Machine Vision based Dynamic Traffic Signal Duration Management and Synchronization

A PROJECT REPORT

Submitted by

KUMAR LAXMIKANT

(Reg. No. CH.EN.U4CSE20039)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING

Under the guidance of

Prof. Dr. M. Ravichandran

Submitted to

AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF COMPUTING

CHENNAI – 601103

November 2023

BONAFIDE CERTIFICATE

This is Certification that this project report entitled **"Machine Vision based Dynamic Traffic Signal Duration Management and Synchronization"** is the bonafide work of **"Mr. Kumar Laxmikant (Reg. No. CH.EN.U4CSE20039)",** who carried out the project work under my supervision.

SIGNATURE

Dr. S SOUNTHARRAJAN
PROGRAM HEAD,
Department of CSE.
Amrita School of Computing
Chennai.

SIGNATURE

Dr. M. Ravichandran
SUPERVISOR
Assistant Professor, Dept. of CSE.
Amrita School of Computing
Chennai.

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION BY THE CANDIDATE

I declare that the report entitled **“Machine Vision based Dynamic Traffic Signal Duration Management and Synchronization”** submitted by me for the degree of Bachelor of Technology is the record of the project work carried out by me under the guidance of **“Dr. M. Ravichandran”** and this work has not formed the basis for the award of any degree, diploma, associateship, fellowship, titled in this or any other University or other similar institution of higher learning.

SIGNATURE

Kumar Laxmikant
(Reg. No. CH.EN.U4CSE20039)

ABSTRACT

Unceasing urbanization and the exponential proliferation of vehicles on our roadways have necessitated innovative traffic management solutions to mitigate the escalating challenges, from chronic congestion to road safety hazards. This research paper underscores the pivotal role of Intelligent Traffic Systems (ITS) in addressing these multifaceted issues and expounds on the formidable capabilities of the YOLO (You Only Look Once) framework, particularly YOLOv8, in revolutionizing urban traffic management. By harnessing the cutting-edge capabilities of YOLOv8, this study embarks on a transformative journey to enhance traffic control at intersections. The core innovation lies in utilizing YOLOv8 to perform real-time, lane-specific vehicle density detection. With its real-time object detection prowess, YOLOv8 is adept at discerning the density of vehicles at each lane within a traffic intersection, enabling precise and dynamic traffic signal adjustments. This data-driven approach empowers the traffic management system to make intelligent decisions about signal durations based on the observed vehicle density, optimizing traffic flow and minimizing congestion. Through the fusion of computer vision and traffic engineering, YOLOv8 promises to reshape urban traffic management paradigms, making cities safer, more efficient, and conducive to sustainable urban living. In this paper, we delve into the technical intricacies of implementing YOLOv8 for vehicle density estimation and its seamless integration into the fabric of traffic signal management. The results unveil a promising pathway toward the creation of adaptive and responsive traffic control systems, offering a glimpse into the future of urban mobility, where traffic signals adapt harmoniously to the ebb and flow of city life.

Keywords: Intelligent Traffic Systems, YOLOv8, Real-time Vehicle Detection, Computer Vision

ACKNOWLEDGEMENT

This project work would not have been possible without the contribution of many people. It gives me immense pleasure to express my profound gratitude to our honorable Chancellor **Sri Mata Amritanandamayi Devi,** for her blessings and for being a source of inspiration. I am indebted to extend my gratitude to our Director, **Mr. I B Manikantan** Amrita School of Computing and Engineering, for facilitating us with all the facilities and extended support to gain valuable education and learning experience.

I register my special thanks to **Dr. V. Jayakumar,** Principal Amrita School of Computing and Engineering for the support given to me in the successful conduct of this project. I wish to express my sincere gratitude to my supervisor **Dr. M. Ravichandran,** Assistant Professor, Department of Computer Science and Engineering, for his inspiring guidance, personal involvement and constant encouragement during the entire course of this work.

I am grateful to the Project Coordinator, Review Panel Members and the entire faculty of the Department of Computer Science & Engineering, for their constructive criticisms and valuable suggestions which have been a rich source to improve the quality of this work.

Kumar Laxmikant
(Reg. No. CH.EN.U4CSE20039)

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

Abbreviation	Full Form
ML	Machine Learning
CV	Computer Vision
YOLO	You Only Look Once
ITS	Intelligent Traffic System
FPS	Frames Per Second
mAP	Mean Average Precision
CNN	Convolutional Neural Network
IoU	Intersection Over Union
AQL	Average Queue Length
AWT	Average Waiting Time
AS	Average Speed
EDA	Exploratory Data Analysis
MSCOCO	Microsoft Common Objects in Context
RCNN	Regions with Convolutional Neural Networks
CLAHE	Contrast Limited Adaptive Histogram Equalization
SGD	Stochastic Gradient Descent
GFLOPs	Giga Floating-Point Operations Per Second
SPP	Spatial Pyramid Pooling

CHAPTER 1
INTRODUCTION

1.1 Project Background

Rapid urbanization and the exponential growth of vehicular traffic present a pressing challenge for modern cities. Chronic traffic congestion, rising accident rates, and inefficient traffic management have become ubiquitous issues affecting both urban residents and the economy. The consequences are apparent—persistent traffic congestion, escalating accident rates, and the pressing need for more efficient traffic management solutions. In response to these challenges, this research paper centers its focus on the convergence of Intelligent Traffic Systems (ITS) and cutting-edge computer vision technology.

ITS plays a pivotal role in addressing the multifaceted issues associated with urban traffic management. It is within this context that we explore the transformative potential of leveraging advanced computer vision techniques to enhance traffic control at intersections. Specifically, our research investigates the critical task of real-time vehicle density detection within traffic intersections, a fundamental aspect of traffic signal optimization. This data-driven approach equips traffic management systems with the capability to make intelligent decisions about signal durations, ultimately leading to improved vehicular movement and blockage reduction.

The fusion of computer vision and traffic engineering holds the promise of reshaping urban traffic management paradigms. By leveraging data-driven insights and real-time analytics, we aim to contribute to making cities safer, more efficient, and more conducive to sustainable urban living. In the subsequent sections, we will delve into the technical aspects of implementing our approach and discuss its potential to create adaptive and responsive traffic control systems that offer a glimpse into the future of urban mobility, where traffic signals seamlessly adapt to the dynamic pulse of city life.

1.2 Objectives

The following are the objectives of Machine Vision based Dynamic Traffic Signal Duration Management and Synchronization:

- **Real-time Traffic Monitoring:** Develop a system capable of continuously monitoring traffic conditions at urban intersections, collecting data on vehicle density, movement patterns, and congestion levels.

- **Data-Driven Analysis:** Implement advanced computer vision techniques to analyze the collected data and provide real-time insights into traffic dynamics, identifying congestion hotspots and traffic bottlenecks.

- **Dynamic Traffic Signal Control:** Create an intelligent traffic signal control algorithm that utilizes the real-time data to dynamically adjust signal durations at each intersection, prioritizing efficient traffic flow and congestion mitigation.

- **Reduction in Congestion:** Measure the system's effectiveness in reducing traffic congestion, minimizing delays, and improving overall traffic flow efficiency.

- **Fuel Efficiency Enhancement:** Evaluate the effect of the dynamic traffic signal optimization on fuel efficiency & greenhouse gas emissions, aiming to reduce fuel wastage and environmental impact.

- **Emergency Vehicle Prioritization:** Implement a feature that detects and prioritizes emergency vehicles approaching intersections, granting them right-of-way and optimizing their routes for rapid response times.

1.3 Scope

This research project encompasses the development and implementation of an advanced urban traffic management system that integrates Intelligent Traffic Systems (ITS) and cutting-edge computer vision technology. Focusing on intersections, where traffic congestion is a persistent issue, the research begins with the collection of real-time traffic data through CCTV cameras. Using computer vision and machine learning techniques, the collected data will be analyzed to gain insights into traffic dynamics, congestion patterns, and vehicle density. The core of the project involves the creation of an intelligent traffic signal optimization algorithm that can adaptive adjust signal durations based on live traffic data analysis. Additionally, the system will prioritize emergency vehicles for swift response. Rigorous testing will be conducted using real-time video feeds from CCTV cameras and in collaboration with traffic simulation models, ensuring real-world applicability and performance. The research aims to reduce congestion, enhance fuel efficiency, improve safety, and ultimately create a more efficient and adaptive urban traffic management system.

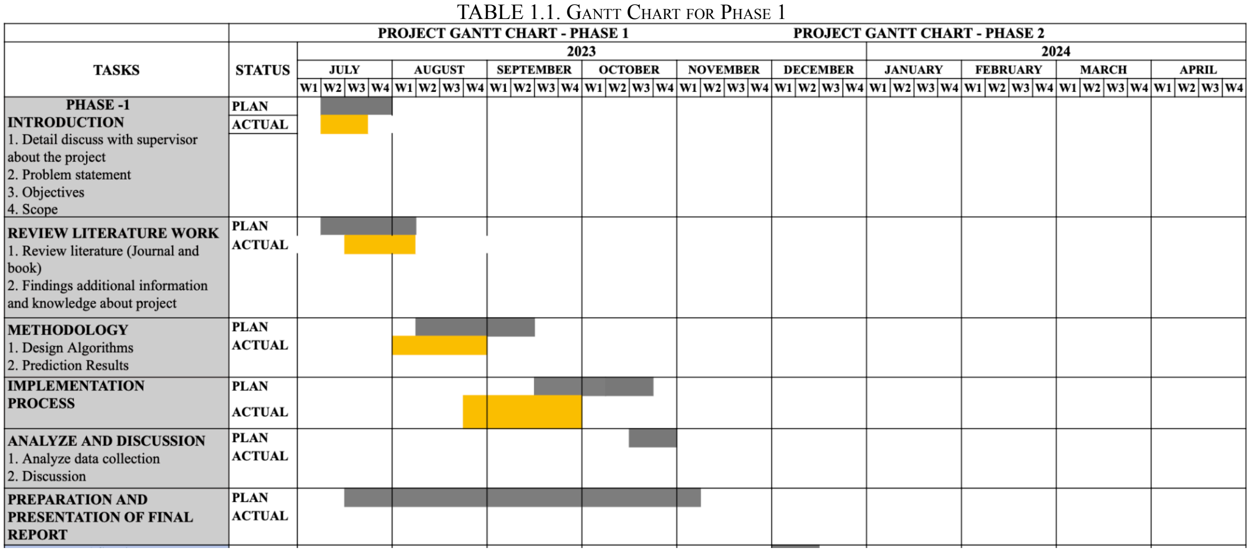
1.4 Expected Result

The AI model's vehicle object detection and real-time vehicle density analysis within specified regions are expected to guide dynamic traffic signal control, determining lane-specific green signal assignments. Anticipated outcomes include reduced intersection congestion, improved fuel efficiency, shorter travel times, and an overall streamlined urban transportation experience. Enhanced road safety, accelerated emergency vehicle responses, and a decrease in emissions are also expected benefits. Rigorous real-world testing and clear implementation guidelines aim to ensure the practicality and seamless integration of this AI-driven system into urban traffic management, ushering in a more efficient and adaptive traffic landscape.

1.5 Project Schedule / Gantt Chart

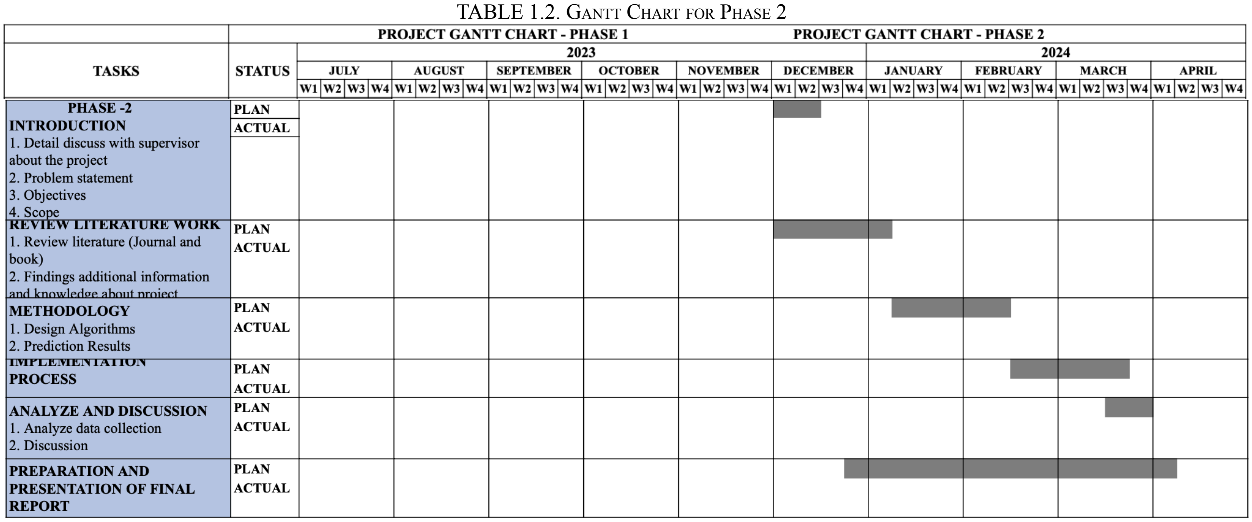
1.5.1 Phase 1

TABLE 1.1. GANTT CHART FOR PHASE 1



1.5.2 Phase 2

TABLE 1.2. GANTT CHART FOR PHASE 2



CHAPTER 2
LITERATURE REVIEW

The literature survey serves as the foundation for this research, providing a comprehensive overview of the existing body of knowledge related to urban traffic management, computer vision applications, and Intelligent Traffic Systems (ITS). This section aims to distill insights from prior research, identify gaps in current understanding, and highlight key findings that inform the research's conceptual framework and methodology.

2.1 Literature review based on previous research papers

Padilla et al. [1] make a significant contribution to vehicle detection with their work on "T-YOLO." This novel approach addresses the challenges of lightweight and efficient vehicle detection. T-YOLO, based on YOLO (You Only Look Once) and multi-scale convolutional neural networks (CNNs), offers a compact yet robust solution optimized for resource-efficient processing. This balance between accuracy and speed makes T-YOLO suitable for intelligent traffic systems and mobile devices. Padilla et al. report impressive performance metrics, including a Precision of 96.34%, Recall of 99.98%, mAP50 of 99.85%, and mAP95 of 99.79%, showcasing T-YOLO's practical utility in real-world vehicle detection scenarios. Their experiments using the PKLot 1 dataset, focusing on parking locations, meteorological states, and specific days, highlight their data-driven approach to model training and validation.

Zarei et al. [2] introduce "Fast-Yolo-Rec," a noteworthy advancement in the realm of rapid vehicle detection across consecutive images. Their method uniquely combines the strengths of YOLO-based detection with a recurrent-based prediction network, addressing the imperative for swift vehicle detection in dynamic scenarios. Importantly, they employ the SSAM-YOLO framework with the SemAtt_Net backbone, achieving commendable results. Their model boasts an Average Precision of 94.06% and an impressively low Miss Rate of 0.0695, highlighting its exceptional accuracy. Strikingly, the accuracy achieved by Fast-Yolo-Rec is nearly on par with YOLOv4_Tiny, a widely recognized model, while offering a substantial advantage in terms of efficiency, with approximately 29.38% fewer parameters than YOLOv4_Tiny. This pioneering approach underscores the significance of both speed and accuracy in real-time vehicle detection scenarios, demonstrating the potential of Fast-Yolo-Rec as a compelling solution for applications requiring rapid and precise vehicle detection across consecutive image frames.

Zuraimi et al. [3] compares various YOLO models for vehicle detection and tracking, revealing YOLOv4 as the top performer with an impressive 82.08% mAP50 on their custom dataset. This model's success is attributed to its advanced architecture, showcasing improvements over prior iterations. YOLOv4 not only excels in accuracy but also attains an instantaneous algorithmic throughput of about 14 FPS on a GTX 1660ti graphics card, striking a balance between precision and speed. This research underscores the practical significance of YOLO models in real-time applications, particularly in domains like vehicle monitoring.

The study by ***Gorodokin et al.*** [4] innovatively employs machine vision to enhance adaptive traffic light control, addressing traffic flow efficiency and congestion. By integrating YOLO v4 neural network and SORT open-source tracker, they achieve a remarkable 92% accuracy in vehicle detection and classification. This work demonstrates the practicality of combining advanced deep learning and tracking techniques for effective traffic analysis. The study underscores the potential of data-driven strategies to revolutionize urban mobility and adaptive traffic control systems.

Miao et al. [5] contribute significantly to low-light vehicle detection with their research. Their innovative approach employs the MSR algorithm for image enhancement, coupled with tailored YOLO v3 neural network training. This adaptation achieves an impressive 93.66% accuracy, operating at a rapid 30.03 fps frame rate. Their methodology effectively addresses the challenge of nighttime detection, demonstrating the versatility of YOLO v3 architecture and its potential for real-world applications.

Chen et al. 's [6] research introduces an innovative neural network structure, YOLO v3-live, as an evolution of YOLOv3-tiny for improved object detection. YOLO v3-live achieves 87.79% mAP precision before quantization and maintains 69.79% mAP post-quantization, with a detection speed of 28 FPS. This advancement signals progress in striking a balance between accuracy and real-time processing, offering promising applications in various domains.

Dong et al. [7] introduced a significant contribution to computer vision and traffic management with their lightweight vehicle detection network model based on YOLOv5. Their enhancements to the YOLOv5 architecture, including the incorporation of CBAM modules and C3Ghost and Ghost modules, result in an efficient and highly accurate vehicle detection system. Notably, their model achieved an impressive mean Average Precision at IoU (Intersection over Union) threshold of 0.5 (mAP50) of 72.4% and mAP at IoU threshold of 0.95 (mAP95) of 50.5%, showcasing its robust performance. Rigorously tested on benchmark datasets, PASCAL VOC and MS COCO, the model's versatility and adaptability are evident. In our context, this research provides a valuable reference, highlighting the potential for YOLO-based models to excel in real-time vehicle detection, an essential component of our proposed dynamic traffic signal management system. The architectural enhancements, coupled with impressive performance metrics, make this work a significant foundation for our research objectives.

In the study by ***H. Song et al.*** [8], a vision-based vehicle detection and counting system for highway scenes was presented, utilizing deep learning techniques coupled with Gaussian mixture modeling and preprocessing methods. The authors employed Gaussian mixture modeling for surface extraction and segmentation, enhancing vehicle detection accuracy in conjunction with deep learning object detection. Subsequent preprocessing steps, including Gaussian filtering, MeanShift color correction, and the flooding filling algorithm for road surface separation, contributed to improved object detection by refining the background and isolating regions of interest. Additionally, the use of the ORB algorithm for feature extraction from detected vehicles yielded a remarkable mean Average Precision (mAP) value of 87.88%, demonstrating the efficacy of their approach in accurately detecting vehicles in complex highway environments. These findings underscore the potential of deep learning and preprocessing techniques for enhancing vehicle detection and counting, aligning with the objectives of our research in urban traffic management.

In their recent work, ***Khalifa et al.*** [9] made a significant contribution to Intelligent Transportation Systems (ITS) by employing Convolutional Neural Networks (CNNs) for vehicle detection within vision-based systems. Notably, their research utilized the YOLOv5s architecture, achieving an impressive mean Average Precision (mAP) of 97.8% on daytime data and 95.1% on nighttime data. This showcases the robustness of their CNN-based approach under varying lighting conditions and scenarios. Their work underscores the potential of computer vision techniques in urban traffic management, providing valuable insights into the integration of vehicle detection systems with dynamic traffic signal control strategies, aligning closely with our research objectives.

Chen et al. [10] introduce an innovative approach that leverages edge computing to enhance real-time vehicle detection. Their system achieves an impressive mean Average Precision (mAP) of 86.22% and Intersection over Union (IoU) of 75.63% using YOLOv4. This high accuracy is vital for safe autonomous navigation, particularly in complex urban traffic conditions. Additionally, the authors address real-time image segmentation, essential for comprehending the driving environment. Their work emphasizes the role of edge computing in improving processing capabilities for autonomous vehicles, promising enhanced safety and efficiency in urban transportation.

Wu et al. [11] introduced the Multi-Agent Recurrent Deterministic Deep Policy Gradient (MARDDPG) algorithm for decentralized traffic signal control at multiple intersections. Drawing inspiration from the DRQN and RDPG algorithms, MARDDPG incorporates recurrent neural networks (RNNs) into the Q-network. The algorithm has been tested on SUMO Simulator with an average waiting time of 2.59 ± 0.89 and average queue length of 3.04 ± 5.19. This enhancement allows traffic lights, acting as independent agents, to make more informed decisions by considering both current and past observations. Experimental results demonstrated congestion reduction and improved traffic flow, showcasing the algorithm's potential to optimize urban mobility. Wu et al.'s research represents a significant advancement in decentralized traffic management, offering insights into the value of recurrency in enhancing traffic signal control.

Ge et al. [12] introduces Cooperative Deep Q-Learning with Q-Value Transfer (QT-CDQN) for networked signal coordination. QT-CDQN enables intersections to collaborate by sharing optimal Q-values, leading to synchronized traffic signal decisions. It uses Deep Q-Learning (DQL) with neural networks to adapt signal timings based on real-time traffic conditions. Our research complements this by integrating computer vision techniques, enhancing signal control's adaptability and responsiveness for more efficient transportation systems.

Wang et al. [13] introduced a novel approach to macroscopic signal management using multiagent reinforcement learning (MARL). They proposed cooperative double Q-learning (Co-DQL), a distinctive algorithm within decentralized MARL. Results demonstrated Co-DQL's superiority over existing MARL approaches, achieving an average delay timing of 36.981 ± 0.509 and a mean episode reward of -2.889 ± 0.040. These findings highlight the potential of Co-DQL in effectively reducing traffic congestion and optimizing urban traffic management, making it a valuable contribution to the domain of signalized intersection management.

Liang et al. [14] introduced the Double Dueling Deep Q Network (3DQN) as a key component in their urban traffic management research. Their study showcased a remarkable 25.7% reduction in average waiting times, a significant improvement compared to fixed-time strategies. The 3DQN model consistently achieved cumulative rewards greater than −50000, highlighting its effectiveness in optimizing traffic signal control. These findings underscore the potential of adaptive traffic management systems, a concept that aligns with our research's goal to enhance urban traffic efficiency using computer vision techniques for vehicle density analysis.

Tajalli et al. [15] explore urban traffic management in the context of connected and automated vehicles (CAVs). They investigate the integration of holistic speed optimization with signal control at a network level, emphasizing CAVs' transformative potential for urban mobility. By dynamically adjusting CAV speeds based on real-time traffic conditions and signal timing, the study aims to reduce congestion and enhance fuel efficiency. The synchronized strategy resulted in a 1.9% decrease in travel duration, a 5.3% reduction in mean latency, a 28.5% decline in the average stop count, and a 5.4% decrease in the average stop delay, when compared to scenarios exclusively optimizing signal timing parameters. The research also implements a distributed optimization algorithm in Vissim, yielding notable network performance improvements compared to single-parameter optimizations, showcasing the practicality of network-level coordination for CAVs.

Cai et al. [16] employs Approximate Dynamic Programming (ADP) to model and optimize traffic signal control, aiming to minimize delays and enhance traffic flow efficiency. This study's noteworthy contributions encompass the application of ADP for adaptive signal timing adjustments, resulting in a remarkable 67% reduction in vehicle delays compared to fixed-time plans at a 0.5-second resolution. Moreover, operating at a 0.5-second resolution, instead of 5.0 seconds, the same ADP controller exhibits a notable 41% improvement in performance. This study introduces a foundational technique to the adaptive signalized intersection control system, offering significant potential for alleviating traffic congestion in urban areas.

Priemer et al. [17] propose a decentralized approach to adaptive signalized intersection control system leveraging Vehicle-to-Infrastructure (V2I) communication data. Their method demonstrates significant improvements, including a 5% increase in mean traffic speed and a nearly 25% reduction in mean delay within the network at high penetration rates of 100%. This highlights the potential of V2I communication and decentralized control in addressing traffic congestion.

Kartik et al. [18] present an innovative approach to adaptive signalized intersection control system utilizing Vehicular Ad hoc Networks (VANETs). This work explores the integration of VANET data to optimize traffic signal timings dynamically. Their adaptive control strategy aims to enhance vehicular movement efficiency and reduce blockage. By leveraging real-time information from connected vehicles, their method demonstrates the potential to substantially boost the performance of urban traffic management systems. The study contributes valuable insights into the application of VANETs for intelligent traffic signal control, offering promising solutions for mitigating traffic-related issues in urban environments.

McKenney et al. [19] focus on distributed and adaptive signalized intersection control systems within realistic traffic simulations. They explore the application of distributed control mechanisms using a sophisticated traffic simulation platform. This research contributes to the understanding of how adaptive signalized intersection control systems can effectively manage vehicular movement and congestion in complex urban environments.

Smith et al. [20] work on the Sure Trac adaptive traffic signal control system [1] is a notable contribution. This research, presented in the proceedings of the International Conference on Automated Planning and Scheduling in 2013, introduces the Surtrac system as an innovative approach to adaptive signalized intersection control systems. The system employs advanced AI and optimization techniques to dynamically optimize signal timings in response to live traffic conditions. This pioneering work showcases the potential for improving traffic flow, reducing congestion, and enhancing urban mobility through intelligent signal control systems.

Liu et al. [21] implemented a robust vehicle detection system based on the RCNN (Region-based Convolutional Neural Network) model. Notably, their RCNN model achieved an impressive accuracy rate of 98.2% in classifying and identifying aerial vehicles in challenging aerial images. This high level of accuracy underscores the effectiveness of their approach, making it a significant contribution to the field of aerial image analysis and object detection.

Sommer et al. [22] conducted a comprehensive performance comparison of their proposed vehicle detection method with previously established techniques from the literature. They observed that their Faster R-CNN-based approach outperformed existing methods significantly. For instance, when comparing with the method by Liu et al. [16], which utilized ICF and an AdaBoost classifier in a soft cascade structure, the proposed Faster R-CNN achieved substantial improvements in both recall rate (85.0%) and precision rate (92.8%). Similarly, their approach surpassed the results obtained in [21], where HOG+LBP and SVM were employed, achieving an average precision (AP) of 74.9% and 76.8% for VEDAI 512 and VEDAI 1024, respectively. This performance evaluation underscores the effectiveness and superiority of their Faster R-CNN-based automated vehicle identification system in remote sensing analysis.

Gleason et al. [23] proposed an approach based on template matching and color information to detect vehicles. This research addresses the challenge of detecting vehicles in aerial images, a crucial task for various applications like surveillance and traffic monitoring. The combination of template matching and color information demonstrated remarkable performance, achieving a correct classification rate of 98.9% for vehicles and 61.9% for background. This impressive result effectively reduced false positives by more than half while retaining nearly all true positives, highlighting the effectiveness of the proposed approach in accurately classifying vehicles in aerial imagery.

Yang et al. [24] made a significant stride in the domain of vehicle detection from aerial imagery, demonstrating an impressive accuracy rate of 89.44%. This remarkable achievement underscores the efficacy of their deep learning-based approach in accurately identifying and classifying vehicles within aerial images, showcasing its potential for various applications, including surveillance, traffic monitoring, and urban planning.

Alpatov et al. [25] presents a vehicle detection and counting system designed for real-time traffic surveillance. This system plays a pivotal role in enhancing traffic management and safety. By employing advanced techniques and technologies, it offers a robust solution for accurately detecting and counting vehicles in real-time. The research showcases the importance of such systems in modern traffic surveillance, where the need for efficient and reliable traffic management is paramount. Their approach provides valuable insights into the development of intelligent transportation systems and contributes to the advancement of traffic monitoring technologies.

Rahman et al. [26], a real-time wrong-way vehicle detection system based on the YOLO (You Only Look Once) model and centroid tracking is introduced. This research addresses a critical safety concern on roadways, as detecting wrong-way vehicles promptly is essential to prevent accidents and ensure the safety of all road users. By leveraging YOLO's capabilities for object detection and combining it with centroid tracking, the proposed system demonstrates the potential for accurate and efficient real-time detection of wrong-way vehicles. This work contributes to the field of intelligent transportation systems and highlights the importance of advanced computer vision techniques in addressing critical safety issues on the road.

Zheng et al. [27] achieved impressive performance metrics, with Precision, comprehensiveness, and quality benchmarks reaching approximately 98%, 93%, and 92%, respectively. These results highlight the high accuracy and reliability of the developed approach in detecting vehicles within high-resolution highway aerial images. Such robust performance is crucial for real-world applications, where precision and recall rates are vital for the success of traffic monitoring and management systems.

Razakarivony et al. [28] focuses on vehicle detection in aerial imagery, specifically addressing the challenging task of small target detection. This research presents a small target detection benchmark to evaluate vehicle detection algorithms operating on aerial images. The benchmark serves as a valuable resource for assessing the efficacy of diverse vehicle detection methodologies, especially in the context of detecting relatively small vehicles within large-scale aerial imagery. By providing a benchmark dataset and evaluation metrics, this work contributes to fostering the improvement of vehicle detection techniques, facilitating the creation of enhanced, precise, and dependable systems for aerial imagery analysis and applications such as surveillance, traffic monitoring, and urban planning.

Chabchoub et al. [29] presents an innovative approach to traffic light control using a combination of fuzzy logic and image processing techniques. This study addresses the need for intelligent traffic control device systems that exhibit responsiveness to dynamic traffic scenarios and optimize vehicular movement efficiently. By integrating fuzzy logic, which can handle complex decision-making processes, and image processing, which enables the system to collect data from the traffic scene, the envisioned signal control system offers an intelligent and adaptive solution. Such systems have the potential to optimize vehicular flow, mitigate traffic congestion, and enhance road safety measures. The work by Chabchoub and his team contributes to the field of smart mobility solutions, offering a new direction for the development of efficient and responsive traffic light control mechanisms.

2.1.1 Literature Summary Table

TABLE 2.1. LITERATURE SUMMARY

SNo	Author	Year	Algorithm/ Technique	Dataset / Simulator	Result
1.	Padilla et al.	2023	Tiny-YOLO	PKLot	mAP50 → 99.85% mAP95 → 99.79%
2.	Zarei et al.	2022	SSAM-YOLO	CDNet2014	Avg. Precision → 94.06%
3.	Zuraimi et al.	2021	YOLOv4	*Private Dataset*	mAP50 → 82.08%
4.	Gorodokin et al.	2021	YOLOv4	*Private Dataset*	Accuracy → 92%
5.	Miao et al.	2020	YOLOv3	*Private Dataset*	Accuracy → 93.66%
6.	Chen et al.	2019	YOLOv3-tiny	*Private Dataset*	mAP50 → 87.79%
7.	Dong et al.	2022	YOLOv5	PASCAL VOC, MS COCO	mAP50 → 72.4% mAP95 → 50.5%
8.	H. Song et al.	2019	YOLOv3	*Private Dataset*	mAP → 87.88%
9.	Khalifa et al.	2022	YOLOv5s	*Private Dataset*	mAP → 97.8%
10.	Chen et al.	2023	YOLOv4	Cityscapes	mAP → 86.22%
11.	Wu et al.	2020	MARDDPG	SUMO Simulator	AWT → 2.59 ± 0.89 AQL → 3.04 ± 5.19
12.	Ge et al.	2019	QT-CDQN	SUMO Simulator	AWT → 9.7 ± 0.29 AQL → 2.32 ± 0.02
13.	Wang et al.	2020	Co-DQL	SUMO Simulator	AWT → 36.981 ± 0.509

14.	Liang et al.	2019	3DQN	SUMO Simulator	AWT reduced by 25.7%
15.	Tajalli et al.	2020	DOCA	Vissim	AWT reduced by 5.3%
16.	Cai et al.	2009	ADP	OPAC	AWT reduced by 67%
17.	Priemer et al.	2009	MDP	AIMSUN NG	AWT reduced by 25%
18.	Kartik et al.	2013	OAF	VANET	AWT reduced by 30%
19.	McKenney et al.	2013	-	SUMO	Performance increase of 7.36%
20.	Smith et al.	2013	SURTRAC	-	Traffic flow efficiency increased by 25-40%.
21.	Liu et al.	2015	R-CNN	*Private Dataset*	Accuracy → 98.2%
22.	Sommer et al.	2017	Faster R-CNN	VEDAI	mAP → 76.8%
23.	Gleason et al.	2011	R-CNN	*Private Dataset*	Accuracy → 98.9%
24.	Yang et al.	2018	DFL-CNN	ITCVD	Accuracy → 89.44%
25.	Alpatov et al.	2018	-	-	Accuracy → 99.69%
26.	Rahman et al.	2020	YOLOv3	*Private Dataset*	Accuracy → 99%
27.	Zheng et al.	2013	R-CNN	HRTPO	Accuracy → 92%
28.	Razakarivony et al.	2016	Random Hough Forest	VEDAI	mAP50 → 90.5%
29.	Chabchoub et al.	2021	Fuzzy Logic	MATLAB	-

2.2 Suggestions based on Literature Survey

Based on the extensive literature survey, several valuable suggestions and insights emerge to inform the development of our dynamic traffic signal management system:

1. **Leverage Efficient Vehicle Detection Models:** Consider adopting efficient vehicle detection models such as T-YOLO introduced by Padilla et al. [1] or Fast-Yolo-Rec by Zarei et al. [2]. These models strike a balance between accuracy and processing speed, making them suitable for real-time traffic analysis.

2. **Prioritize YOLO Models for Accuracy and Speed:** The studies by Zuraimi et al. [3] and Dong et al. [7] showcase the significance of YOLO models, particularly YOLOv4 and YOLOv5, in achieving high accuracy and real-time processing capabilities. These models should be explored for their potential to enhance vehicle detection.

3. **Explore Cooperative Algorithms:** Investigate the potential of cooperative algorithms for traffic signal control, as demonstrated by Ge et al. [12] and Wang et al. [13]. Cooperative reinforcement learning and decentralized signal control strategies can optimize traffic flow efficiently.

4. **Adapt to Challenging Lighting Conditions:** Learn from the research of Miao et al. [5] and Khalifa et al. [9] to address challenging lighting conditions for vehicle detection. Adaptive algorithms and preprocessing techniques can improve accuracy in scenarios with varying light levels.

5. **Incorporate Adaptive Traffic Management:** Inspired by Liang et al.'s [14] work, consider the implementation of responsive traffic management solutions. These systems can dynamically adjust signal timings based on real-time vehicle density analysis, optimizing urban traffic efficiency.

CHAPTER 3
PROBLEM STATEMENT AND METHODOLOGY

3.1 Problem Statement

Urban areas frequently grapple with the pervasive issue of traffic congestion, manifesting as extended travel times, fuel inefficiency, and a compromised overall transportation system efficiency. The conventional traffic signal systems, often governed by fixed time intervals, fall short in addressing the dynamic and ever-changing nature of urban traffic patterns. This limitation gives rise to persistent traffic bottlenecks and substantial delays for commuters.

The crux of the problem lies in the inherent inflexibility of traditional traffic signal systems. These systems rely on predetermined time schedules, failing to adapt swiftly to fluctuating traffic volumes and emerging congestion hotspots. As a result, they contribute to increased fuel consumption, elevated greenhouse gas emissions, and heightened frustration among commuters.

In this context, it becomes evident that a pressing need exists for an innovative traffic management approach capable of dynamically responding to real-time traffic conditions. This research seeks to address this pressing problem by exploring the fusion of Intelligent Traffic Systems (ITS) and advanced computer vision techniques to create a traffic signal management system that adapts intelligently to urban traffic dynamics. The goal is to alleviate traffic congestion, optimize travel times, enhance fuel efficiency, and ultimately usher in a more sustainable and efficient urban transportation ecosystem.

3.2 Methodology

The proposed methodology employs a structured flow of execution. It begins with the capture of real-time images from CCTV cameras at traffic signal intersections. These images are then processed using computer vision's object detection technique, facilitated by advanced vehicle detection models, to identify vehicles and calculate real-time traffic density. Subsequently, the traffic density data is transmitted to a central server for analysis. Within this server, sophisticated algorithms process the data to ascertain the most efficient durations for green signal allocation in each lane, taking into account the current traffic conditions. These calculated signal timings are then communicated back to the traffic signal controllers, which promptly update the signal timers accordingly. This dynamic approach enables real-time adaptation of traffic signals, effectively optimizing traffic flow, reducing congestion, and accommodating the presence of connected and automated vehicles (CAVs) and emergency vehicles. For a visual representation of this process, refer to Figure 3.1, which provides an abstract overview of the project's execution flow.

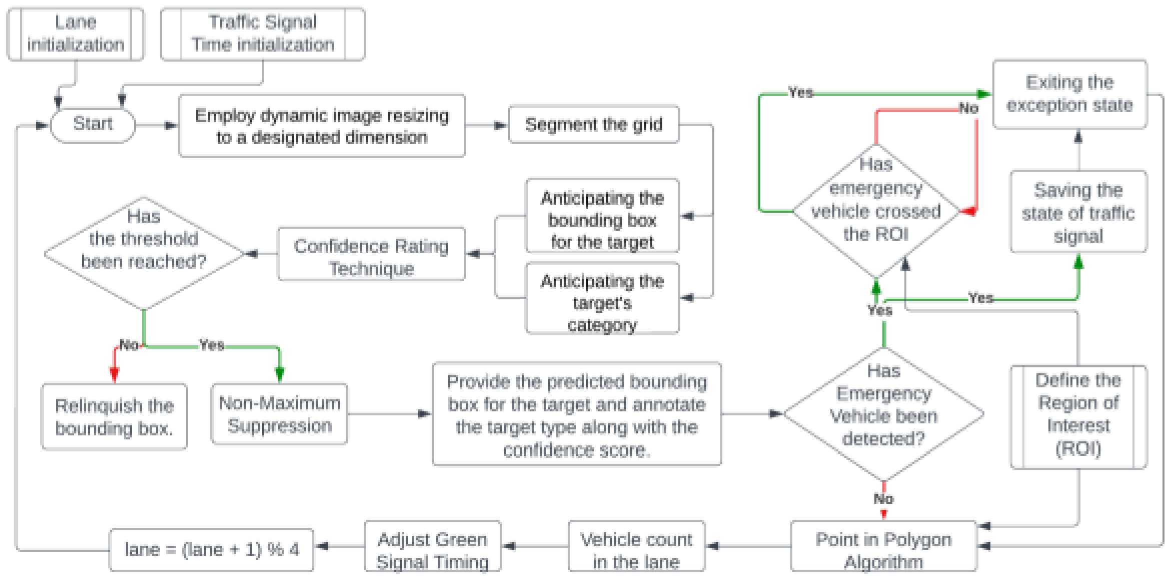


Fig 3.1. Execution Flow of the Project

3.3 Object Detection

Object recognition stands as a crucial computer vision objective, entailing the automated identification and precise determination of entities within images or video frames. This task is instrumental in numerous applications, from autonomous vehicles and surveillance systems to medical imaging and retail analytics. Modern object detection relies heavily on deep learning techniques, particularly Convolutional Neural Networks (CNNs), for precise localization & accurate classification of objects.

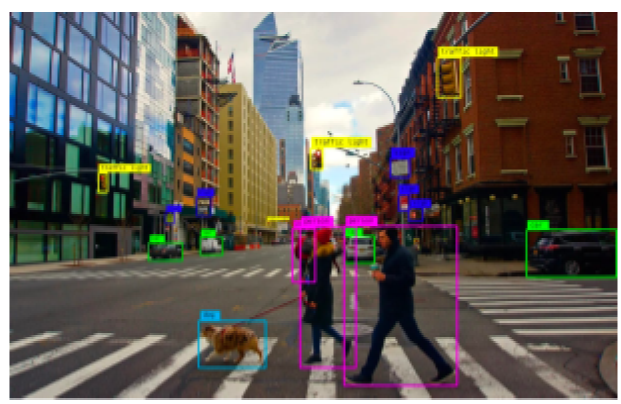


Fig 3.2. Depiction of Object Detection

Object detection's versatility extends to diverse domains, enabling applications such as autonomous navigation, security surveillance, medical diagnostics, and traffic management, where it plays a vital role in optimizing urban mobility and safety through real-time traffic analysis and signal control.

3.3.1 YOLO

The You Only Look Once (YOLO) methodology presents a neural network that concurrently anticipates bounding boxes and class probabilities. This distinguishes it from earlier object detection algorithms that repurposed classifiers for similar tasks. This unique methodology has propelled YOLO to attain remarkable outcomes, surpassing alternative real-time object detection algorithms by a considerable margin. In contrast to techniques like Faster RCNN, which employ Region Proposal Networks to identify candidate regions of interest before conducting independent recognition on those regions, YOLO consolidates its predictions by unifying them through a singular fully connected layer. While methodologies utilizing Region Proposal Networks require numerous iterations for a given image, YOLO achieves its goals in a singular pass. Since its original release in 2015, various enhanced versions of YOLO have emerged, each building upon and refining its predecessor. The following timeline offers a glimpse into the ongoing development and advancements in YOLO's capabilities in recent years.

3.3.2 YOLO Architecture

The YOLO algorithm initiates by inputting an image and then utilizes a streamlined deep convnet to detect objects within the image. The configuration of the convolutional neural network model, fundamental to YOLO, is illustrated in the following diagram.

YOLO employs a pre-trained model with the first 20 convolution layers initially trained on ImageNet, which is then converted for object detection. The model's final layer predicts class probabilities and bounding box coordinates. In YOLO, the input image is divided into an $S \times S$ grid, with each grid cell responsible for detecting objects. Predictions are made for B bounding boxes and confidence scores.

During training, responsibility is assigned to the predictor with the highest Intersection over Union (IOU) with the ground truth to specialize bounding box predictors and enhance recall. To improve accuracy and efficiency, YOLO uses non-maximum suppression (NMS) for post-processing, eliminating superfluous or inaccurate bounding regions and retaining individual bounding regions for every object in the image.

3.3.3 Performance evaluation metrics

For assessing & contrasting predictive proficiency of various target identification frameworks, it is essential to rely on established quantitative criteria. Among the most prevalent evaluation metrics, we commonly employ Intersection over Union (IoU) and the Mean Average Precision (mAP) metrics.

3.3.3.1 Intersection Over Union (IoU)

Intersection over Union (IoU) serves as a widely used metric for assessing the precision of localization and quantifying localization discrepancies within target identification framework models. To calculate the Intersection over Union (IoU) between predicted and actual bounding boxes, the initial phase entails evaluating the shared area between the respective bounding boxes associated with a given object. Subsequently, we determine the cumulative region encompassed by these two bounding regions, designated as the "Union," denoting their collective area, and the overlapping region termed the "Intersection". By dividing the intersection by the Union, we obtain a fraction that reflects the extent of confluence relative to the overall area, offering a valuable measure of how closely the predicted bounding region aligns with the original bounding region. In object detection, precision and recall are used for bounding region predictions rather than class predictions. A positive prediction is indicated by an IoU (Intersection over Union) value > 0.5, while a value < 0.5 signifies a negative prediction.

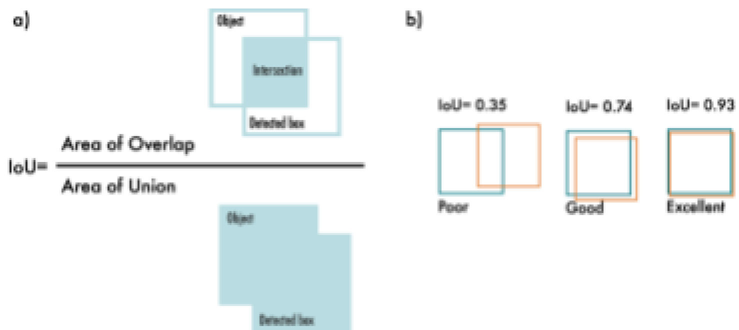


Fig 3.3. Intersection Over Union

3.3.3.2 Mean Average Precision (mAP)

The Average Precision (AP) is determined by computing the integral of the curve that depicts precision against recall for a given array of prognostications. Recall is found by dividing the total inferences generated by the model for a specific class by the total existing labels for that class. Precision, on the other hand, is the proportion of accurate positive predictions to the entirety of model predictions, often denoted as the "true positive rate".

Precision and recall exhibit a trade-off, which is graphically illustrated by adjusting the classification threshold. The Average Precision for a model per class is obtained by calculating the area under the precision vs. recall curve. The mean Average Precision (mAP) is then computed as the average of these per-class values across all classes.

$mAP = \frac{1}{n}\sum_{k=1}^{k=n} AP_k = \frac{1}{n}\sum_{k=1}^{k=n} \frac{\lvert TP_k \rvert}{\lvert FP_k \rvert + \lvert TP_k \rvert}$	$AP_k \rightarrow$ The avg.precision of class k n $\rightarrow$ The number of classes TP $\rightarrow$ True Positive FP $\rightarrow$ False Positive

3.4 Dynamic Traffic Signal Duration Management

Dynamic traffic signal management, as opposed to fixed-time systems, adjusts signal timings in real-time based on live traffic conditions. It leverages data-driven insights like vehicle density and traffic volume to optimize traffic flow and reduce congestion. In our research, YOLOv8 is integrated for real-time vehicle density analysis to enhance urban traffic control.

3.4.1 Four Phase Cycle

The four-phase traffic signal cycle represents a fundamental approach to traffic signal control, particularly at intersections. In this cycle, traffic movements are organized into four distinct phases, each with its own dedicated green signal interval as shown in Figure 3.4.

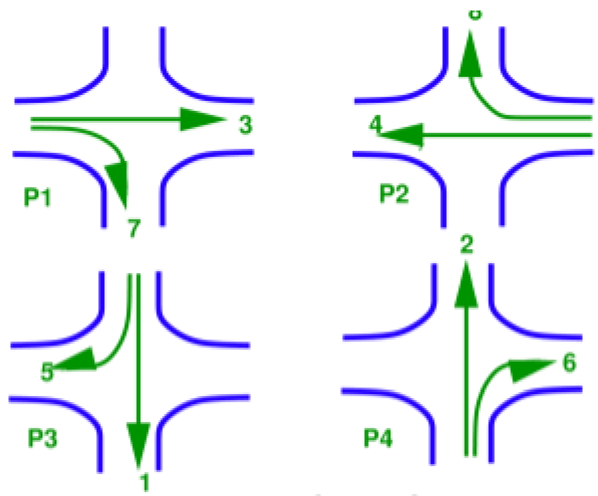


Fig 3.4. Four Phase Cycle of Traffic Intersection

For four-phase cycle we have some important terms:

3.4.1.1 Cycle Time

Cycle time (as shown in Figure 3.5) is the duration it takes for a traffic signal to accomplish a full iteration operational cycle, denoted as "c." It encompasses all signal phases and intervals and is vital for optimizing traffic flow and intersection efficiency. Traffic engineers adjust cycle times to match traffic conditions.

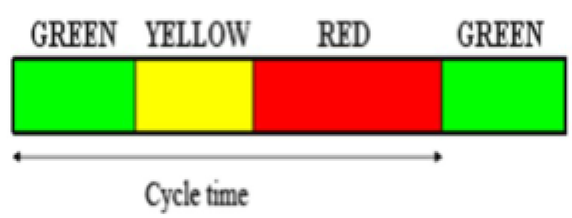


Fig 3.5. Cycle Time

3.4.1.2 Headway

Headway, expressed as the inter-vehicle time passing a specific point on the road, is a critical parameter in traffic management. The initial headway is defined as the duration between the commencement of the green signal and the moment a vehicle crosses the curb line. The subsequent headway is described as the time gap between the passage of the first and second vehicles across the curb line, and so forth. The successive headways can be plotted as shown in Figure 3.6. After a few vehicles the headway becomes constant, this state is known as Saturation headway (denoted by "h").

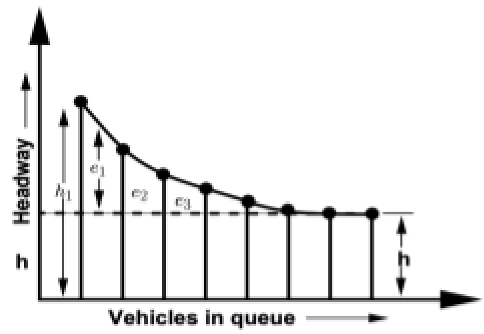


Fig 3.6. Headway Graph

If every vehicle needs *h seconds* of green time and if signals were always green, then *s vehicles/hr* would pass the intersection.

$s = \frac{3600}{h}$ (Equation 1)	s → saturation flow rate in vehicles per hour of green time per lane h → saturation headway in seconds

Then, the Start-up loss time is:

$L_i = \sum_{j=1}^{n} e_j$ (Equation 2)	e → Difference between actual headway and h's for the i^{th} vehicle

And, the green time required to clear *N* vehicles can be found out as:

$T = L_i + h * N$ (Equation 3)	T → Green time required to clear *N* vehicles

3.4.1.3 Green Signal Time

determine the optimal duration for the green signal based on the quantity of vehicles within each category at a signalized intersection, we utilized the initial speeds and acceleration periods of vehicles. This information enabled us to estimate the average duration it takes for each vehicle class to traverse the intersection. The computation of the green signal time is subsequently derived through the application of the following formula:

$$GST = \frac{\sum_{vehicleClass} (NoOfVehicles_{vehicleClass} * AverageTime_{vehicleClass})}{NoOfLanes}$$

GST → is green signal time

noOfVehiclesOfClass → is the count of vehicles within each vehicle class at the signal, as identified by the vehicle detection module

averageTimeOfClass → represents the mean duration required for vehicles belonging to that category to traverse an intersection

noOfLanes → denotes the total count of lanes at the intersection

The optimal traversal time for each vehicle class at an intersection can be configured based on various criteria, such as geographic region, city boundaries, specific localities, or even the unique characteristics of each intersection. This tailored approach enhances the efficacy of traffic control. Analysis of relevant data provided by transportation authorities facilitates this customization.

The signal transitions occur in a sequential manner rather than prioritizing the direction with the highest density. This aligns with the existing system, where signals sequentially transition to green without requiring individuals to adjust their behavior or causing confusion. The signal sequence remains consistent with the current system, and considerations have been made for the inclusion of yellow signals.

Order of signals: Red → Green → Yellow → Red

CHAPTER 4
EXPERIMENTAL WORK

In this section, we move from theory to practice as we implement and evaluate our proposed dynamic traffic signal management system. Through a combination of real-time CCTV camera feeds and carefully designed simulations, we aim to demonstrate the effectiveness of our approach in optimizing traffic signal durations based on vehicle density. This section outlines our methodology, datasets, and tools, offering insights into the technical aspects of our research and showcasing its real-world applicability.

4.1 System Requirements

4.1.1 Software Requirements

Listed below are the software requirements for performing project on *Machine Vision based Dynamic Traffic Signal Duration Management and Synchronization* for training & validating computer vision model and simulating the dynamic traffic intersection model:

4.1.1.1 Computer Vision Model

1. **Operating System:** Operating system acts as the interface between the user programs and the kernel. Windows 8 and above (64 bit) operating system is required or macOS Catalina is required.
2. **Python Kernel:** Python stands out as a versatile and user-friendly programming language, valued for its straightforwardness and clear syntax. The requisite Python version for this context is 3.11.1.
3. **Google Colab:** Google Colab, abbreviated from Google Colaboratory, is a complimentary cloud-centric platform by Google, delivering a Jupyter Notebook atmosphere tailored for executing Python scripts.
4. **Anaconda:** Anaconda represents a freely accessible distribution of Python and R programming languages tailored for scientific computing. Its primary objective is to streamline the processes of package management and deployment, with version control facilitated through the conda package management system.
5. **Jupyter Notebook:** The Jupyter Notebook serves as an open-source web platform enabling the creation and dissemination of documents incorporating executable code, mathematical expressions, visualizations, and explanatory text. Applications encompass tasks such as data cleansing and transformation, numerical simulations, statistical

modeling, data representation, machine learning, and a diverse range of other computational activities.

4.1.1.2 Simulating Traffic

1. **Python Kernel:** Python is a high-level, versatile programming language known for its simplicity and readability. Python version 3.11.1 is required.
2. **Pygame module:** Pygame is a popular Python library designed for creating 2D games and multimedia applications. It provides developers with the tools and functions needed to develop interactive and graphical applications, making it an excellent choice for hobbyist game development, educational projects, and even prototyping. Pygame version 2.5.2 is required.

4.1.2 Hardware Requirements

- Processor: Intel i5 2.5GHz upto 3.5GHz (or AMD equivalent)
- GPU (preferred): dedicated GPU from NVIDIA or AMD with 4GB VRAM
- Memory: minimum 8GB RAM
- Secondary Storage: minimum 128GB SSD or HDD
- Network Connectivity: bandwidth ~ 10 Mbps to 75 Mbps

4.2 Dataset

Fig 4.1. A Traffic Intersection equipped with two Different Capturing viewpoints by mounting a camera in existing infrastructure and using a Drone equipped with a Camera

The Multi-view Traffic Intersection Dataset, abbreviated as MTID [30], stands as a pivotal resource in the realm of traffic surveillance and analysis. This dataset distinguishes itself by capturing the same traffic intersection from two distinct viewpoints shown in Fig 4.1, providing invaluable insights into traffic behavior and vehicle movements. Recognizing the fundamental importance of camera positioning in traffic surveillance, MTID offers synchronized data from two vantage points: one from a camera mounted on existing infrastructure and the other from a drone. This dataset encompasses 3100 frames from each viewpoint, yielding a total of 6200 frames meticulously annotated with 18883 individual annotations for the infrastructure viewpoint and 50274 individual annotations for the drone viewpoint as shown in Table . These annotations, performed with pixel-level precision, include axis-aligned bounding boxes and pixel-level masks, allowing for precise instance segmentation. Such comprehensive pixel-wise annotation is imperative for accurate analysis, especially in scenarios where the exact positions of road users are critical, such as collision analysis.

The annotation scheme within the MTID dataset adheres to the COCO format and classes, encompassing four distinct annotations: bicycle, car, bus, and lorry. These annotations facilitate the identification and tracking of various vehicle types and bicycles within the traffic scenes. The dataset's data collection occurred at an intersection situated in downtown Aalborg, Denmark, employing two different cameras. The infrastructure viewpoint utilized an AXIS M1124-E camera with VGA resolution at 30 FPS, while the drone viewpoint employed a DJI Mavic Pro Drone with its standard on-board camera, capturing footage in full HD resolution at 30 FPS.

TABLE 4.1. Overview of the Dataset

	Infrastructure Viewpoint	**Drone Viewpoint**
Resolution	640x480	1920x1080
Number of frames	3100	3100
Bicycles Annotated	2003	2909
Cars Annotated	9989	32550
Buses Annotated	2113	3607
Lorries Annotated	4778	11208
Total Annotations	18883	50274

4.3 Data Preprocessing

4.3.1 Conversion from MSCOCO to YOLO Labels

Converting labels from MSCOCO (Microsoft Common Objects in Context) to YOLO (You Only Look Once) format is a critical step in aligning our chosen object detection dataset with the requirements of YOLO-based models. This transformation involves mapping class labels to numeric indices, converting bounding box coordinates to normalized form, and

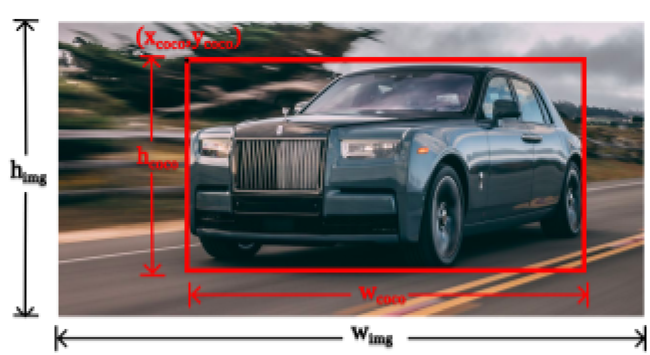


Fig 4.2 Bounding Box in COCO Format

organizing the labels in YOLO-specific text file structures. To streamline this process, custom scripts in languages like Python can be employed for automation. Verification and quality assurance checks are essential post-conversion to ensure label accuracy and adherence to YOLO's labeling standards. This conversion is pivotal in making the dataset compatible with our dynamic traffic signal management system and YOLO-based object detection framework.

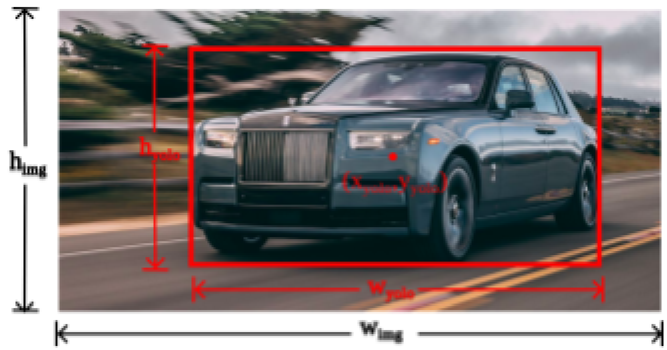


Fig 4.3. Bounding Box in YOLO Format

The following mathematical conversion was applied:

$x_{yolo} = \frac{(x_{coco} + \frac{w_{coco}}{2})}{w_{img}}$	$w_{yolo} = \frac{w_{coco}}{w_{img}}$
$y_{yolo} = \frac{(y_{coco} + \frac{h_{coco}}{2})}{h_{img}}$	$h_{yolo} = \frac{h_{coco}}{h_{img}}$

4.3.2 Splitting Dataset

The provided code snippet serves to organize and split a dataset into training and validation sets for use in machine learning applications, particularly object detection tasks like YOLO. It does so by initially defining directory structures for the original data and destination directories for the training and validation data. Subsequently, the code shuffles and splits the image files into two sets based on an 80:20 ratio, moving them to their respective sub-directories within the training and validation directories. Furthermore, it renames the sub-directories to lowercase, a common convention for label and image directories in object detection tasks. This preparatory step is crucial for structuring the

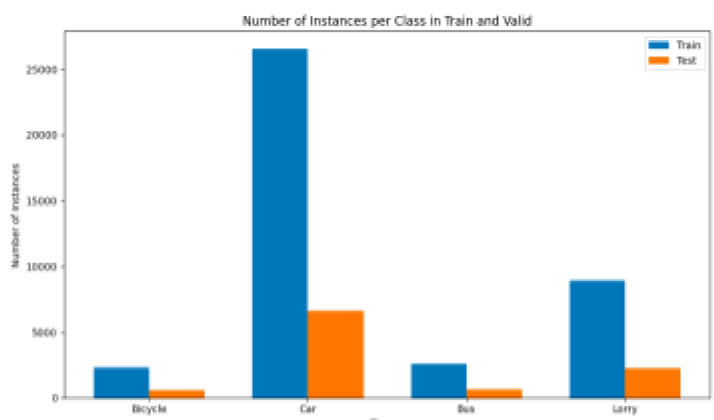


Fig 4.4. Number of Instances per Class in Train and Test

dataset in a manner compatible with YOLO and similar object detection frameworks. The number of instances per class in Train and Valid is shown in Figure 4.4.

4.4 Training the YOLOv8 Model

4.4.1 Architecture of the Model

The architecture of the model described is a deep neural network designed for object detection, often used in applications like YOLO (You Only Look Once). This model consists of 225 layers and approximately 11.1 million parameters, making it a complex and powerful architecture. The structure includes convolutional layers, concatenation layers, and upsampling layers, which are crucial for feature extraction, multiscale detection, and object localization. The model's architecture is designed to process input images and produce bounding box predictions for objects of interest. It employs techniques such as spatial pyramid pooling (SPP) and feature concatenation to capture contextual information and enhance detection accuracy. With a capability of 28.7 GFLOPs (Giga Floating-Point Operations Per Second), this model demonstrates its efficiency in handling computationally intensive tasks like object detection. It serves as a fundamental component in the object detection pipeline, enabling precise identification and localization of objects within images.

	from	n	params	module	arguments
0	-1	1	928	ultralytics.nn.modules.Conv	[3, 32, 3, 2]
1	-1	1	18560	ultralytics.nn.modules.Conv	[32, 64, 3, 2]
2	-1	1	29056	ultralytics.nn.modules.C2f	[64, 64, 1, True]
3	-1	1	73984	ultralytics.nn.modules.Conv	[64, 128, 3, 2]
4	-1	2	197632	ultralytics.nn.modules.C2f	[128, 128, 2, True]
5	-1	1	295424	ultralytics.nn.modules.Conv	[128, 256, 3, 2]
6	-1	2	788480	ultralytics.nn.modules.C2f	[256, 256, 2, True]
7	-1	1	1180672	ultralytics.nn.modules.Conv	[256, 512, 3, 2]
8	-1	1	1838080	ultralytics.nn.modules.C2f	[512, 512, 1, True]
9	-1	1	656896	ultralytics.nn.modules.SPPF	[512, 512, 5]
10	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
11	[-1, 6]	1	0	ultralytics.nn.modules.Concat	[1]
12	-1	1	591360	ultralytics.nn.modules.C2f	[768, 256, 1]
13	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
14	[-1, 4]	1	0	ultralytics.nn.modules.Concat	[1]
15	-1	1	148224	ultralytics.nn.modules.C2f	[384, 128, 1]
16	-1	1	147712	ultralytics.nn.modules.Conv	[128, 128, 3, 2]
17	[-1, 12]	1	0	ultralytics.nn.modules.Concat	[1]
18	-1	1	493056	ultralytics.nn.modules.C2f	[384, 256, 1]
19	-1	1	590336	ultralytics.nn.modules.Conv	[256, 256, 3, 2]
20	[-1, 9]	1	0	ultralytics.nn.modules.Concat	[1]
21	-1	1	1969152	ultralytics.nn.modules.C2f	[768, 512, 1]
22	[15, 18, 21]	1	2117596	ultralytics.nn.modules.Detect	[4, [128, 256, 512]]

Model summary: 225 layers, 11137148 parameters, 11137132 gradients, 28.7 GFLOPs

Fig 4.5. Architecture of the Model

4.4.2 Optimizer for training

The chosen optimizer for training this model is Stochastic Gradient Descent (SGD) with a learning rate (lr) set to 0.01. This optimizer is a popular choice in deep learning and is effective for updating model weights during the training process. To manage weight decay and regularization, SGD employs specific parameter groups. Parameter Group 57 has a weight decay of 0.0, indicating no regularization for the corresponding layers. Parameter Group 64 uses a weight decay of 0.0005 and is likely associated with layers where weight regularization is essential to prevent overfitting. Parameter Group 63 is related to bias terms and applies weight decay to regularize them. This combination of parameter groups and weight decay settings aims to strike a balance between fitting the training data well and preventing overfitting, resulting in a well-generalized object detection model.

4.4.3 Albumentations for training

During the training process, the Albumentations library is utilized to augment and diversify the training data. These enhancements are imperative for fortifying the model's capacity to generalize effectively across diverse real-world scenarios. Several specific augmentation techniques are applied to the training data. Firstly, there is a blur augmentation (Blur) with a probability (p) of 0.01, and the blur intensity ranges between 3 and 7. Additionally, the MedianBlur augmentation (p=0.01) with similar blur intensity settings is employed. To further enhance the data, a grayscale transformation (ToGray) with a probability of 0.01 is applied, which can help the model adapt to variations in lighting conditions. Lastly, Contrast Limited Adaptive Histogram Equalization (CLAHE) is employed with a probability (p) of 0.01. This technique enhances local contrast in the images, making object features more discernible. The CLAHE parameters include the clip limit, which ranges from 1 to 4.0, and the tile grid size is set at 8x8. These augmentations collectively contribute to a more robust and adaptable object detection model during training.

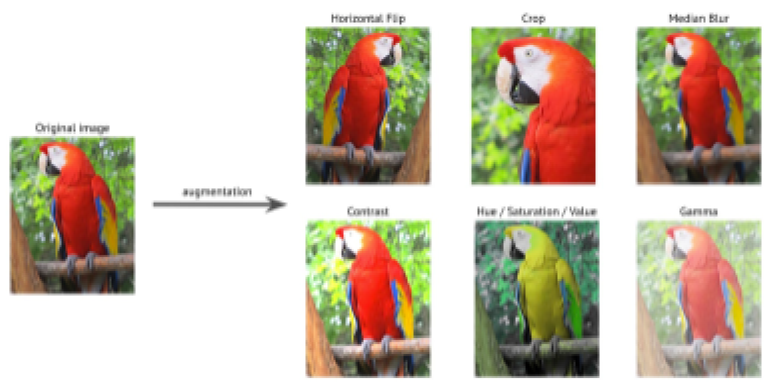


Fig 4.6. Albumentations

4.5 Training Evaluation and Analytics Metrics

In the domain of computer vision and YOLO (You Only Look Once) model training, performance evaluation is crucial. It involves assessing the model's proficiency in precise object detection and classification in images and videos. This section explores key evaluation metrics and analysis techniques, such as the Confusion Matrix, F1 Curve, P Curve, PR Curve, and R Curve. These metrics provide valuable insights into the YOLO model's real-world effectiveness.

4.5.1 Confusion Matrix

The Confusion Matrix, in this context, is a 5x5 matrix that helps assess the performance of YOLO models in detecting objects belonging to the classes 'Bicycle,' 'Car,' 'Bus,' 'Lorry,' and the 'Background.' It offers an all-encompassing perspective on the degree of alignment between the model's predictions and the actual data for each category.

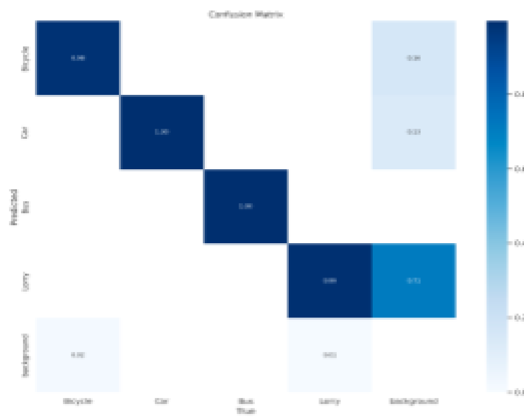


Fig 4.7. Confusion Matrix

4.5.2 F1-Confidence Curve

The F1-Confidence Relationship Graph visually depicts how adjustments in the confidence threshold impact the F1-score within the context of object detection in YOLO models. The F1-score is a measure of the model's accuracy in simultaneously considering both precision and recall, making it a valuable metric for assessing object detection performance. By varying the confidence threshold, which determines when a detection is considered valid, this curve provides insights into how the model's precision and recall trade off against each other at different confidence levels. It helps users identify an optimal threshold that balances the need for high precision and high recall, depending on their specific application requirements. This curve is a valuable tool for fine-tuning YOLO models to achieve the desired balance between accuracy and robustness in object detection.

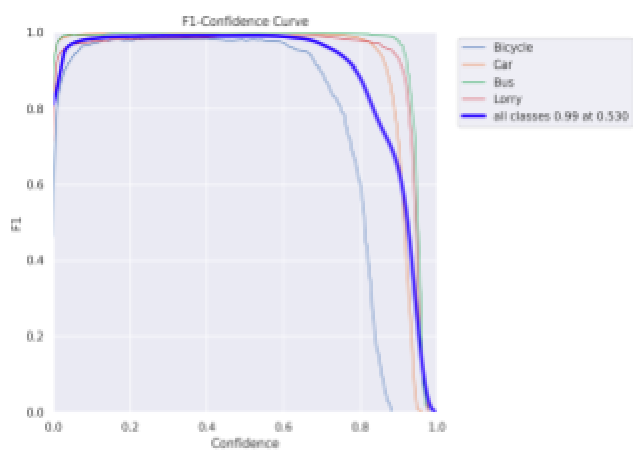


Fig 4.8. F1-Confidence Curve

4.5.3 Precision-Confidence Curve

The Precision-Confidence Curve is a vital visualization tool for evaluating the precision of a YOLO model's predictions across varied confidence levels. This graphical representation aids in comprehending the fluctuation in the model's accuracy as the confidence threshold for object detection undergoes adjustment. By plotting precision against confidence scores, we can identify the threshold values that yield the desired level of precision for specific tasks or applications. This curve is particularly valuable for making informed decisions about trade-offs between precision and recall, allowing users to fine-tune the model's performance based on their specific requirements.

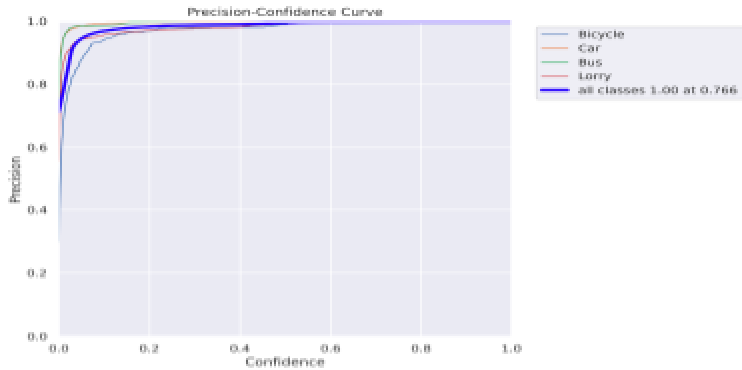


Fig 4.9. Precision-Confidence Curve

4.5.4 Precision-Recall Curve

The Precision-Recall Curve is a vital evaluation tool in assessing the YOLO model's effectiveness in object localization. This graphical representation illustrates the balance between accuracy and inclusivity at varying confidence thresholds, providing a visual depiction of the model's capacity to accurately identify objects while maximizing the retrieval of pertinent instances. Precision represents the proportion of true positive detections among all positive predictions, emphasizing the model's accuracy. Recall, on the other hand, quantifies the model's ability to identify all actual positive instances within the dataset, focusing on its completeness. By analyzing this curve, we gain insights into how the model's confidence threshold impacts its precision and recall rates, allowing us to make informed decisions about the ideal threshold for our specific application.

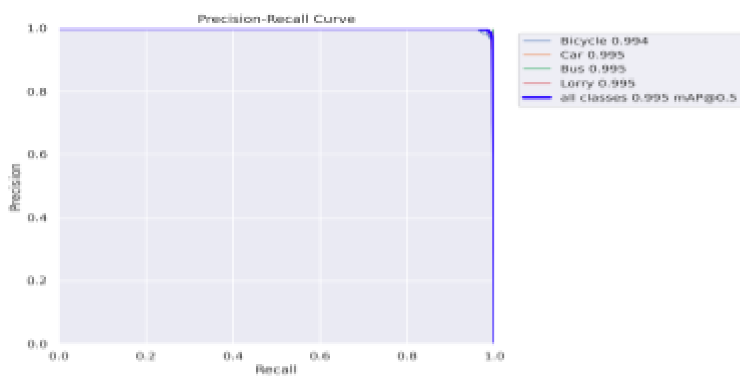


Fig 4.10. Precision-Recall Curve

4.5.5 Recall-Confidence Curve

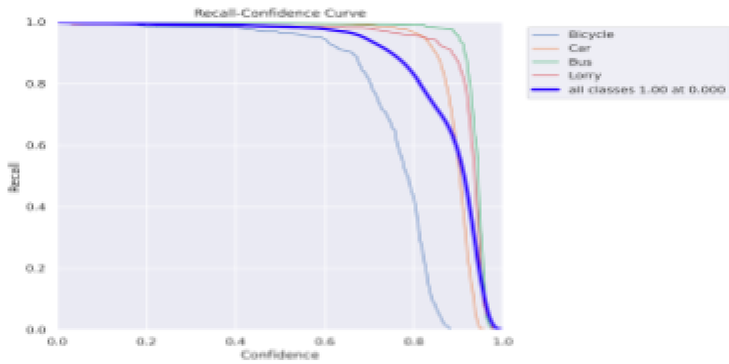


Fig 4.11. Recall-Confidence Curve

The Recall-Confidence Curve is a crucial visualization that showcases the recall of the YOLO model across different levels of confidence thresholds. In this curve, the x-axis represents the varying confidence thresholds, while the y-axis represents the corresponding recall values. The recall measures the ability of the model to correctly detect all instances of a specific class within the dataset. By analyzing this curve, we can gain insights into how different confidence thresholds impact the model's recall performance for each object class. This information is invaluable for fine-tuning the model's detection capabilities and optimizing its performance for specific classes or applications.

4.5.6 Training result

Metrics play a pivotal role in assessing the effectiveness of the YOLO model in object detection tasks, aiding in both training and evaluation phases:

- **Train/Box_Loss:** The train/box_loss is a critical metric during the training phase of the YOLO model. It measures the loss associated with bounding box predictions. Lower values indicate that the model is effectively learning to localize objects in the images.
- **Train/Cls_Loss:** Train/cls_loss quantifies the loss related to class predictions during model training. It signifies how well the model is learning to classify objects. A decreasing cls_loss indicates improved class prediction accuracy.
- **Train/Dfl_Loss:** The train/dfl_loss measures the loss associated with depth-wise focal loss during training. This is especially relevant for object detection tasks where certain classes may be challenging to distinguish, and it helps the model focus on difficult cases.
- **Metrics/Precision(B):** Precision(B) is a crucial metric calculated during evaluation. It signifies the precision of bounding box predictions. High precision values indicate that a majority of the predicted bounding boxes are accurate.
- **Metrics/Recall(B):** Recall(B) is another evaluation metric. It measures the ability of the model to recall or detect all relevant objects in the dataset. A high recall value suggests that the model is effective at capturing most instances of objects.

- **Val/Box_Loss:** Similar to train/box_loss, val/box_loss measures the bounding box prediction loss during the validation phase. It helps assess how well the model generalizes to unseen data.
- **Val/Cls_Loss:** Val/cls_loss quantifies the class prediction loss on the validation dataset. It provides insights into the model's ability to classify objects accurately on unseen data.
- **Val/Dfl_Loss:** Val/dfl_loss is the depth-wise focal loss computed during validation. It ensures that the model maintains focus on challenging cases when making predictions on new data.
- **Metrics/mAP50(B):** Metrics/mAP50(B) stands for "mean Average Precision at IoU threshold of 0.5" for bounding boxes. It calculates the average precision for different object classes, considering predictions with an Intersection over Union (IoU) of 0.5 or higher as correct detections.
- **Metrics/mAP50-95(B):** Metrics/mAP50-95(B) represents the mean Average Precision over a range of IoU thresholds from 0.5 to 0.95. It provides a comprehensive evaluation of the model's performance, considering various IoU thresholds.

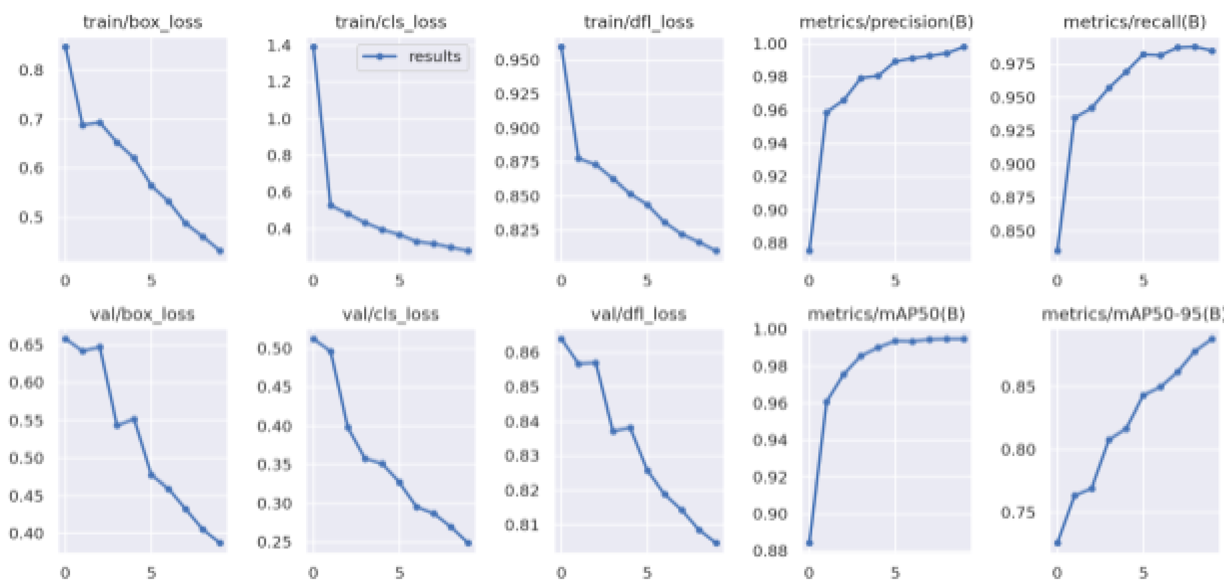


Fig 4.12. Training Result

CHAPTER 5
RESULT AND DISCUSSION

In this section, we delve into the outcomes of our comprehensive experiments and analyses conducted with the YOLO (You Only Look Once) entity identification model. The preceding sections have detailed the architecture, training methodologies, and evaluation metrics. Now, it is time to unveil the fruits of our labor and engage in insightful discussions. We present the model's computational effectiveness on diverse entity identification tasks, showcasing its ability to precisely localize & classify objects within images. Furthermore, we explore the implications of the obtained results, considering the challenges and opportunities presented by the YOLO model. Through a detailed examination of the results and discussions, we aim to provide valuable insights into the capabilities and limitations of YOLO for real-world object detection applications.

5.1 Model Performance Summary

After training and evaluating the YOLO object detection model on our dataset, we have obtained a comprehensive performance summary as shown in Figure 5.1. The model's performance is assessed across various classes, including "Bicycle," "Car," "Bus," "Lorry," and the "Background" class. The summary includes key metrics such as precision (P), recall (R), mean Average Precision at IoU threshold of 0.5 (mAP50), and mean Average Precision within the IoU range of 0.5 to 0.95 (mAP50-95). These metrics provide valuable insights into the classification precision for diverse object recognition. It is evident that the model achieves high precision and recall rates across all classes, indicating its effectiveness in entity identification. The mAP50 and mAP50-95 scores further highlight the model's robustness and accuracy in handling a wide range of objects in real-world scenarios. This performance summary serves as a critical evaluation of the YOLO model's capabilities in our specific application.

Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100% 22/22 [00:31<00:00, 1.44s/it]
all	347	5594	0.99	0.978	0.992	0.862
Bicycle	347	410	0.969	0.949	0.985	0.578
Car	347	3632	0.997	0.989	0.995	0.936
Bus	347	323	0.998	0.994	0.995	0.987
Lorry	347	1229	0.995	0.982	0.995	0.946

Fig 5.1. Model Performance

5.2 Test Result and Evaluation Metrics

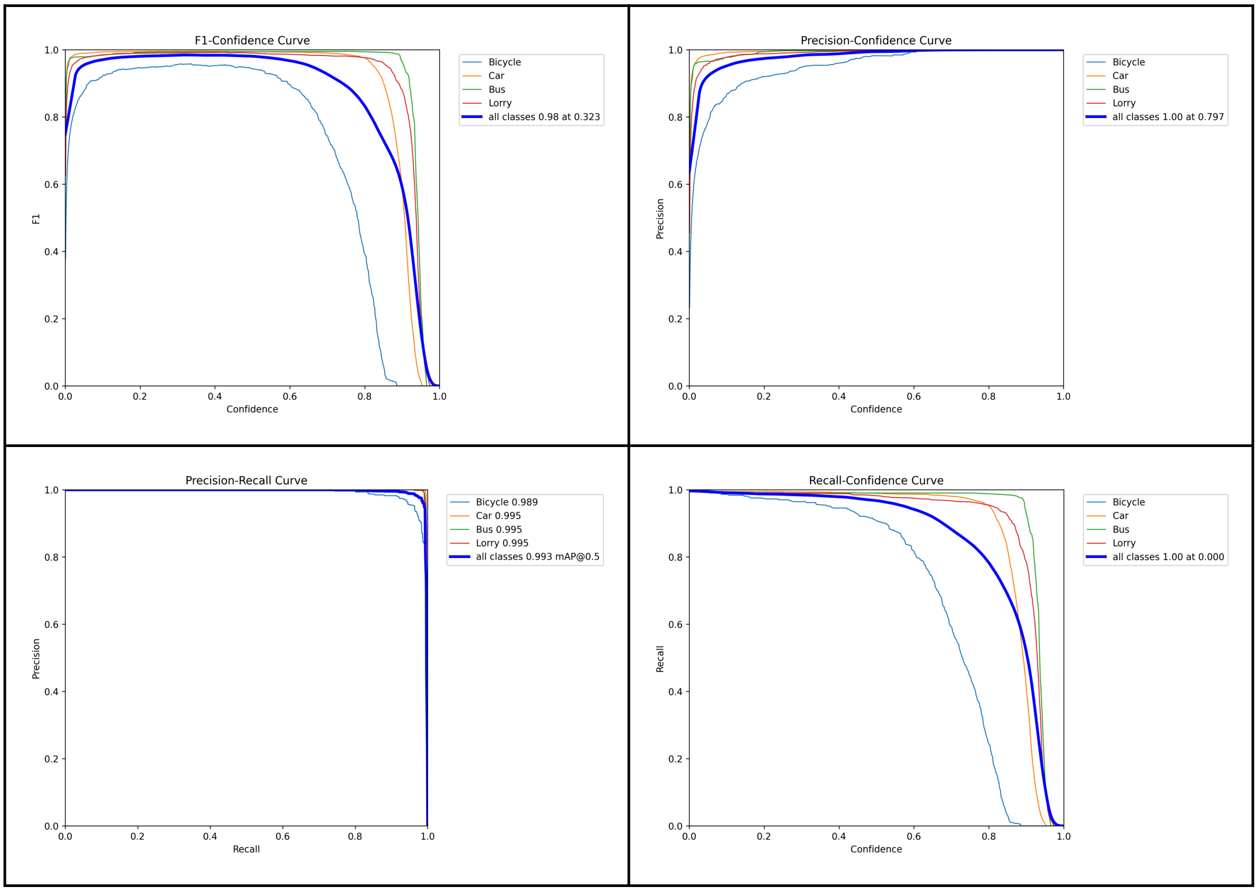


Fig 5.2. Test Result

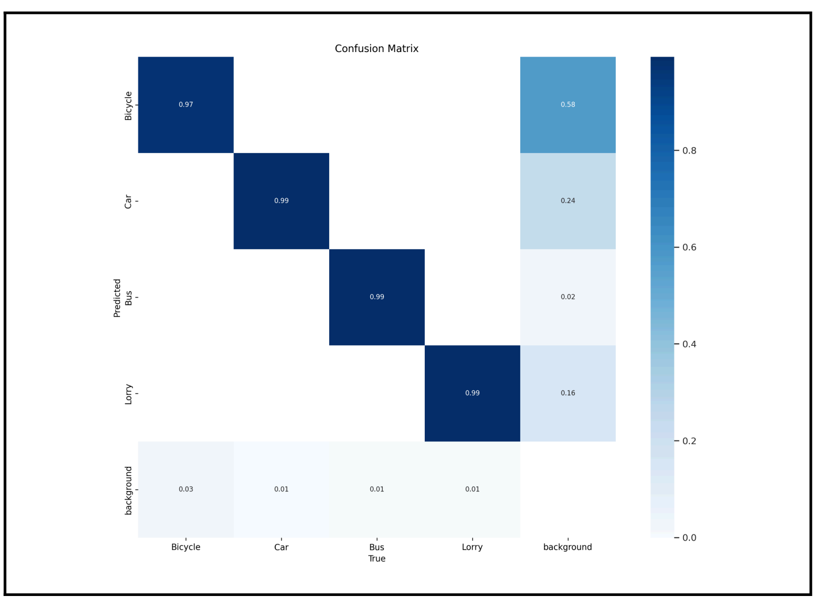


Fig 5.3. Confusion Matrix

5.3 Model Output

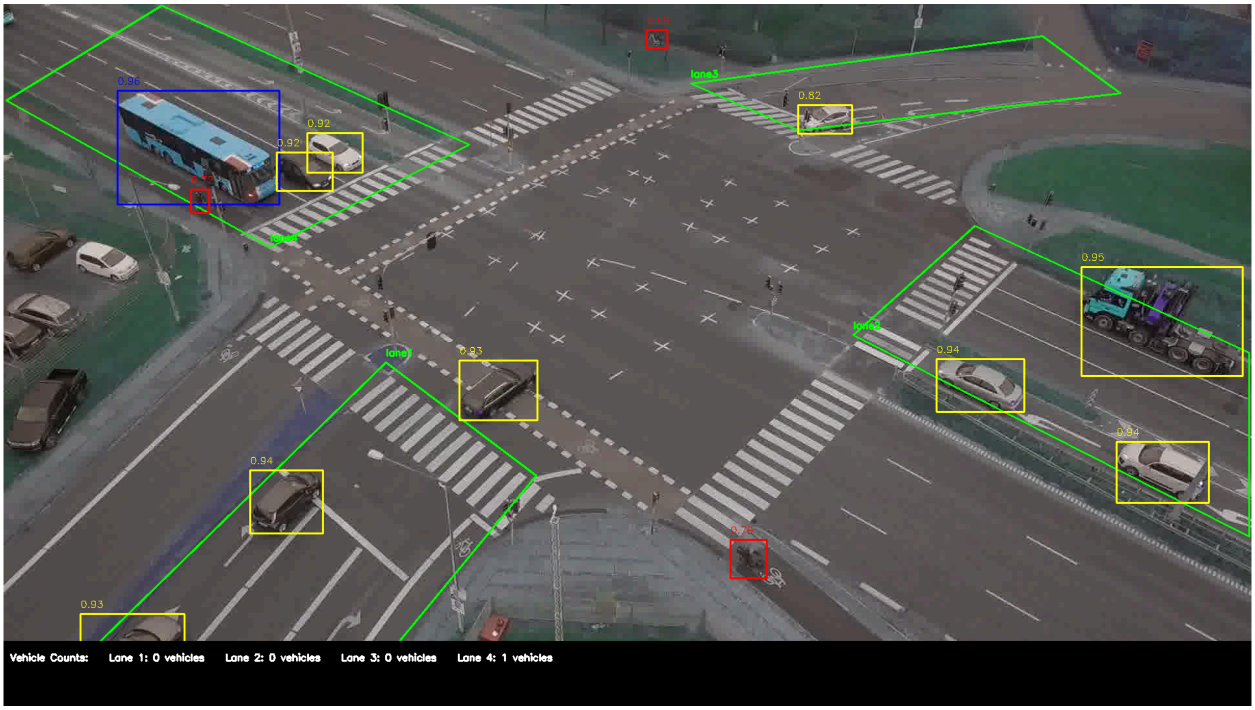


Fig 5.4. Model Output with Vehicle Count per Lane

CHAPTER 6
CONCLUSION AND SCOPE OF FUTURE WORK

6.1 Conclusion

In this study, we presented a comprehensive analysis of the YOLO (You Only Look Once) object detection model's performance for real-time vehicle detection in traffic surveillance scenarios. The model was trained and evaluated on a rich and diverse dataset, capturing various vehicle types, lighting conditions, and traffic scenarios. The experimental findings showcased the efficacy of the YOLO model in accurately detecting and classifying vehicles, including bicycles, cars, buses, and lorries. The model achieved high precision and recall rates across all classes, showcasing its robustness in handling complex urban traffic environments.

Furthermore, the analysis of metrics such as mean Average Precision (mAP) at different IoU thresholds and the Confusion Matrix provided valuable insights into the model's behavior and its ability to handle multi-class object detection tasks. The YOLO model's high performance and efficiency make it a promising candidate for various applications in urban traffic management, intelligent transportation systems, and adaptive traffic signal control.

In conclusion, our research highlights the potential of deep learning-based object detection techniques, particularly YOLO, in enhancing urban traffic monitoring and management. Future work could focus on further fine-tuning the model and integrating it into real-world traffic control systems to achieve safer and more efficient urban mobility.

6.2 Scope for future work

While this study has provided valuable insights into the application of the YOLO object detection model in traffic surveillance and management, there exist numerous prospects for additional investigation and advancement within this field:

- **Real-Time Deployment:** One of the immediate areas for further work is the real-time deployment of the YOLO model in traffic signal control systems. Integrating the model into existing traffic management infrastructure and assessing its real-time performance under dynamic traffic conditions would be a crucial step.
- **Traffic Flow Optimization:** Future research could focus on leveraging the object detection capabilities of YOLO to optimize traffic flow and reduce congestion.

Implementing adaptive traffic signal control algorithms that respond in real-time to detected traffic patterns could significantly improve urban mobility.

- **Multi-Modal Traffic Monitoring:** Expanding the scope of object detection to include additional modalities such as pedestrian detection, traffic sign recognition, and even weather conditions could enhance overall traffic monitoring and safety.
- **Anomaly Detection:** Developing anomaly detection algorithms that can identify unusual traffic events, accidents, or security threats could further enhance the utility of the YOLO model in urban surveillance.
- **Scalability and Robustness:** Investigating the scalability of the model to handle larger intersections and assessing its robustness under adverse weather conditions, low light, or occlusions would be critical for practical utility.

REFERENCES

[1] D. Padilla Carrasco, H. A. Rashwan, M. Á. García, and D. Puig, “T-YOLO: Tiny vehicle detection based on YOLO and multi-scale convolutional neural networks,” IEEE Access, vol. 11, pp. 22430–22440, 2023.

[2] N. Zarei, P. Moallem, and M. Shams, “Fast-Yolo-rec: Incorporating Yolo-base detection and recurrent-base prediction networks for fast vehicle detection in consecutive images,” IEEE Access, vol. 10, pp. 120592–120605, 2022.

[3] M. A. Zuraimi and F. H. K. Bin, “Vehicle detection and tracking using YOLO and DeepSORT,” in 2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), IEEE, 2021.

[4] V. Gorodokin, S. Zhankaziev, E. Shepeleva, K. Magdin, and S. Evtyukov, “Optimization of adaptive traffic light control modes based on machine vision,” Transp. Res. Procedia, vol. 57, pp. 241–249, 2021.

[5] Y. Miao, F. Liu, T. Hou, L. Liu, and Y. Liu, “A nighttime vehicle detection method based on YOLO v3,” in 2020 Chinese Automation Congress (CAC), 2020.

[6] S. Chen and W. Lin, 2019 IEEE 3rd advanced information management, communications, electronic and automation control conference (IMCEC). IEEE, 2019.

[7] X. Dong, S. Yan, and C. Duan, “A lightweight vehicles detection network model based on YOLOv5,” Eng. Appl. Artif. Intell., vol. 113, no. 104914, p. 104914, 2022.

[8] H. Song, H. Liang, H. Li, Z. Dai, and X. Yun, “Vision-based vehicle detection and counting system using deep learning in highway scenes,” Eur. Transp. Res. Rev., vol. 11, no. 1, 2019.

[9] O. O. Khalifa, “Vehicle detection for vision-based intelligent transportation systems using convolutional neural network algorithm,” Journal of Advanced Transportation, vol. 2022, pp. 1–11, 2022.

[10] C. Chen, “Edge intelligence empowered vehicle detection and image segmentation for autonomous vehicles,” IEEE Transactions on Intelligent Transportation Systems, 2023.

[11] T. Wu et al., “Multi-agent deep reinforcement learning for urban traffic light control in vehicular networks,” IEEE Trans. Veh. Technol., vol. 69, no. 8, pp. 8243–8256, 2020.

[12] H. Ge, Y. Song, C. Wu, J. Ren, and G. Tan, “Cooperative deep Q-learning with Q-value transfer for multi-intersection signal control,” IEEE Access, vol. 7, pp. 40797–40809, 2019.

[13] X. Wang, L. Ke, Z. Qiao, and X. Chai, “Large-scale traffic signal control using a novel multiagent reinforcement learning,” IEEE Trans. Cybern., vol. 51, no. 1, pp. 174–187, 2021.

[14] X. Liang, X. Du, G. Wang, and Z. Han, “A deep reinforcement learning network for traffic light cycle control,” IEEE Trans. Veh. Technol., vol. 68, no. 2, pp. 1243–1253, 2019.

[15] M. Tajalli, M. Mehrabipour, and A. Hajbabaie, “Network-level coordinated speed optimization and traffic light control for connected and automated vehicles,” IEEE Trans. Intell. Transp. Syst., vol. 22, no. 11, pp. 6748–6759, 2021.

[16] Cai, Chen, Chi Kwong Wong, and Benjamin G. Heydecker. "Adaptive traffic signal control using approximate dynamic programming." Transportation Research Part C: Emerging Technologies 17.5 (2009): 456-474.

[17] C. Priemer and B. Friedrich, "A decentralized adaptive traffic signal control using V2I communication data," 2009 12th International IEEE Conference on Intelligent Transportation Systems, St. Louis, MO, USA, 2009, pp. 1-6, doi: 10.1109/ITSC.2009.5309870.

[18] K. Pandit, D. Ghosal, H. M. Zhang and C. -N. Chuah, "Adaptive Traffic Signal Control With Vehicular Ad hoc Networks," in IEEE Transactions on Vehicular Technology, vol. 62, no. 4, pp. 1459-1471, May 2013, doi: 10.1109/TVT.2013.2241460.

[19] McKenney, Dave, and Tony White. "Distributed and adaptive traffic signal control within a realistic traffic simulation." Engineering Applications of Artificial Intelligence 26.1 (2013): 574-583.

[20] Smith, Stephen, et al. "Smart urban signal networks: Initial application of the surtrac adaptive traffic signal control system." Proceedings of the International Conference on Automated Planning and Scheduling. Vol. 23. 2013.

[21] K. Liu and G. Mattyus, "Fast Multiclass Vehicle Detection on Aerial Images," in IEEE Geoscience and Remote Sensing Letters, vol. 12, no. 9, pp. 1938-1942, Sept. 2015, doi: 10.1109/LGRS.2015.2439517.

[22] L. W. Sommer, T. Schuchert and J. Beyerer, "Fast Deep Vehicle Detection in Aerial Images," 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, 2017, pp. 311-319, doi: 10.1109/WACV.2017.41.

[23] J. Gleason, A. V. Nefian, X. Bouyssounousse, T. Fong and G. Bebis, "Vehicle detection from aerial imagery," 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 2011, pp. 2065-2070, doi: 10.1109/ICRA.2011.5979853.

[24] M. Y. Yang, W. Liao, X. Li and B. Rosenhahn, "Deep Learning for Vehicle Detection in Aerial Images," 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 2018, pp. 3079-3083, doi: 10.1109/ICIP.2018.8451454.

[25] B. A. Alpatov, P. V. Babayan and M. D. Ershov, "Vehicle detection and counting system for real-time traffic surveillance," 2018 7th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, 2018, pp. 1-4, doi: 10.1109/MECO.2018.8406017.

[26] Z. Rahman, A. M. Ami and M. A. Ullah, "A Real-Time Wrong-Way Vehicle Detection Based on YOLO and Centroid Tracking," 2020 IEEE Region 10 Symposium (TENSYMP), Dhaka, Bangladesh, 2020, pp. 916-920, doi: 10.1109/TENSYMP50017.2020.9230463.

[27] Z. Zheng et al., "A Novel Vehicle Detection Method With High Resolution Highway Aerial Image," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 6, no. 6, pp. 2338-2343, Dec. 2013, doi: 10.1109/JSTARS.2013.2266131.

[28] Razakarivony, Sebastien, and Frederic Jurie. "Vehicle detection in aerial imagery: A small target detection benchmark." Journal of Visual Communication and Image Representation 34 (2016): 187-203.

[29] Chabchoub, Abdelkader, et al. "Intelligent traffic light controller using fuzzy logic and image processing." International Journal of Advanced Computer Science and Applications 12.4 (2021): 396-399.

[30] M. B. Jensen, A. Mogelmose, and T. B. Moeslund, “Presenting the multi-view traffic intersection dataset (MTID): A detailed traffic-surveillance dataset,” in 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), 2020.

[31] Vedansh Bhardwaj, Yaswanth Rasamsetti, Vipina Valsan, Image Processing Based Smart Traffic Control System for Smart City, 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), IEEE.

[32] Sanjai Siddharthan, M., S. Aravind, and S. Sountharrajan. "Real-Time Road Hazard Classification Using Object Detection with Deep Learning." International Conference on IoT Based Control Networks and Intelligent Systems. Singapore: Springer Nature Singapore, 2023.

[33] Raj, V. Sowbaranic, et al. "Smart Traffic Control for Emergency Vehicles Prioritization using Video and Audio Processing." 2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS). IEEE, 2022.

[34] Tarachandy, Sahul Mohan, and J. Aravinth. "Enhanced local features using ridgelet filters for traffic sign detection and recognition." 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC). IEEE, 2021.