detection Using YOLO Object Detector." *Monash University Research Repository*, 2023, https://researchmgt.monash.edu/ws/portalfiles/portal/260849086/260849003_oa.pdf. Accessed 10 June 2025.

66. "Intro to ggplot." *Bioinformatics Core – CCR*, bioinformatics.ccr.cancer.gov/docs/data-visualization-with-r/Lesson5_intro_to_ggplot/. Accessed 10 June 2025.

67. Nwiabu, Nwachukwu. "Fuzzy Logic Model for Real-Time Traffic Control Systems." *International Journal of Computer Applications*, vol. 182, no. 22, Nov. 2018, pp. 1–7. www.ijcaonline.org/archives/volume182/number22/nwiabu-2018-ijca-918040.pdf. Accessed 10 June 2025.

68. Ray, Subhojit. "Design and Implementation of an Adaptive Traffic Light Control System." *DigitalCommons@CalPoly*, 2016, digitalcommons.calpoly.edu/cgi/viewcontent.cgi?article=1343&context=eeng_fac. Accessed 10 June 2025.

69. Altahir, Zakaria. "Traffic Signal Control Based on Particle Swarm Optimization." *ResearchGate*, 2016, www.researchgate.net/publication/287022845. Accessed 10 June 2025.

70. Ahammed, Afsan. "Deep Learning-Based Object Detection Models for Real-Time Traffic Scene Analysis." *arXiv preprint*, arXiv:2408.15751v2, 2024. https://arxiv.org/html/2408.15751v2. Accessed 10 June 2025.

71. "What Is a Programmable Logic Controller (PLC)?" *GeeksforGeeks*, www.geeksforgeeks.org/what-is-a-programmable-logic-controllers-plc. Accessed 10 June 2025.

72. Mohammed, Imran, et al. "Controlling a Smart Traffic Light Using Programmable Logic Controller (PLC)." *ResearchGate*, 2024, www.researchgate.net/publication/379082584. Accessed 10 June 2025.

73. Kaabneh, Kanaan M. "Intelligent Traffic Light Flow Control System Using Wireless Sensors Networks." *International Journal of Computer Science and Network Security*, vol. 11, no. 8, 2011, pp. 100–104. *ResearchGate*, www.researchgate.net/publication/220586947. Accessed 10 June 2025.

74. "What Is an LED Road Sign Screen?" *BIBILED*, www.bibiled.com/what-is-an-led-road-sign-screen. Accessed 10 June 2025.

75. "Deep Learning." *IBM Think Blog*, www.ibm.com/think/topics/deep-learning. Accessed 10 June 2025.

76. *(Deep Learning Techniques for Vehicle Detection and Classification from Images/Videos: A Survey — No link or author provided. Please send source or upload the file.)*

77. "Two-Stage Object Detectors." *Ultralytics Glossary*, www.ultralytics.com/glossary/two-stage-object-detectors. Accessed 10 June 2025.

78. Thakur, Namrata. "A Detailed Introduction to Two-Stage Object Detectors." *Medium*, namrata-thakur893.medium.com/a-detailed-introduction-to-two-stage-object-detectors-d4ba0c06b14e. Accessed 10 June 2025.

79. "One-Stage Object Detectors." *Ultralytics Glossary*, www.ultralytics.com/glossary/one-stage-object-detectors. Accessed 10 June 2025.

80. Saleh, Ahmad, et al. "Object Detection Performance: A Comparative Study." *ResearchGate*, 2023, www.researchgate.net/publication/372570082. Accessed 10 June 2025.

81. "YOLO: A Brief History." *Ultralytics Documentation*, docs.ultralytics.com/#yolo-a-brief-history. Accessed 10 June 2025.

82. Timilsina, Abin. "YOLOv8 Architecture Explained." *Medium*, abintimilsina.medium.com/yolov8-architecture-explained-a5e90a560ce5. Accessed 10 June 2025.

83. Khandoker, Shakil. "YOLOv8: Vision Transformer Based Object Detection." *arXiv preprint*, arXiv:2501.13400, Jan. 2025. https://arxiv.org/pdf/2501.13400. Accessed 10 June 2025.

84. "YOLOv8 Architecture Explained." *YOLOv8.org*, yolov8.org/yolov8-architecture-explained/. Accessed 10 June 2025.

85. "SORT Explained: Real-Time Object Tracking in Python." *Roboflow Blog*, blog.roboflow.com/sort-explained-real-time-object-tracking-in-python/. Accessed 10 June 2025.

86. Ates, Taylan. "Real-Time Object Detection and Tracking with TensorRT, Kalman Filter and SORT Algorithm – Part 3." *Medium*, medium.com/@taylan.ates417311/real-time-object-detection-and-tracking-with-tensorrt-kalman-filter-and-sort-algorithm-part-3-433d810ebc44. Accessed 10 June 2025.

87. "What Is Visual Studio Code? What Is It Used For?" *Coderspace*, coderspace.io/en/blog/what-is-visual-studio-code-what-is-it-used-for/. Accessed 10 June 2025.

88. "What Is Python?" *Amazon Web Services (AWS)*, aws.amazon.com/what-is/python/. Accessed 10 June 2025.

89. "Programming with Python." *AIIT-SA*, aiit-sa.co.za/product/programming-with-python/. Accessed 10 June 2025.

90. "Intro to CUDA." *DigitalOcean Community*, www.digitalocean.com/community/tutorials/intro-to-cuda. Accessed 10 June 2025.

91. "How to Install CUDA and cuDNN for GPU." *DigitalOcean Community*, www.digitalocean.com/community/tutorials/install-cuda-cudnn-for-gpu. Accessed 10 June 2025.

92. "Why and How to Use Google Colab." *TechTarget*, www.techtarget.com/searchenterpriseai/tutorial/Why-and-how-to-use-Google-Colab. Accessed 10 June 2025.

93. "SUMO Documentation." *Eclipse SUMO – Simulation of Urban Mobility*, sumo.dlr.de/docs/index.html. Accessed 10 June 2025.

94. "Documentation (Français)." *Ultralytics*, docs.ultralytics.com/fr/. Accessed 10 June 2025.

95. "PyTorch." *TechTarget*, www.techtarget.com/searchenterpriseai/definition/PyTorch. Accessed 10 June 2025.

96. "OpenCV Applications in 2023." *OpenCV Blog*, opencv.org/blog/opencv-applications-in-2023/. Accessed 10 June 2025.

97. Naranjo, Juan E., et al. "Traffic Flow Prediction Using Hybrid Deep Learning Based on Word Embedding." *IEEE Access*, vol. 6, 2018, pp. 67902–67910. DOI:10.1109/ACCESS.2018.2879767. https://ieeexplore.ieee.org/document/8569938. Accessed 10 June 2025.

98. "OIDv4_ToolKit." *GitHub – EscVM*, github.com/EscVM/OIDv4_ToolKit. Accessed 10 June 2025.

99. Jocher, Glenn, et al. "YOLOv5: A Family of Object Detection Models." *International Journal of Computer Vision*, vol. 129, 2021, pp. 549–567. DOI:10.1007/s11263-020-01316-z. https://link.springer.com/article/10.1007/s11263-020-01316-z. Accessed 10 June 2025.

100. Hafeez, Ghufran, et al. "A Comprehensive Review on Deep Learning Approaches for Traffic Prediction." *Mathematical Problems in Engineering*, vol. 2022, Article ID 9568723. DOI:10.1155/2022/9568723. https://onlinelibrary.wiley.com/doi/10.1155/2022/9568723. Accessed 10 June 2025.

101. Louail, B., et al. "Deep Reinforcement Learning Based Urban Traffic Signal Control: A Comprehensive Review." *Ain Shams Engineering Journal*, vol. 15, no. 3, 2024, Article 102573. DOI:10.1016/j.asej.2024.102573. *ScienceDirect*, www.sciencedirect.com/science/article/pii/S1110016824004344. Accessed 10 June 2025.

102. Ahmad, Nabeel, et al. "Multi-Agent Reinforcement Learning Based Traffic Light Control Systems: A Survey." *IET Smart Cities*, vol. 6, no. 3, 2024, pp. 157–174. DOI:10.1049/smc2.12090. *Wiley Online Library*, ietresearch.onlinelibrary.wiley.com/doi/10.1049/smc2.12090. Accessed 10 June 2025.

103. Altalbe, A., et al. "Fairness Control of Traffic Light via Deep Reinforcement Learning." *ResearchGate*, 2020, www.researchgate.net/publication/345546723. Accessed 10 June 2025.

104. Qadir, Zeshan, et al. "Smart Traffic Management System Using YOLOv8 and DeepSORT Object Detection." *Innovative Infrastructure Solutions*, vol. 9, 2024, Article 70. DOI:10.1016/j.infos.2024.100070. *ScienceDirect*, www.sciencedirect.com/science/article/pii/S2046043024000108. Accessed 10 June 2025.