Birzeit University
Faculty of Engineering & Technology
Department of Electrical & Computer Engineering

Traffic Light System

Prepared By:
Abdelrhman Abed 1193191
Ody Shbayeh 1201462
Nafe AbuBaker 1200047
Loai saad 1191582

Supervised By:
Dr. Ibrahim Nemer

ENCS5300, Section 11

A Graduation Project submitted to the Department of Electrical and Computer Engineering in partial fulfillment of the requirements for the degree of B.Sc. in Computer Engineering

Birzeit
JULY, 2025

Abstract

Traffic congestion and red light running at busy urban intersections threaten mobility, safety, and environmental sustainability. This work presents an end to end computer vision framework that unifies real time queue aware signal optimisation with automated red light violation detection in a single, scalable platform. Overhead drone footage of Ramallah's "Five Stars" intersection is analyzed with YOLOv8 and SORT to track vehicles inside four polygonal regions of interest; each second, the system records a JSON snapshot of the current phase, the recommended next phase (RED $\rightarrow$ YELLOW $\rightarrow$ GREEN), and the instantaneous queue length per approach. A rule based engine processes these snapshots, applies a congestion minimization algorithm using configurable thresholds, and issues an optimized next state decision for exactly one approach while holding the others constant. Correctness is validated in a lightweight agent based simulator that replays the JSON stream: vehicle arrivals follow rates calibrated from field video, and departures occur at a representative service rate (e.g., 2 cars/s) when the light is green; compared with a conventional fixed-timing cycle, this system raises average throughput by about 0.25 cars/s. For enforcement, a street level pipeline overlays a virtual signal on frames and again employs YOLOv8, SORT, and a custom license-plate detector to track vehicles, flag any that cross the stop-line during red, extract the clearest evidence frame, run OCR on the plate, and archive violation clips plus metadata in MongoDB triggering immediate SMS/e-mail alerts to authorities or vehicle owners. Both subsystems feed a React/FastAPI dashboard that visualizes live queue lengths, historical performance, and timestamped violation evidence, giving engineers and enforcement officers actionable real-time insight. Spanning video acquisition, detection, decision logic, simulation, violation logging, and web visualization, the modular architecture offers a practical blueprint for municipalities seeking to reduce congestion and deter red light violations with minimal human oversight.

المستخلص

تُشكّل الازدحامات المروريّة وتجاوز إشارات المرور الحمراء عند التقاطعات مشكلة كبيرة حيث تعرقل انسيابية الحركة المرورية و تؤدي الى تجاوزات غير قانونية و تعرض حياة المواطنين للخطر . يعرض هذا المشروع إطارًا متكاملاً قائمًا على الرؤية الحاسوبية يوطّد بين تحسين توقيت الإشارات المرورية ورصد الانتهاكات أثناء تجاوز الإشارة الحمراء في منصة واحدة قابلة للتوسع. تُحلّل لقطات فيديو جوية مُلتقطة لمفترق "خمس نجوم" في مدينة رام الله بواسطة نماذج YOLOv8 وخوارزمية SORT لتعقب المركبات داخل أربعة تقاطعات من المفترق ، حيث يُسجَّل كل فريم ملف JSON يحتوي على الحالة الحالية للإشارة، والحالة الموصى بها التالية (أحمر→أصفر→أخضر)، وطول الصف الانتظاري الفوري لكل اتجاه. بحيث يتم تخزين جميع هذه المعلومات في قاعدة بيانات MongoDB, يعالج النظام هذه اللقطات، ويطبّق خوارزمية تقليل الازدحام عن طريق تفضيل بدء الإشارة للمفترق الأكثر ازدحاما وتقليل تفضيل تشغيل اشارة المفترق الأقل ازدحاما حتى يتم عرض خطة تشغيل جديدة للمفترق بترتيب جديد بناء على حالات الازدحام للمفترقات ويتم حفظ هذه الخطة وترتيباتها في ملف JSON, يتم قراءة ملف JSON من نظام محاكاة يقوم بعرض المفترق والتفضيلات المتواجدة ويعرض التفضيلات, حيث تتبع معدّلات وصول المركبات الإحصائيات المُستمدة من الفيديو الميداني، وتتم عمليات المغادرة بمعدل خدمة نموذجي (مثل 2 مركبة/ثانية) عند الإشارة الخضراء؛ مقارنةً بالدورة الثابتة التقليدية، يزيد هذا النظام متوسط عدد المركبات المارة بمقدار حوالي 0.25 مركبة/ثانية.

في جانب المراقبة وتنفيذ نظام رصد تجاوز الإشارة الحمراء، يعتمد الإطار على أنبوب معالجة مرئي يغطّي الإشارة الحقيقية بإشارة افتراضية ضمن الإطار نفسه. يستخدم النظام مجدّدًا YOLOv8 وخوارزمية SORT لاكتشاف لوحات المركبات و لتعقب السيارات وتحديد تلك التي تعبر خط التوقف أثناء الإشارة الحمراء. ثم يستخلص الإطار الأنقى كدليل، ويطبق تقنيةOCR على لوحة المركبة لقراءة الأرقام والحروف، ثم يؤرشف مقاطع فيديو الانتهاك وصور المركبة والبيانات الوصفية (رقم لوحة المركبة ، مالك المركبة ، وقت و تاريخ الانتهاك) في قاعدة بيانات MongoDB ، مع إرسال تنبيه فوري برسالة نصية إلى صاحب المركبة.

تُغذّي هذه المكوّنات لوحة تحكّم تفاعلية مبنية بـ ReactوFastAPI تعرض عدد المركبات على التقاطعات، وأداء النظام في مختلف الفترات، إضافةً إلى صور وفيديوهات للمركبات التي قامت بقطع الإشارة الحمراء . وبذلك يحصل المهندسون وضباط التنفيذ على رؤية عملية وفورية. انطلاقًا من التقاط الفيديو، مرورًا بعمليات الكشف واتخاذ القرار والمحاكاة وتسجيل الانتهاكات، وصولًا إلى العرض على الويب، يقدم هذا التصميم المعياري دليل عمل متكامل للجهات المعنية التي تسعى إلى تقليل الازدحام وردع مخالفات تجاوز الإشارة الحمراء بأقل جهد بشري وبالاعتماد الكامل على النظام المدعوم بالذكاء الاصطناعي.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction and Motivation

Contents

1.1 Introduction

Palestine has experienced a sharp increase in vehicles, near half a million, leading to severe traffic congestion, especially at intersections. Traffic levels reached unprecedented heights in 2022, particularly in major cities, significantly affecting daily life **[1]**. The Ministry of Transport and Communications attempted to address these issues with synchronized traffic lights, but they have not yet found the optimal solution to accommodate the growing number of vehicles, resulting in continued delays. Consequently, some drivers resorted to illegal behaviors such as running red lights and speeding, contributing to an increase in accidents from about 14,105 in 2021 to over 15,200 in 2022 **[2]**. By 2024, while field police have been deployed to manage traffic and curb violations, the increasing number of vehicles and signals makes it challenging to monitor all intersections. With technological advancements, a smart traffic signal system could better adapt to the flow of cars at intersections, improving traffic management, and reducing accidents.

One significant step in this direction is the development of Automated Traffic Law Enforcement Systems **[3]**. These systems utilize Deep Learning (DL), computer vision, and Internet of Things (IoT) technologies to automate the detection of traffic violations such as red light running. Through real-time video analysis and Automatic Number Plate Recognition (ANPR), they operate with more accuracy than 90%, reducing the need for manual enforcement and improving compliance under various environmental conditions.

With the integration of advanced traffic signal violation detection frameworks using computer vision **[4]**, signals can be dynamically adjusted in real time to optimize vehicle flow and improve driver safety. These systems employ YOLOv8 (You Only Look Once) and Simple Online and Real Time Tracking (SORT) algorithms to detect and track vehicles, providing precise identification of violations from live camera feeds.

Traditionally, traffic management has relied on fixed timing signals pre programmed intervals that do not reflect actual vehicle flow. Although cost effective and simple, such systems often lead to off peak inefficiencies. Manual traffic policing at busy intersections can become overwhelming as vehicle counts increase, lacking the flexibility to adapt to real-time dynamics. End to end object detectors like YOLO holistically analyze each video frame, identifying light states, vehicles, and infractions efficiently. Using YOLO's real-time capabilities **[5]**, congestion monitoring and violation detection are streamlined, improving intersection management and road safety.

1.2 Motivation

The motivation behind this project was inspired by the urgent need to address the challenges of managing traffic in Palestine's rapidly urbanizing environment. With the sharp increase in the number of vehicles and the limited effectiveness of traditional traffic management systems, such as fixed timing signals and manual enforcement, it is evident that innovative solutions are necessary.

This project aims to leverage contemporary technologies to provide a scalable and efficient approach to managing traffic violations and improving road safety, and reducing time waited for the vehicle. The integration of advanced tools like the YOLO algorithm and real time monitoring systems offers the potential to revolutionize traffic management. Algorithms such as YOLO enable the identification of traffic violations and driver behavior at intersections, empowering authorities to respond proactively. By processing live video feeds from surveillance cameras, these systems can detect violations such as red light running, ensuring that traffic laws are enforced effectively and consistently.

The increasing complexity of urban traffic systems demands solutions that go beyond traditional approaches. Limited human resources and the growing number of intersections make it challenging to monitor traffic manually. This project shows a future where automated systems powered by Machine Learning (ML), IoT, and advanced computer vision algorithms take over these responsibilities. These systems not only detect violations but also optimize traffic signals dynamically, adapting to real-time conditions to reduce congestion and improve overall traffic flow. Moreover, this project highlights the broader vision of integrating modern technologies into urban infrastructure. It envisions a useful blend of Artificial Intelligence (AI), and real time data analytics to create a connected ecosystem that enhances safety and efficiency. By acting as both a guardian of traffic discipline and a promoter of road safety education, the proposed solution aims to build trust and cooperation among drivers, and authorities.

This project serves as a great beginning toward a smarter, safer, and more efficient traffic management system, addressing the challenges of modern urban life and laying the foundation for sustainable and Intelligent Transportation Systems (ITS).

1.3 Objectives

The primary objective of this project is to develop an intelligent traffic management and violation detection system tailored to the unique challenges of Palestine's urban traffic. By using advanced technologies such as computer vision, ML, and real time data processing, the project aims to create a scalable, efficient, and reliable solution for modern traffic management. The objectives of the project are as follows:

1. **Reviewing existing approaches:**

   Analyze and evaluate traditional and automated approaches to traffic management and violation detection, identifying their strengths, limitations, and areas for improvement.

2. **Studying relevant technologies:**

   Explore the foundational concepts and methodologies of ML, computer vision, focusing on their applications in traffic systems.

3. **Dataset preparation and analysis:**

   Investigate and preprocess real world traffic datasets, extracting meaningful insights to inform the development and testing of the proposed system.

4. **Proposing an automated detection system:**

   Design and implement a system that uses cutting edge technologies like YOLO for object detection, hopefully to use IoT for connectivity, and real time analytics to identify and respond to traffic violations such as red light violation and traffic management.

5. **Validation and testing:**

   Conduct extensive simulation and testing under various conditions to verify the system's accuracy, reliability, and scalability in detecting traffic violations and managing congestion.

6. **Generating JSON output:**

   Develop a mechanism to export traffic violation data, including details such as time, location, type of violation, and vehicle information, into a structured JSON file saved in a secure database for use by authorities and analysts.

7. **Promoting safety and efficiency:**

   Ensure the proposed system not only enforces traffic laws but also optimizes traffic signal timing dynamically, improving traffic flow and reducing accidents.

1.4 Scope

This graduation project delivers a proof of concept software framework from raw video ingest to web visualization for intelligent traffic management at the "5 Stars Intersection" in Ramallah. The scope is deliberately restricted to the items below; any requirement not listed is considered out of scope.

In-scope deliverables

- **Data acquisition & annotation**
  30 minutes overhead drone video (jammed & light traffic) street level clip for red light testing Manual bounding boxes for vehicles and virtual traffic light panels
- **AI pipelines**
  - *Traffic flow optimization*: YOLOv8 + SORT for per approach vehicle counts; rule based controller that issues a single next state recommendation each second.
  - *Red light violation detection*: default YOLOv8 for vehicles, custom YOLOv8 for licence plates, SORT tracking, best frame selection, and Optical Character Recognition (OCR) extraction .
- **Simulation-based validation**
  Lightweight Pygame agent model that replays JSON recommendations to quantify throughput gains over a fixed 100 s cycle.
- **Web dashboard**
  React + Tailwind frontend, FastAPI backend, MongoDB datastore exposing: traffic metrics, best-frame cards, violation clips, OCR plates.
- **Evaluation metrics**
  mAP for detection accuracy, cars s throughput, peak queue length, OCR precision.
- **Documentation**
  Full LaTeX report, commented source code, and deployment guide for local execution on GPU equipped workstations.

Out-of-scope items

- Hardware integration with physical traffic signal controllers or police SMS gateways beyond the mock API .
- Multi-intersection coordination, city wide optimization, or edge device deployment.
- Adverse weather robustness, night time imaging, or multi lane turning behavior.
- Ticket-issuing workflows (payment portals, legal databases).
- Alternative sensing modalities (LiDAR, radar, inductive loops) and additional traffic violations (speeding, illegal turns).
- Detailed economic / environmental impact analysis.

By clearly delimiting the project this way, all subsequent chapters focus on designing, implementing, and validating the software only solution described above, without over reaching into deployment specific engineering tasks.

Chapter 2

Related Work

Contents

2.1 Introduction

This chapter of the project explores advancements in traffic violation detection and management techniques, addressing the growing inadequacy of traditional traffic management systems such as fixed timing signals and manual enforcement **[1, 2]**. The rapid increase in vehicle numbers and the dynamic nature of urban traffic conditions have highlighted the need for ITS as scalable alternatives. Leveraging AI, computer vision, and DL, ITS technologies optimize traffic flow, enforce compliance, and enhance road safety **[4]**. Algorithms like YOLO are central to these advancements, enabling efficient object detection and real time monitoring of congestion, violations, and traffic flow at intersections **[3, 5]**.

Various methods leveraging computer vision, DL, and AI have been developed to tackle challenges such as real time violation detection, traffic optimization, and vehicle recognition. This chapter provides an analysis of these techniques, examining the approaches used in recent studies and comparing their strengths and limitations. The discussion concludes with a comprehensive table summarizing key aspects of the reviewed works, demonstrating the foundation and motivation for the proposed AI-powered system.

2.2 Techniques

The project has explored various techniques for addressing specific traffic challenges. For instance, weight based braking distance models combine YOLO and physics based calculations to enhance red light violation detection by estimating braking distances based on speed and weight **[6]**. Automated traffic law enforcement systems utilize DL and ANPR to detect violations like lane breaches and speeding with over 90 % accuracy **[7]**, though their real time functionality requires high performance hardware. Similarly, big data plat forms analyze structured and unstructured data using the Mean Absolute Percentage Error (MAPE) methodology **[8]**, identifying risky driver behaviors but relying heavily on Hadoop infrastructure, which can limit deployment in resource constrained areas.

Mobile based systems have introduced decentralized approaches to traffic management. For example, systems combining YOLOv5 and MobileNet OCR analyze dash camera footage for detecting red light running and illegal parking, but scalability is limited by mobile processing constraints **[9]**. Hybrid systems , integrating YOLO and Convolutional Recurrent Neural Network (CRNN), achieve 98.06 % accuracy for multi violation detection and license plate recognition, though they depend heavily on high quality video feeds **[10]**. Enhanced traffic signal detection models, such as Traffic Light Detection Model (TLDM) **[11]**, incorporate Mosaic-9 augmentation to improve small target detection, achieving an mAP of 99.4 %, while YOLOv5s combined with AlexNet excellent in detecting traffic lights under low light conditions with a precision of 99.46 % **[12]**.

DL models have also shown promise in addressing red light violations. YOLOv8-based systems detect speeding violations using live Closed Circuit Television (CCTV) footage, face challenges with dataset diversity and recall rates **[13]**. Meanwhile, automated red light violation systems use YOLOv5s to detect vehicles crossing virtual stop lines, achieving 86 % accuracy but focusing narrowly on red light enforcement **[14]**. Adaptive traffic light systems demonstrate the potential of integrating AI for dynamic traffic management at intersections, though practical deployments often require further refinement **[15]**. Emphasize the potential of DL technologies for traffic management while highlighting the need for broader integration into real-world applications **[16]**. Additionally, traffic flow optimization models, integrating YOLO based detection, have demon started a 23 % improvement in lane efficiency, though they primarily focus on optimizing flows rather than addressing violations. Despite these advances, most systems remain limited in scope, addressing either violations or optimization, but rarely integrating both in a unified framework **[17]**.

Despite these features, many systems remain limited to either specific violations or traffic optimization without broader integration. This Project addresses these gaps by proposing a unified AI-powered system that integrates adaptive traffic signal optimization with violation detection. By combining advanced computer vision techniques and real time analytics, the proposed system provides a scalable and effective solution for modern traffic challenges, enhancing safety and reducing congestion.

2.3 Comparison

The reviewed techniques reveal significant advancements in traffic violation detection and management. However, they vary in terms of problem scope, solution type, datasets, evaluation metrics, and limitations. A summary of these techniques, including their key attributes and limitations, is presented in Table **2.1** below is a detailed comparison based on these attributes.

2.3.1 Problem Scope

The problems addressed by these papers range from specific traffic violations like speeding, red light running, and illegal parking to broader challenges such as traffic flow optimization and risky driver behavior identification. Early techniques, such as weight based braking distance calculation, focus narrowly on a specific violation, while later works expand to multi violation detection and traffic flow management.

For instance, the weight-based approach specifically targets red light violations by predicting braking distances. In contrast, systems like "An Efficient System for Detecting Multiple Traffic Violations and Recognizing License Plates" and "YOLO based Intelligent Traffic Light Control" offer solutions for broader challenges, including traffic flow optimization and real-time multi-violation detection **[18]**.

2.3.2 Solution Types

Most solutions rely on DL based object detection models like YOLO, with some using hybrid approaches (e.g. YOLOv5s with AlexNet or CRNN). Others, such as the Big Data Platform, focus on big data processing, while analytical reviews evaluate frameworks rather than propose For example:

- YOLO based solutions dominate real time traffic violation detection and object tracking applications.
- Hybrid approaches, such as the integration of AlexNet with YOLOv5s, enhance detection in specific conditions like low light environments.
- Big Data platforms and ML reviews emphasize scalable data integration but lack real time applicability.

2.3.3 Datasets

The datasets used vary widely in terms of specificity and scale:

- Custom datasets are prevalent in methods addressing unique challenges, such as weight based braking distance prediction and traffic signal violations.
- Public datasets like Microsoft Common Objects in Context (COCO) provide a standardized benchmark for general object detection tasks.
- Some approaches rely on unstructured data (e.g. video and images) collected through big data platforms.

While custom datasets enable finetuning for specific scenarios, they may limit the generalization of the model. Conversely, public datasets provide broader applicability but may not address nuanced traffic behaviors effectively.

2.3.4 Evaluation Metrics

Evaluation metrics are crucial for assessing the effectiveness of traffic violation detection systems. Metrics such as mAP , detection accuracy, precision, recall, and F1 score are commonly used. These metrics not only highlight the performance of the systems but also indicate areas for improvement.

1. **mAP:** mAP is a standard metric for evaluating object detection models. It measures the precision of predicted bounding boxes over different levels of recall, providing an aggregate measure of how well the system identifies objects. High mAP values, such as the 99.4 % reported by TLDM (An Enhanced Traffic Light Detection Model), indicate robust performance in detecting objects like traffic lights in various scenarios **[9]**.

2. **TLDM:**

   TLDM is an advanced system designed to detect traffic lights, even small or partially obscured ones, in complex urban settings. By utilizing YOLOv5 with enhancements like Mosaic-9 and Squeeze and Excitation (SE) attention mechanisms, it achieved an impressive mAP of 99.4 %, highlighting its effectiveness in real time traffic light detection.

3. **OCR:**

   OCR refers to the technology used to recognize and extract text from images or video, such as license plates. Systems like "An Efficient System for Detecting Multiple Traffic Violations and Recognizing License Plates" achieved high OCR accuracy rates, with 98.22 % for license plate recognition. Such precision underscores the reliability of OCR in traffic law enforcement applications **[18]**.

However, metrics can suffer under limited dataset diversity, reducing recall in varied lighting or congestion.

2.3.5 Limitations

While these techniques showcase promising results, they have specific limitations:

- **Hardware Dependence:**

  Any systems, especially those using YOLO and ANPR, require high performance hardware for real time processing.

- **Dataset Diversity:**

  Limited datasets can affect the system's ability to generalize across different environments, leading to lower recall or precision rates.

- **Scope of Application:**

  Several systems focus on a narrow range of violations or traffic scenarios, reducing their overall impact on traffic management.

- **Environmental Challenges:**

  Techniques addressing varying weather and lighting conditions remain limited in accuracy during extreme conditions.

Table 2.1: Comparison of Traffic Violation Detection and Management Techniques

Paper	**Problem Addressed**	**Solution Type**	**Dataset**	**Key Metrics**	**Limitations**
[6]	Red-light violation considering vehicle weight	YOLO + Physics Model	Custom vehicle image set	Prediction error: 4.79%	Assumes weight categories are accurate
[7]	Real-time traffic violation enforcement	DL, ANPR	Trained on various conditions	Violation detection: 90%, ANPR: 96%	Requires high-performance hardware
[8]	Identifying risky driver behaviors	Big Data + MAPE	Unstructured data	Processing efficiency: 75%	Requires Hadoop infrastructure
[9]	Analytical review of ML-based detection methods	Analytical review	High-resolution video/image data	Accuracy varies across methods	No implementation evaluation
[10]	Multi-violation detection via mobile devices	YOLOv5 + MobileNet OCR	Public footage (YouTube, dashcams)	Detection rate: 100%	Limited by mobile processing power
[11]	Vehicle detection under varying environments	Appearance & motion-based review	Lighting /weather variation datasets	Accuracy varies across conditions	Lack of cohesive system integration
[12]	Multi-violation detection and license recognition	YOLO + CRNN + OCR	Real-time video; custom model	Violation detection: 98.06%, OCR: 98.22%	Dependent on real-time video quality
[13]	Speeding violation detection	YOLOv8	Microsoft COCO	Bounding box accuracy	Limited dataset diversity
[14]	Monitoring and detecting signal violations	YOLOv8 + SORT	Custom labeled dataset	Precision, recall, F1 score	Reduced accuracy in complex scenarios
[15]	Detecting small traffic light targets	YOLOv5 + Mosaic-9 + SE attention	Mosaic-9-enhanced dataset	mAP: 99.4%, 74 fps	Limited to detection only
[16]	Low-light traffic light detection	YOLOv5s + AlexNet	ZeroDCE-enhanced low-light images	Precision: 99.46%	Sensitive to enhancement quality
[17]	Red-light violation detection	YOLOv5s + OCR	Pre-trained YOLOv5s dataset	mAP: 0.677, Violation detection: 86%	Accuracy depends on OCR quality

Chapter 3

Background

Contents

3.1 Overview of Traffic Light Systems and Congestion Problems

Traffic light systems, integral to urban traffic management, were originally designed for manual or fixed timing operations to regulate traffic flow. Over time, these systems evolved to become automated; however, in many parts of the world, including Palestine, their reliance on fixed intervals remains largely unchanged. Traditional systems, while effective under predictable conditions, have struggled with the growing complexity of modern traffic and the increasing density of vehicles. Studies highlight that efficient traffic management is critical to reducing congestion, yet this challenge has placed immense pressure on infrastructure in developing regions like Palestine. The surge in urban populations and vehicle numbers has led to daily delays, higher emissions, and mounting frustration among road users, underscoring the need for more adaptive solutions. Recent advancements in ITS offer innovative approaches to traffic management, combining technological tools with sustainable mobility solutions. As systematic reviews on urban congestion have shown, datasets and algorithms are essential for predicting, detecting, and addressing traffic flow issues. Locally sourced traffic datasets from Palestinian intersections provide a unique opportunity to apply such innovations, creating tailored solutions to meet the region's specific needs while addressing the broader challenges of urban transportation systems **[18]** . Figure **3.1** shows some causes for the congestion problem.

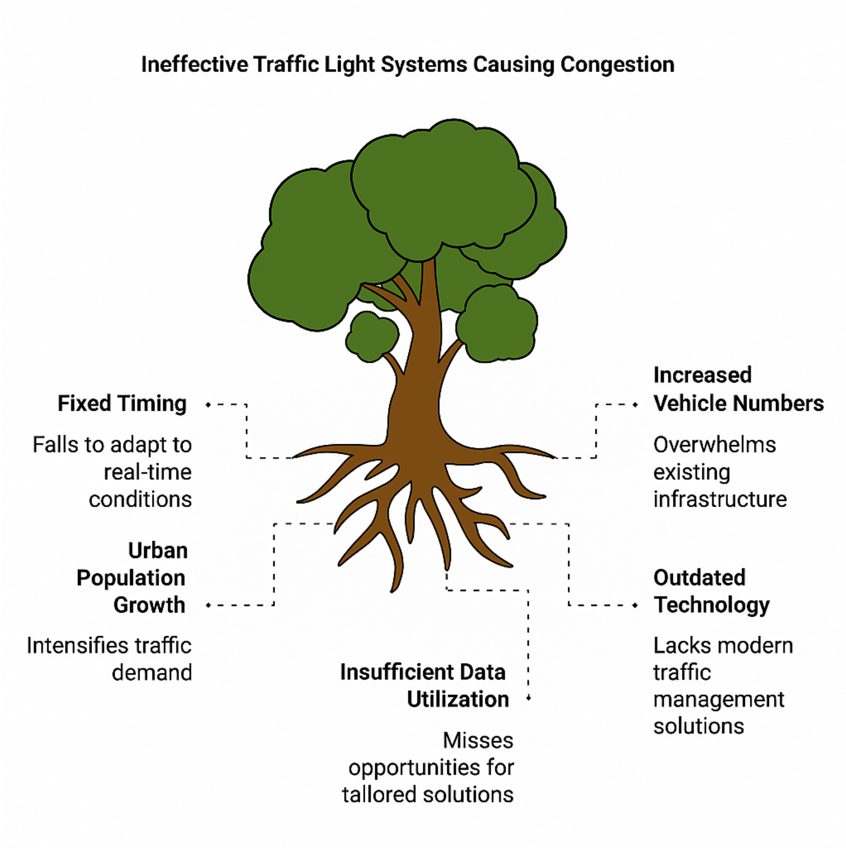

Figure 3.1: Congestion Cause.

3.2 Limitations of Traditional Traffic Light Systems

One of the primary inefficiencies in traditional traffic light systems is their reliance on fixed timing intervals, which fail to adapt to real time traffic conditions. For example, green lights may persist for lanes with minimal traffic while vehicles accumulate in other directions. This mismatch not only causes unnecessary delays but also contributes to higher fuel consumption and carbon emissions. The scalability of manual traffic management is another limitation, often requiring field officers to monitor intersections. However, these methods are inherently constrained by the availability of human resources and their inability to consistently oversee multiple intersections. Such approaches are unsustainable as urban areas continue to grow. Figure **3.2** shows a list of limitations for the traditional systems.

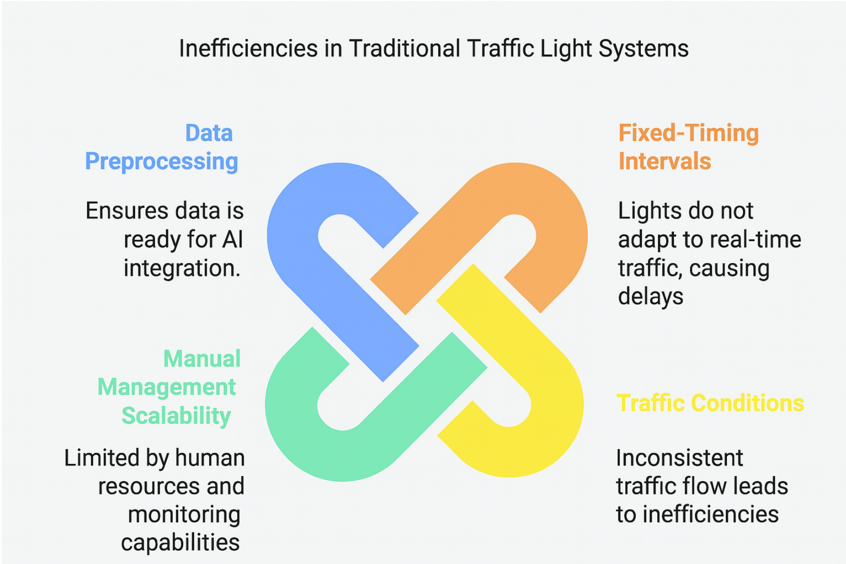

Figure 3.2: Traditional Traffic Light Limitations.

3.3 Introduction to AI-Powered Traffic Light Systems

AI has emerged as a transformative force in modern technology, offering solutions that far surpass traditional approaches in adaptability and efficiency. In the context of traffic management, AI-powered systems leverage advanced algorithms to dynamically analyze data and make real time decisions. This adaptability enables traffic lights to respond to current traffic patterns rather than relying on predefined intervals, significantly improving flow and reducing congestion. ML and DL, as depicted in Figure **3.3**, are considered subsets of AI that have witnessed exponential growth in both research and real world deployment in recent years **[19].**

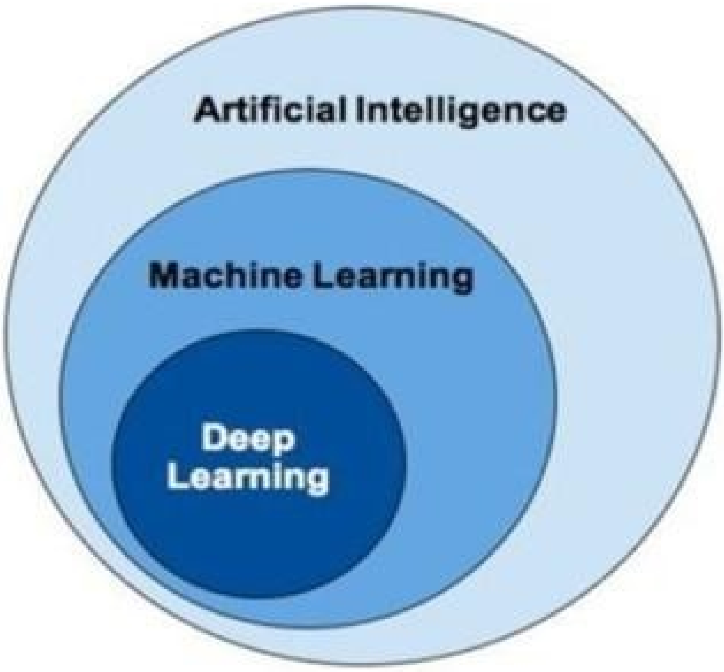

Figure 3.3: AI, ML, and DL Relationship.

3.3.1 Machine Learning

ML involves statistical models used for classifications and predictions based on provided data **[20]**. It is a branch of AI that develops prediction algorithms by identifying patterns within large datasets, without being explicitly programmed for specific tasks **[21]**. ML models are categorized into supervised, unsupervised, and reinforcement learning, with further subdivisions based on various learning approaches, as illustrated in Figure **3.4** **[22]**.

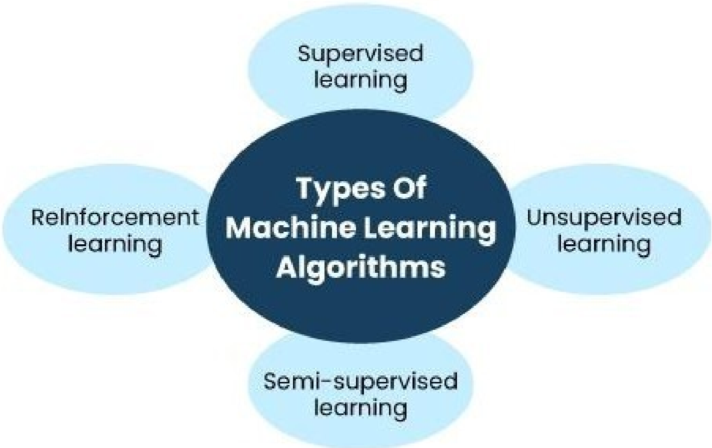

Figure 3.4: Types of ML Algorithms

3.3.2 Deep Learning

DL emerged about a decade ago as a powerful ML technique, achieving notable performance across various fields **[23]**. Its core idea involves using Deep Neural Networks (DNN) to automatically learn complex features from data with minimal external input **[24]**. DL, a significant AI paradigm, has garnered substantial academic interest due to its superior potential over traditional methods. It is more efficient, supervised, time intensive, and cost effective compared to ML techniques. DL not only represents a specific approach to learning but also adapts to diverse methodologies and structures, addressing a wide range of complex problems. It learns illustrative and differential properties in a flexible manner. The procedure of ML and DL is shown in Figure **3.5** **[25]**.

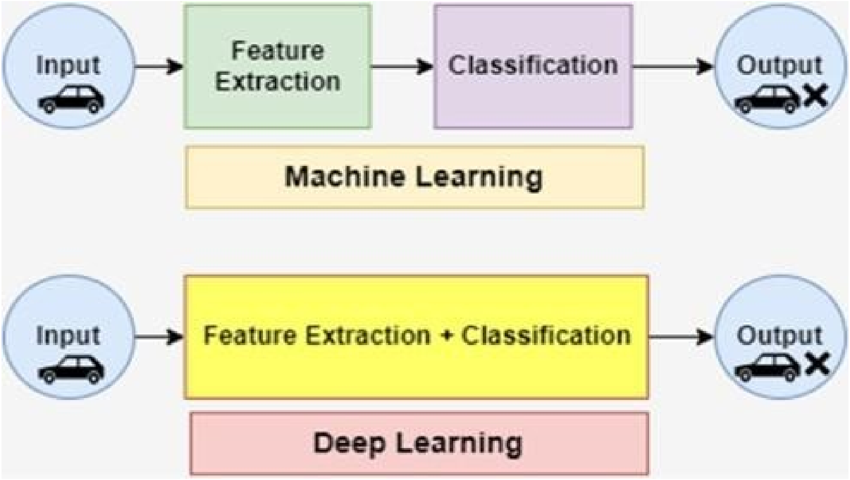

Figure 3.5: Machine Learning vs. Deep Learning

3.3.3 Impact of AI on Traffic Management

Prior studies have highlighted the importance of AI in urban traffic management, showing how real time analytics can enhance safety, reduce accidents, and enforce traffic laws consistently. By processing live video streams and other sensory data, AI systems can detect violations such as red light running, illegal parking, and speeding. Moreover, their ability to adapt to changing traffic conditions whether caused by peak hours, accidents, or environmental factors ensures a more efficient and responsive traffic control system. The integration of AI into traffic light systems represents a significant step forward in ITS, combining technological innovation with practical applications to address the growing challenges of urban mobility.

3.4 AI-Based Solutions for Traffic Management

The integration of advanced AI methods such as computer vision, IoT, and ML has demonstrated significant potential in addressing traffic management challenges. Several AI based solutions have been proposed and implemented in the field of traffic monitoring, leveraging real time data to improve efficiency and reduce violations. Early applications include object detection algorithms like Faster Region based Convolutional Neural Network (R-CNN) and Single Shot MultiBox Detector (SSD), which paved the way for more efficient models such as YOLO. These models have gained prominence due to their ability to process live video feeds in real time, making them particularly suitable for high demand scenarios like traffic management.

3.4.1 Earlier YOLO Models and their Contributions

YOLOv4 and YOLOv5 have been widely used in traffic management systems for tasks such as vehicle detection, traffic signal recognition, and violation monitoring.

- **YOLOv4** introduced innovations like Cross Stage Partial (CSP) networks and improved data augmentation strategies, significantly boosting detection accuracy and training efficiency. Its application in traffic monitoring has been reported to achieve high precision in detecting vehicles and recognizing traffic patterns in diverse conditions.

- **YOLOv5** further enhanced detection performance by focusing on lightweight architecture and faster inference speeds, making it highly effective for real time deployment on resource constrained devices. Its scalability and ease of training have made it a popular choice for traffic management studies. Figure **3.6** **[26]** , shows the results of using both versions of YOLO.

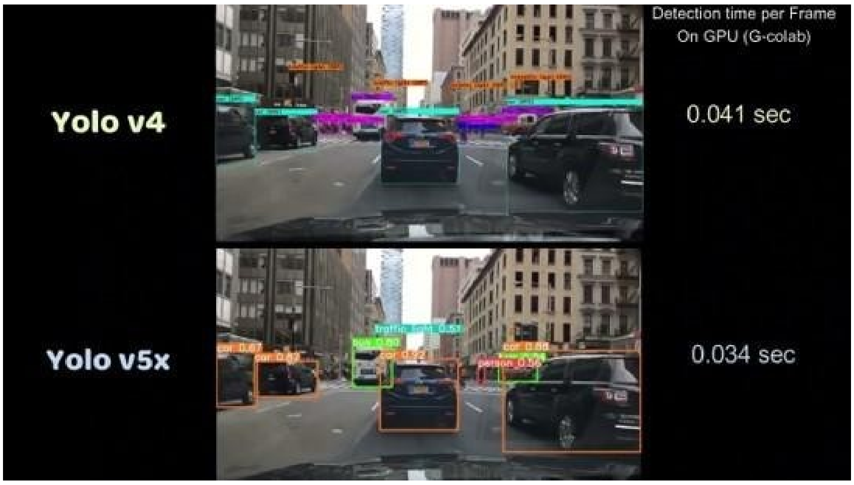

Figure 3.6: YOLOv4 vs YOLOv5.

3.4.2 YOLO Architecture and Advancements

The YOLO architecture is designed to process an entire image in a single forward pass, enabling simultaneous prediction of bounding boxes and object class probabilities. Unlike traditional object detection methods that rely on region proposal networks, YOLO divides the input image into a grid and directly predicts bounding boxes and class probabilities for each cell. This design allows YOLO to achieve a balance between speed and accuracy, making it ideal for real time applications, Figure **3.7** shows a brief architecture of the CNN model that forms the backbone of YOLO **[27]**.

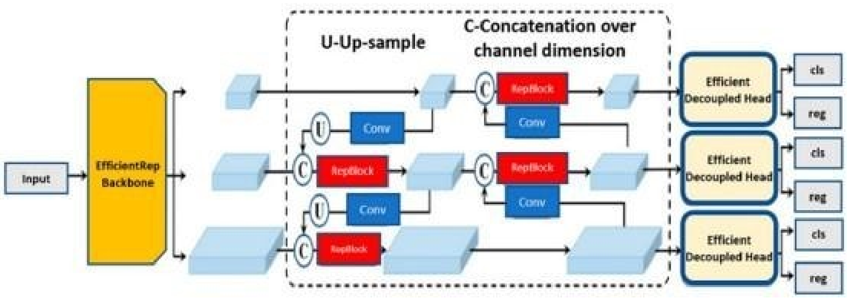

Figure 3.7: YOLO CNN Backbone Architecture

YOLOv8 is chosen as the primary object detection model for its advancements over previous YOLO versions. It offers high speed, real time detection with improved accuracy through refined bounding box predictions and confidence scores, effectively identifying small or occluded objects. Its advanced architecture includes anchor free detection and enhanced spatial attention, making it suitable for complex traffic scenarios. Additionally, its scalability allows it to adapt to various input resolutions, accommodating diverse camera setups.

The integration of YOLOv8 will focus on the following key tasks:

1. **Vehicle Detection and Classification:**

   Vehicle Detection and Classification: YOLOv8 will identify and classify vehicles, including cars, buses, and trucks, while tracking their movement across intersections.

2. **Traffic Signal Detection:**

   By recognizing the state of traffic lights (red, yellow, or green) in real time, YOLOv8 will assist in determining compliance or violations.

3. **Violation Detection:**

   YOLOv8 will process video streams to detect red light violations, and other rule breaches by monitoring vehicle positions relative to stop lines, traffic signals, and pedestrian crossings.

4. **OCR Integration:**

   DL based OCR will be used alongside YOLOv8 to extract license plate information for vehicles flagged in violations, enabling authorities to enforce regulations effectively.

The preprocessing pipeline will ensure the local dataset is optimized for YOLOv8 model training. Figure **3.8** shows the main steps for this model, these steps include:

- **Data Cleaning**

  Removing corrupted entries, duplicates, and irrelevant information.

- **Data Annotation**

  Labeling frames with bounding boxes for vehicles, traffic lights, and violations essential for supervised learning in YOLO.

- **Normalization and Augmentation**

  Scaling pixel values and generating dataset variations to improve YOLOv8's robustness to lighting, weather, and occlusions.

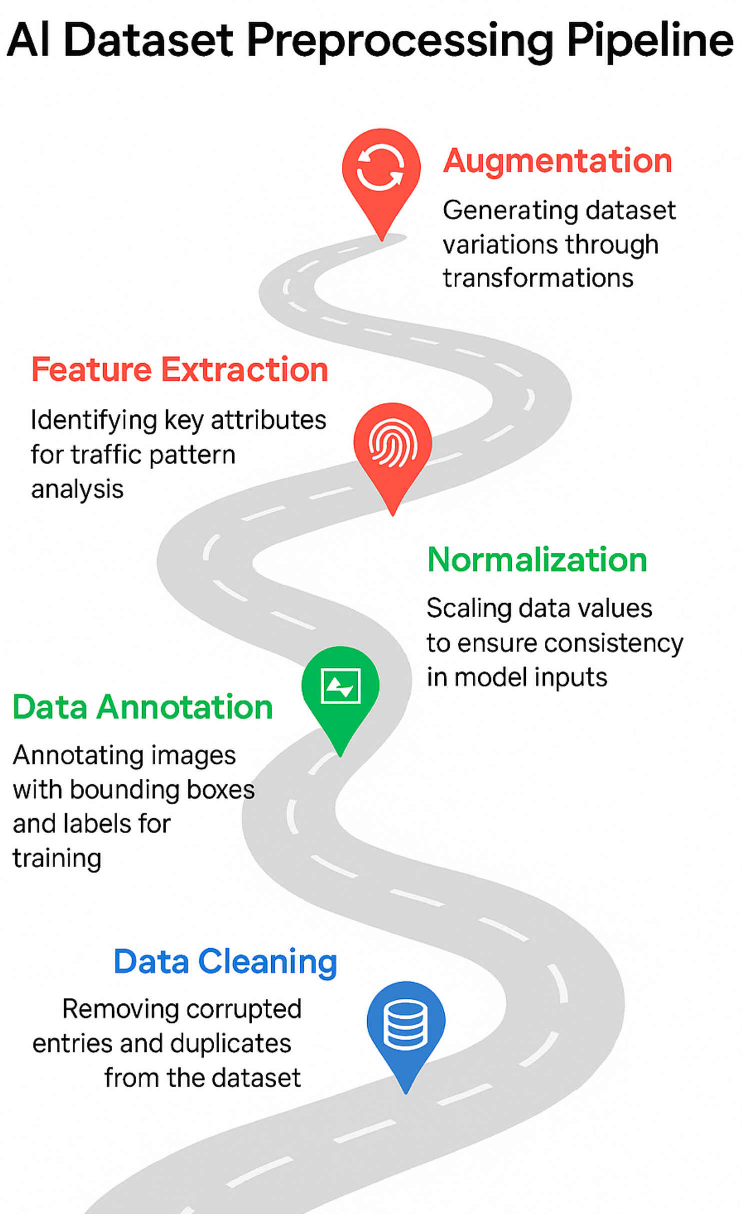

Figure 3.8: Preprocessing Pipeline for YOLOv8

By leveraging the local Palestinian dataset, ensuring that the YOLOv8 model is tailored to regional challenges, such as varying traffic densities, and unique road layouts. The combination of YOLOv8's speed, accuracy, and flexibility with the high quality preprocessing pipeline will enable real time, accurate detection of traffic conditions and violations. This will not only improve traffic monitoring but also enhance enforcement mechanisms, reducing congestion and promoting road safety.

3.5 SORT Tracking Algorithm

Once vehicles are detected in each frame by YOLOv8 detector, maintaining a consistent identity for each vehicle across frames is critical for counting and logging violations. The SORT algorithm is employed , which combines a Kalman filter for motion prediction with a frame by frame data association step based on Intersection over Union (IoU).

- **Kalman Filter Prediction**

  SORT maintains a separate Kalman filter for each active track. At each new frame, the filter projects the previous bounding box state forward in pixel space (position and size), providing a prediction of where each vehicle is expected to appear next.

- **IoU-based Assignment**

  The newly detected bounding boxes are matched to these predictions by computing the IoU between every predicted box and every detection. A simple Hungarian (Munkres) solver then finds the optimal one to one assignment, associating each detection with the most likely existing track. Unmatched predictions and detections initialize "lost" or "new" tracks, respectively.

- **Track Management**

  SORT prunes tracks that disappear for more than a configurable number of frames ("max age") and ignores very short lived tracks ("min hits"), avoiding spurious assignments from intermittent false positives.

This lightweight approach runs in real time on CPU, scales gracefully to dozens of vehicles per frame, and integrates seamlessly with YOLOv8. SORT ensures each car has a stable track ID, enabling accurate per approach counting, congestion analysis, and linking of violation snapshots back to the correct vehicle.

3.6 Web Technologies Overview (React, FastAPI, MongoDB)

To present AI outputs in an intuitive, role aware dashboard, a modern, JavaScript centric fronted paired with a Python API backend and a flexible No SQL database was selected:

- **React (with Type Script)**

  React's component driven model and virtual DOM make it ideal for building interactive UIs. Type Script is Used to enforce compile time type safety, mirroring our backend Pedantic schema and preventing data mismatch. React hooks manage asynchronous data fetching (e.g. chunk lists, violation logs) and local UI state (dark mode toggle, selected chunk), while React Router enables secure, client side navigation between the Traffic and Violations sections.

- **Tailwind CSS**

  Utility first classes allow rapid layout and styling without writing custom CSS. Responsive grids adapt charts and tables for any screen size, and dark mode support comes "for free" via a single class switch, ensuring readability in all lighting conditions.

- **FastAPI**

  On the backend, Fast API exposes traffic records and violation documents as well typed JSON endpoints. Thanks to its asynchronous design, automatic OpenAPI documentation, and Pedantic based validation, data contracts is defined once and have full request/response validation, ultra fast performance, and self documenting API specs. CORS middle ware ensures only authorized front end origin can consume the API.

- **MongoDB**

  The output of both pipelines is stored in two collections records for per chunk traffic data and violations for best frame images, OCR'd plates, and clip references. MongoDB's schema flexible, JSON style documents align perfectly with systems' AI outputs, allowing evolving record and violation formats over time without costly migrations. Its horizontal scalability and cloud native drivers let the capacity grow as data volumes increase.

Together, this stack delivers a seamless experience: the React app fetches exactly the traffic or violation data it needs, Fast API serves it securely and reliably, and MongoDB persists it in a format tailored to systems' dynamic computer vision pipelines.

3.7 Summary and Transition

AI has significant potential to transform traffic management by leveraging locally sourced datasets to develop tailored solutions. Ensuring the quality and relevance of the data is a critical step in driving the accuracy and reliability of AI-powered systems. Advanced computer vision techniques, such as YOLOv8, combined with robust methodologies, can address the limitations of traditional systems while providing scalable solutions. However, as noted earlier, these approaches face challenges, particularly regarding infrastructure requirements and scalability in resource-constrained environments.

Chapter 4

Traffic Management System

Contents

4.1 Introduction

Urban intersections are often the bottlenecks that determine overall network performance. In this work tackled the “5 Stars Intersection” in Ramallah a four way crossing connecting Al-Tireh, Ein Munjed, Beitunia, and Ramallah by building a fully end to end computer vision based system. Rather than relying on fixed timing signals, our solution ingests live video from overhead drones, detects and tracks every vehicle in real time, and computes per intersection queue lengths. At each second our recommendation engine then chooses exactly one light to switch (RED→YELLOW→GREEN), holding the others constant, in order to minimize overall congestion. The result is a highly modular pipeline from raw video through YOLOv8 detection, SORT tracking, JSON recommendation logging, to an agent based simulator and web dashboard that demonstrates clear performance gains over fixed schedules.

4.2 Methodology

4.2.1 Problem Description

Intersections concentrate conflicting vehicle movements straight through, left turns, right turns into a single node, so any imbalance in signal timings quickly cascades into long queues and stop and go traffic. Our case study involved four virtual traffic light IDs (ID-1. . . ID-4), each governing one approach. In traditional systems, each light cycles on a timer (e.g. 20 s green, 3 s yellow, 20 s red) regardless of queue length. During peak hours (07:30–08:00), one approach might see 20 vehicles queued while a neighboring approach is empty but the timer still turns its light green even if no cars can take advantage. We replace that rigid timing with a data driven controller that, each second, reads the current vehicle counts and issues a “which light next?” recommendation to clear the worst queue first and keep traffic flowing smoothly across all four approaches.

Managing Traffic Lights

Traffic light management at intersections is crucial for controlling vehicle flow and ensuring safety. Traditionally, fixed timing systems operate on predefined intervals and lack the flexibility to adapt to real time conditions. For instance, during peak hours, one intersection may experience severe congestion while another remains relatively clear. Without dynamic adjustments, this imbalance results in extended delays and increased fuel consumption.

The goal of intelligent traffic light management is to optimize traffic flow by dynamically adjusting signal timings based on real time data. Consider the following scenarios:

1. **Congested Intersection:**

   When an intersection becomes congested, its signal switches to green to clear the backlog, while neighboring intersections with lighter traffic maintain their current lights.

2. **Balanced Traffic Flow:**

   By monitoring vehicle counts and traffic density, the system coordinates signal changes across multiple intersections to achieve a smooth flow of vehicles.

Visual representations

Various traffic states and congestion scenarios from the dataset help illustrate these challenges and highlight potential solutions:

- **Scenario 1:**

  A congested intersection with a queue of vehicles waiting at a red light, while adjacent intersections have minimal traffic (see Figure **4.1**).

- **Scenario 2:**

  Balanced traffic flow across all four intersections during off peak hours (see Figure **4.2**).

These examples highlight the importance of a dynamic, adaptive system to address the inefficiencies of traditional traffic light management.

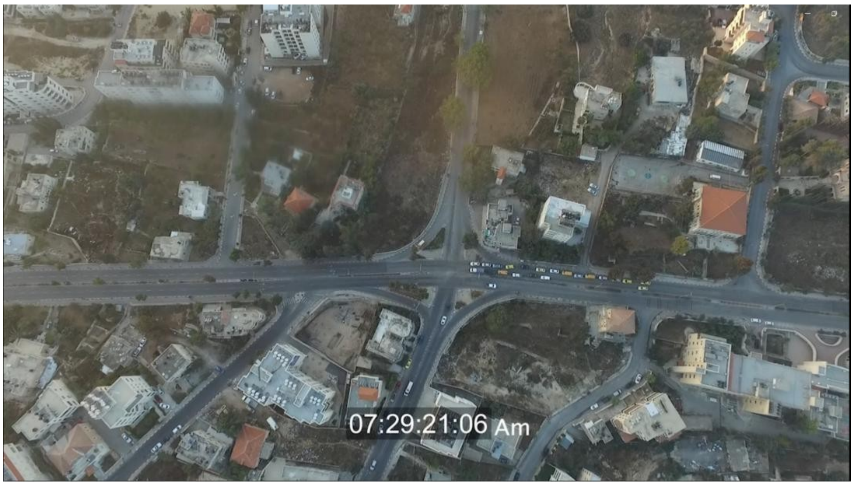

Figure 4.1: Jammed Intersection while Adjacent Approaches Remain Empty.

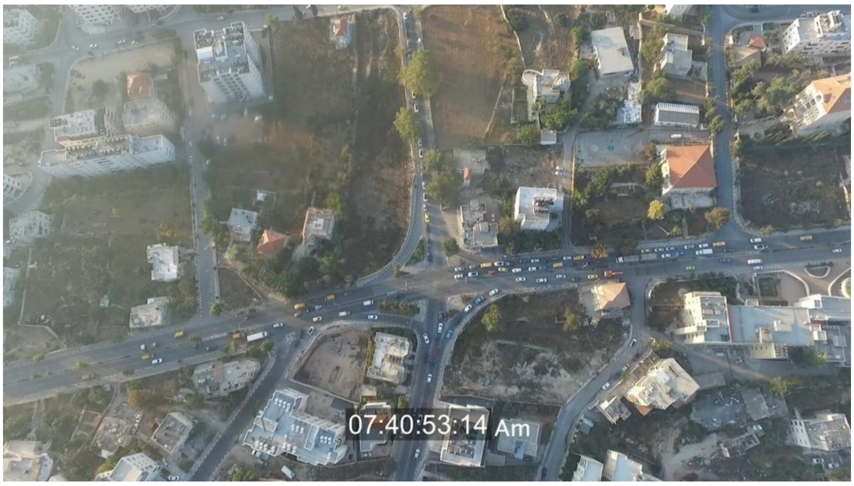

Figure 4.2: Balanced Traffic Flow During off-Peak Hours.

4.2.2 System Assumptions and Components

The proposed system focuses on managing traffic lights at intersections, with a specific case study based on the "5 Stars Intersection," which connects four key locations: Al-Tireh, Ein Munjed, Beitunia, and Ramallah. This intersection network represents a complex and dynamic urban traffic scenario requiring an adaptive traffic management approach. The following assumptions and components define the system:

Intersection Characteristics

- **Intersection Layout:**
  – Each intersection is a cross intersection, where four roads converge.
  – Traffic lights managing forward movement.
- **Traffic Light Distribution:**
  – Four traffic lights are placed at each entry point to the intersection to control vehicles moving forward.

Traffic Light System Components

- **Dynamic Signal Control:**

  The traffic lights at each intersection can be dynamically adjusted based on real time traffic conditions. Signal timings adapt to the vehicle density at each entry point, ensuring efficient traffic flow.
- **Coordination Between Intersections:**

  The traffic lights are not managed in isolation. Adjacent intersections are interconnected, enabling coordinated signal changes to balance traffic flow across the network. For example, if one intersection is heavily congested, neighboring intersections may adjust their signals to reduce the buildup.

Assumptions

1. **Known Locations and Layouts:**

   The placement of all intersections and their corresponding traffic lights is predefined, with the "5 Stars Intersection" serving as the focal point for this case study.
2. **Traffic Density Variations:**

   Traffic density varies significantly across intersections and times of day, requiring adaptive signal control to prevent congestion.
3. **Scalability:**

   While the system is designed for four traffic lights at this intersection, it is scalable and can be extended to additional intersections with similar configurations.
4. **Virtual Traffic Lights:**

   Because drone footage rarely shows real signal heads clearly, we overlaid synthetic traffic light bounding boxes (manually placed once per video frame via annotation) to define exactly where to detect red, yellow, and green states.

5. **Performance Metrics:**

   We measure success both in simulation (cars passing per second) and in detection accuracy (mAP for cars and lights).

4.2.3 Dataset Description

The dataset utilized in this traffic management study comprises drone video data collected from the "5 Stars Intersection" in Ramallah. It is specifically designed to represent distinct traffic conditions, providing a diverse and realistic basis for analyzing and optimizing traffic flow.

Dataset Details:

1. **Data Type:**

   The dataset consists of a medium resolution drone video.

2. **Traffic Scenarios:**

   - **Jammed Traffic:** Recorded at 07:40 AM on Thursday, representing peak hour congestion.
   - **Minimal Traffic:** Recorded at 07:00 AM on Thursday, reflecting off peak conditions.

3. **Duration and Size:**

   Video has a duration of 30 minutes and a size of 231 MB, providing sufficient data for training and testing purposes.

4. **Source:**

   The dataset was provided by the Ramallah Municipality, ensuring its relevance and authenticity for the case study. The videos represent real world traffic dynamics at the specified intersection.

5. **Diverse Scenarios:**

   By including both jammed and minimal traffic conditions, the dataset ensures that the model can handle a wide range of scenarios, from peak congestion to smooth traffic flow.

Dataset Characteristics:

1. **Time of Day:**

   Captures dynamic morning conditions around 07:00–07:40 AM, when intersection densities change rapidly.

2. **Traffic Variability:**

   The inclusion of both heavily jammed and minimally congested traffic scenarios provides a balanced representation of real world conditions, ensuring robust model performance.

3. **Video Format:**

   The high resolution videos allow for detailed analysis of vehicle movements, traffic signal states, and overall intersection dynamics. The dataset serves as a foundation for developing and testing the proposed traffic management system. Its authenticity and relevance were ensured through a collaborative process with the Ramallah Municipality, making it well suited for real world applications in urban environments.

4.3 Proposed System

4.3.1 Introduction

Our core detection model is YOLOv8 for real time speed and aerial accuracy. For each video frame, the pipeline (see figure **4.3**):

1. Runs YOLOv8 to detect vehicles (class 0) and lights (classes 1–3).
2. Counts vehicles per Region of Interest (ROI) polygon for each intersection.
3. Resolves light states inside virtual boxes by priority (RED>YELLOW>GREEN).
4. Applies recommendation logic to choose the next green light and its duration.
5. Logs JSON entries of counts, current states, and recommendations.
6. Stores records in MongoDB for downstream dashboard and simulation.

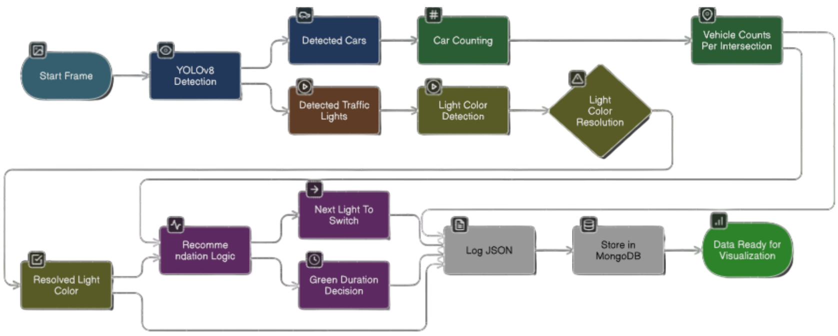

Figure 4.3: Proposed Model Sequence.

4.3.2 Dataset Collection

The dataset collection process was a critical phase of the project, involving considerable effort and collaboration with local institutions to acquire high quality and relevant data for traffic management. Below is a detailed account of how the dataset was obtained:

Efforts to Locate the Dataset

Initially, the search for an appropriate dataset spanned three to four weeks, during which various institutions were approached, including:

- **Academic Sources**:

  Attempts were made to leverage existing traffic datasets from university projects and public research archives.

- **Governmental Bodies**:

  Agencies responsible for urban planning and traffic management were contacted to explore the availability of real world traffic data.

Despite these extensive efforts, none of the sources could provide a dataset suitable for the project's objectives particularly concerning the "5 Stars Intersection" in Ramallah.

Collaboration with Ramallah Municipality

After exhausting other avenues, contact was made with the Ramallah Municipality, which oversees traffic management in the region. Initially, the municipality declined the request, citing the need for formal documentation to justify the dataset's usage.

Official Authorization

To address the municipality's concerns, an official letter was drafted and signed by **Dean Ahmad Alsa'dah**, of the *Birzeit University Faculty of Engineering & Technology, Department of Electrical & Computer Engineering.* The letter formally outlined the project's objectives, emphasizing its academic nature and its aim to develop an AI driven traffic management system as part of a graduation project. After a month long review process, the municipality approved the request, recognizing the project's potential benefits.

Significance of the Dataset

The dataset provided by the Ramallah Municipality is highly representative of real world traffic conditions. This makes it invaluable for training and evaluating the proposed model. Its authenticity ensures that the system can be tailored to the specific challenges of managing traffic in Palestinian urban intersections.

4.3.3 Model Training

Closing the performance gap between public aerial datasets and our much higher drone viewpoint took several weeks of iteration. We first trained YOLO v8 on VisDrone (8625 images) and a Satellite collection, but validation on our own video never climbed above 18.8 % mAP (see figure **4.8**). The core problem was that those reference cameras sit much closer to the roadway, whereas our footage is captured from far above the intersection. So, we decided to bring our dataset into the training, and because time was limited, we managed to manually annotate only ninety six frames of our own video fewer than we would have liked, yet enough to mark every car and the four virtual traffic-light panels that appear in each frame. Figure **4.9** shows a sample annotation in Roboflow. To enlarge this seed set we enabled four augmentation steps, including:

- **Ninety-degree rotations in both directions.**
- **Random rotations between minus fifteen and plus fifteen degrees.**
- **Brightness shifts between minus fifteen and plus fifteen percent.**
- **Gaussian blur with $\sigma \leq 2.5\,\text{px}$.**

The result was the 230-image dataset summarized in Figure **4.4**.

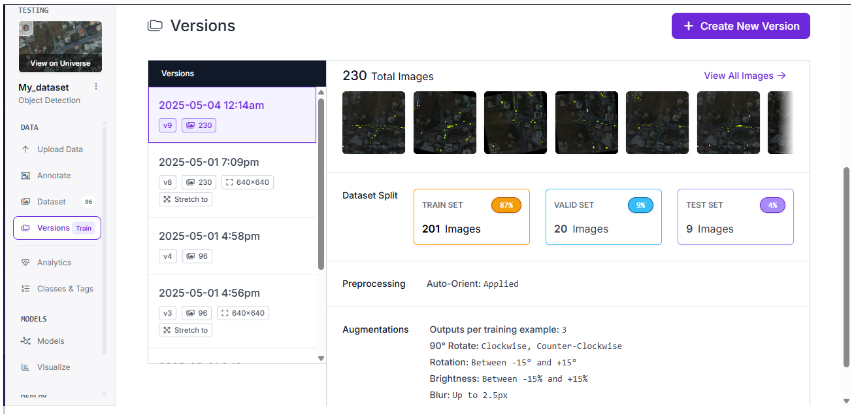

Figure 4.4: Roboflow Version After Augmentation.

A typical augmented frame is illustrated in Figure **4.5**, which already contains our virtual traffic light overlays.

Figure 4.5: Augmented Frame Example

We then merged the 230 custom images with the VisDrone corpus and adopted a seventy percent / thirty percent split for training and testing across the combined set. Figure **4.6** juxtaposes our high altitude scene with a VisDrone frame, highlighting the domain shift the model had to learn.

Figure 4.6: High and Low Altitude Drone from Same Training.

After fine-tuning on this blended data, the detector finally achieved 71.5 % mAP (Figure **4.7**), nearly four times better than the VisDrone only baseline in Figure **4.8**. The model can now recognize both distant vehicles and the overlay traffic lights inserted during preprocessing.

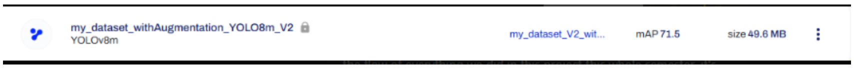

Figure 4.7: MAP When Trained on Both VisDrone and our Custom Annotations.

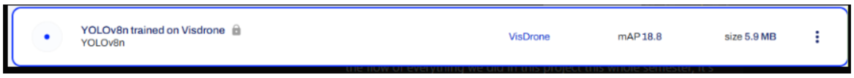

Figure 4.8: MAP When Trained on VisDrone Alone.

Figure 4.9: Example of Custom Annotation.

The concentrated annotation effort, targeted augmentation strategy, and balanced split were therefore essential in producing a detector suitable for the intersection management pipeline.

4.3.4 Data Preprocessing

Before the inference pipeline can interpret live traffic conditions and recommend signal changes, each raw video clip must be distilled into milestone frames and annotated with the information our algorithms expect. The preprocessing stage therefore filters for yellow-phase transition frames and enriches them with high altitude vehicle and traffic light detections, producing clean, labeled inputs for the downstream analysis described below:

- **Frame Selection**

  The dataset consists of videos recorded at 30 Frames Per Second (FPS). To focus on critical moments in traffic regulation, only frames in which one of the traffic lights appears yellow are selected. These moments represent key transition points in vehicle flow, making them ideal for analysis and further processing.

- **Feature Extraction**

  Each selected frame is analyzed to extract key features essential for traffic management, including:

  - **Vehicle and Traffic Light Detection:** Detection is performed using aerial drone footage with a YOLOv8 model trained specifically to detect vehicles from above and to identify and locate traffic lights within the frames.
  - **Bounding Boxes:** Each detected vehicle and traffic light is enclosed in a bounding box with a corresponding label, facilitating subsequent analysis and processing.

4.3.5 Image Processing

Video Frame Processing and Analysis

In this stage, we focused on processing video frames to extract meaningful spatial and contextual information needed for effective traffic management.

Region of Interest (ROI) Detection

We used polygonal (ROIs) to identify key intersections within the aerial footage. Each intersection was manually defined and assigned a unique Intersection ID (e.g., ID-1, ID-2, etc.), allowing for precise tracking and analysis.

Vehicle Counting per Intersection

For every selected frame, we counted the number of detected vehicles within each intersection's ROI. This allowed us to evaluate traffic density and flow patterns for each area individually, enabling better control decisions later in recommendations.

Realistic Traffic Light Placement

Due to the nature of aerial drone footage, traditional traffic lights were not directly visible. Therefore, we created virtual traffic lights that mimic real world behavior and placement. These virtual lights were overlaid on the frames in positions that match real intersections, preserving realism and enabling clear state detection (red/yellow/green).

4.3.6 System Implementation

Vehicle and Traffic Light Detection Logic

- Loaded YOLOv8 model once per worker.
- Ran inference on each frame, filtering detections by class (cars vs. lights).

- Transformed normalized coordinates into pixel-space and queried within each ROI.

Car Counting and Congestion Analysis

- For each polygonal ROI, used `cv2.pointPolygonTest` on each detected car's center to match per intersection counts.
- Maintained per-chunk JSON files recording counts at frames where lights transitioned from yellow to green.

Dynamic Signal Recommendation Algorithm

- At every yellow onset, flagged the intersection to start counting.
- On the yellow→green transition, wrote the counts plus a 75-frame buffer to file.
- Because YOLOv8 sometimes misses cars, we base each per chunk recommendation on the single best frame the one that contains the highest detected car count for the intersection. Counts from that frame are then weighted ($2\times$ for ID-2 and ID-4, $1\times$ for ID-1 and ID-3) before entering the decision engine. Figure **4.10** places the best frame on the left and a representative neighboring frame on the right, illustrating the gain in perceived car numbers.
- Sequentially filled a 4-entry recommendation array: for each "current" intersection (in fixed order), selected the remaining highest weight candidate and set its green duration to count $\times 2 + 2$ seconds.

Figure 4.10: Example of Best Frame Selection Effect.

Finally, after processing each chunk, the following info are combined into a json record and saved:

- `Chunk's number:`

  Number (ID) of the chunk.

- `Video path:`

  Video of the current chunk.

- `Best frame:`

  Array of dictionary, having "ID": base64(image) of best frame when taking recommendation action.

- `Recommendations:`

  Containing details of each chunk's current state, recommended state, duration given for recommended state, counts in each intersection, and states of each intersection recommendation was taken.

- `Real World:`

  Array of dictionaries that represents cars passing in real world from each intersection

4.3.7 System Outputs

All outputs flow into a MongoDB collection named records with this schema:

```
{
  "chunk": 0,
  "video_path": "clips/chunk_0.mp4",
  "best_frames": [
    { "id":"ID-1", "image": "<base64>" }, …
  ],
  "recommendations": [ {current, recommended, duration_sec, all_counts, all_states}, … ],
  "real_world": [ {"id":"ID-1","cars_passed_in_real":12}, … ]
}
```

Figure 4.11: System Output.

This uniform format powers both our FastAPI backend and the React dashboard, allowing seamless retrieval of chunk lists, detailed recommendations, and best-frame images for each intersection.

4.3.8 Validation and Simulation

To validate our recommendation engine's ability to optimize congestion, we built a lightweight, agent based simulator in Pygame that "replays" the JSON outputs of the system in a virtual intersection environment. The goals of the simulation are:

1. Verify correctness of the traffic light state recommendations.

2. Measure performance improvements in particular average queue length and throughput against a fixed timing baseline.

Simulation Engine

- World Setup
  - Four cross intersection zones, each mapped to one of our virtual traffic light IDs and represented by a rectangular ROI matching the drone footage geometry.
  - Vehicle agents queue at red lights and, when their approach turns green, depart at a fixed service rate calibrated so that all queued cars clear within the green phase.
- Control Loop (1 s timesteps)
  1. Read the next JSON recommendation entry.

2. Update the corresponding light's state (2.5 s yellow → green duration → 2.5 s yellow).
3. Advance the simulation by one second:
   - **Departures:** vehicles at green lights exit at the calibrated service rate (e.g., 1 car/s).
   - **Arrivals:** new vehicles join each queue according to a Poisson process, with rates estimated from the drone video.
4. Log metrics: per phase queue lengths and throughput (cars served per second).

Results:

When compared to a conventional fixed 100 s cycle, our dynamic controller consistently increased average throughput by approximately 0.25 cars/s and reduced peak queue lengths across all approaches and the simulation we build reflect the recommended case for a full cycle, (see figure **4.12**).

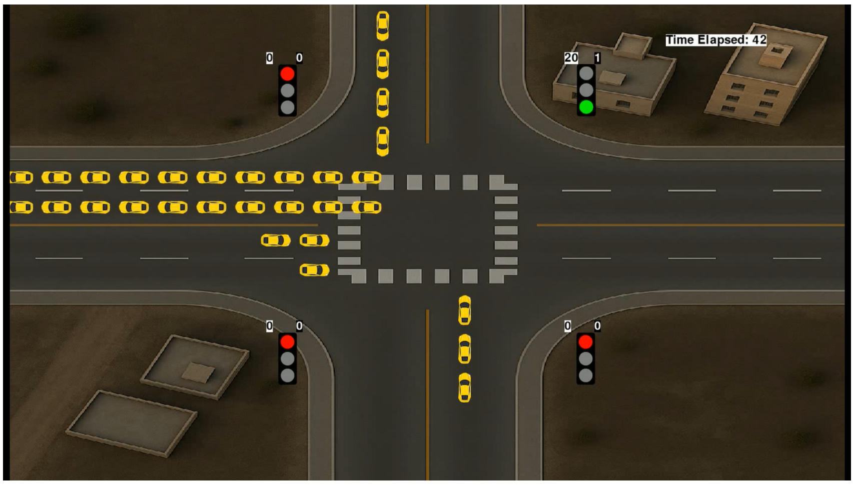

Figure 4.12: Simulation.

In summary, this methodology delivers a comprehensive, end to end intelligent traffic management solution from dataset annotation and YOLOv8 fine-tuning through real time inference, recommendation logging, and rigorous simulation validation.

Chapter 5

Red Light Violation Detection System

Contents

5.1 Introduction

Traffic violations, particularly running red lights, significantly compromise road safety, increasing accident risks and potential injuries. Addressing this challenge effectively requires robust and automated detection systems capable of accurately identifying and documenting violations in real time. This chapter presents an automated solution for detecting red light violations using advanced computer vision techniques and DL algorithms. Unlike Problem 1, this problem addresses a roadside scenario with street level data collection and a different methodological approach.

5.2 Methodology

5.2.1 Problem Description

In contrast to the previous problem, which utilized aerial drone footage for intersection management, this scenario involves detecting red light violations from a street level point of view (POV). The chosen setup represents typical roadside conditions, featuring a single lane controlled by a virtual traffic signal. The system aims to automate the detection process and accurately document each violation event with sufficient detail for enforcement purposes.

- **Roadside Scenario (street level, Single Lane)**:

  The system focuses on a single lane road scenario. This controlled environment simplifies the detection process, ensuring clear visibility of vehicles and traffic signals for accurate detection.

- **Virtual Traffic Light Concept**:

  Since the provided dataset is recorded under the physical traffic light, a virtual traffic signal was superimposed onto video frames using OpenCV, enabling precise control over the state of the light and facilitating more accurate violation detection.

5.2.2 System Assumptions and Components

The violation detection system is built around several key assumptions and components:

- The video data is captured from a fixed camera angle positioned at street level, ensuring clear visibility of approaching vehicles.
- Vehicles are expected to clearly cross the defined stop line during a red light state for a violation to be detected and registered.
- The system components include:
  - YOLOv8 (default model, no fine-tuning) for vehicle detection.
  - A customized YOLOv8 based model specifically trained for license plate detection.
  - SORT tracking algorithm to maintain consistent vehicle identification and tracking across frames.
  - OCR integration for automated license plate recognition.

5.2.3 Dataset Description

The dataset utilized in this red light violation study comprises street level video data captured by our project team at the "5 Stars Intersection". Unlike the overhead drone footage in Chapter **4**, these recordings place the camera at driver eye level, beneath the physical traffic signal, thereby providing an unobstructed view of both approaching vehicles and their license plates and the stop line.

Dataset Details:

1. **Data Type:**

   The data set consists of a high resolution mobile camera video.

2. **Duration and Size:**

   Video has a duration of 29 sec and a size of 620 MB, providing sufficient data for testing purposes.

3. **Source:**

   The dataset was collected first hand by the team, ensuring full control over data quality and its relevance and authenticity for the project. The videos represent real world cars movement at the specified intersection.

4. **Visibility of Critical Elements:** Camera placement guarantees full visibility of the cars and their number plates,stop line interactions, making the footage ideal for the project.

Dataset Characteristics:

1. **Time of Day:**

   Captures normal cars flow around 08:02–08:04 PM, when intersection contains a decent cars flow.

2. **Video Format:**

   The high resolution videos allow for detailed analysis of vehicle movements. The dataset serves as a foundation for developing and testing the proposed red light violation detection. Its authenticity and relevance were ensured through a collaborative process with the local police attended at the intersection, making it well suited for real world applications in urban environments.

Data Utilization:

The street level dataset was utilized for system testing. No frames from these clips were used for training or fine-tuning either YOLOv8 or the plate detector. One model used the original weights provided by the Ultralytics official page, and the other model used a custom weight provided by a GitHub source **[28, 29]**.

5.3 Proposed Model

5.3.1 Introduction

This section introduces the detailed methodology behind the proposed violation detection model, highlighting the integration of real time vehicle detection and tracking, license plate recognition, and automated violation documentation processes.

5.3.2 Dataset Collection

The dataset collection process was a critical phase of the project, involving considerable effort and collaboration with local institutions to acquire high quality and relevant data for red light violation detection Below is a detailed account of how the dataset was obtained:

Efforts to Locate the Dataset:

Initially, the search for an appropriate dataset spanned three to four weeks, during which various institutions were approached, including:

- **Academic Sources:**

  Attempts were made to leverage existing violation datasets from university projects and public research archives for a local dataset captured at the same intersection we are working on.

- **Governmental Bodies:**

  Agencies responsible for urban planning and traffic management were contacted to explore the availability of real world traffic data.

Despite these extensive efforts, none of the sources could provide a dataset suitable for the project's objectives particularly concerning the "5 Stars Intersection" in Ramallah.

Collaboration with Police:

After exhausting other avenues, contact was made with the police department, which oversees road management in the region. The local head police attended at the intersection approved our request of recording a one lane of the intersection and by that we had the needed dataset for the project.

5.3.3 Data Preprocessing

Preprocessing involved critical preparation steps to ensure dataset readiness:

- **Virtual Traffic Light Creation**:

  Since the video is a raw recording under the physical traffic light, which is unreadable, the team created a virtual traffic light to replace the unseen light, with state changes (red, yellow, green).

- **ROI Extraction**:

  Since a violation happens when a car crosses a defined stop line during a red light, the ROIs for the stop line and traffic light state were extracted and saved to focus the detection system.

5.3.4 Model and Tracking Methodology

The detection methodology integrates several advanced techniques and algorithms:

- **YOLOv8 Default Model** :

  The YOLOv8 default pre-trained model was employed to detect vehicles and virtual traffic lights efficiently and accurately without additional training to provide the images required for the violating car

- **License Plate Detection Model (Custom YOLOv8)**:

  A specialized YOLOv8 model was selected to detect license plates within vehicle bounding boxes, providing precise plate localization which is necessary for OCR processes.

- **SORT Integration and Vehicle Tracking**:

  SORT algorithm was incorporated to continuously track each detected vehicle, assigning unique identifiers (IDs) to vehicles across multiple frames, thus ensuring robust violation tracking and documentation is correct for both cars and plates detected as shown in figure **5.1**.

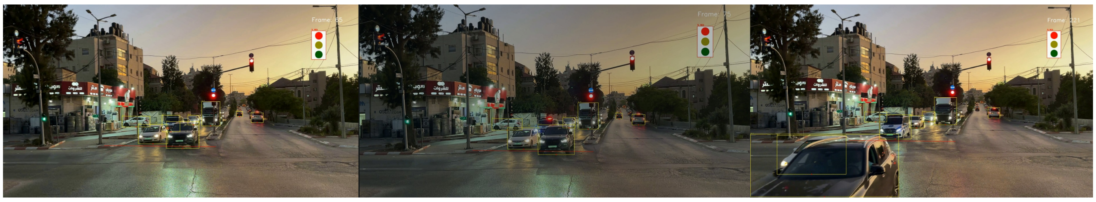

Figure 5.1: Tracker Algorithm in Action.

As shown in the figure a case for tacking the cars if YOLO missed detections for some frames the SORT saved the id for the missed car assisting the YOLO to keep track of the cars in the video.

5.3.5 Violation Detection Logic

The system employs rigorous logic to identify red light violations:

- **Traffic Light State Detection**:

  The system analyzes the virtual traffic light state (red, yellow, green) in real time using simple yet effective color detection algorithms based on Hue, Saturation, and Value (HSV) thresholds to determine the traffic light state

- **Violation Trigger**:

  Violations are detected based on vehicle positions represented by a dot underneath the detected car when the dot is crossing the predefined virtual stop line. A violation is triggered during a red light state.

- **Best-frame Selection Criteria**:

  To accurately document each violation, the system selects the optimal frame based on criteria including license plate clarity, image sharpness, and vehicle positioning relative to the stop line , to feed the frame to the OCR.

5.3.6 System Outputs

The model produces detailed outputs to document violations:

- **Violation Snapshots (Best-frame Selection)**:

  High quality images capturing the exact moment of violation and selecting the best frame among them to clearly display the vehicle and license plate also to use them in the plate characters recognition (see figure **5.2**).

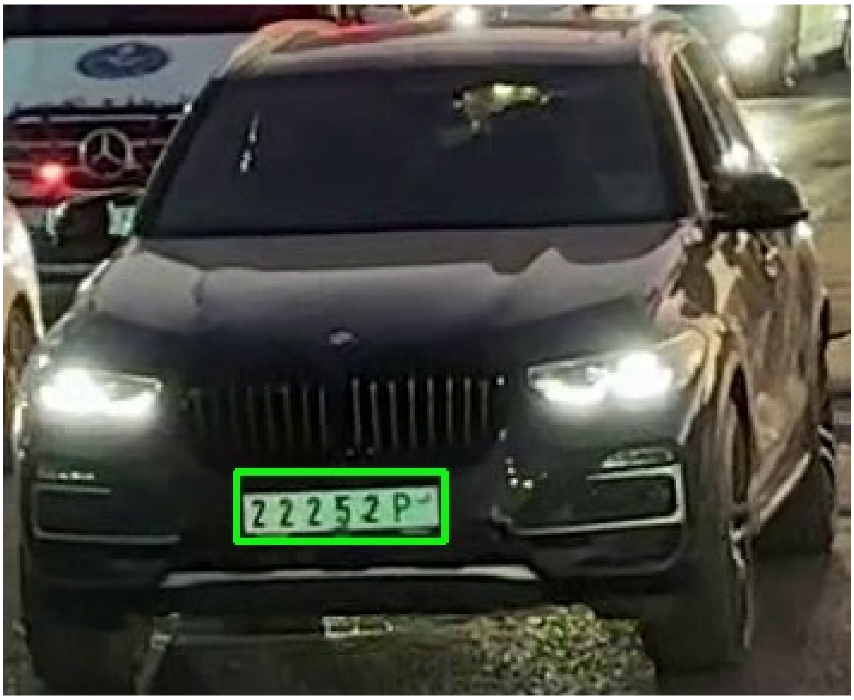

Figure 5.2: Vehicle's Plate.

- **OCR for License Plate Recognition**:

  OCR is applied to the best frame selected to extract the license plate image, automating the recognition and storage of plate information for enforcement use (see figure **5.3**).

Figure 5.3: License Plate Region.

- **Violation Logs and JSON Reports**:

  Structured logs detailing each violation event, including timestamp, vehicle identification, and license plate information, stored in JSON format.

- **Short Violation Clips Generation**:

Short video clips highlighting the few seconds surrounding each violation 1 second before the crossing and 2 second after the crossing, events are automatically generated, providing contextual evidence supporting each detected violation (see figure **5.4**).

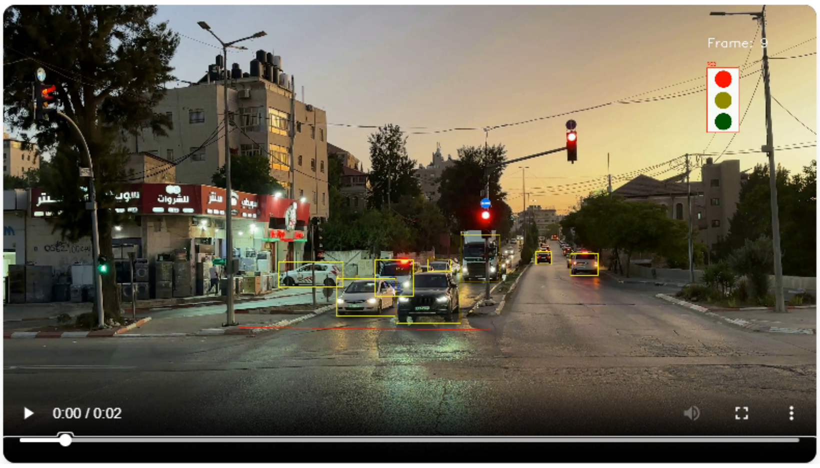

Figure 5.4: Violation Video.

- **MongoDB Storage of Violations**:

  In addition to local storage, all relevant data is uploaded to a MongoDB database.

  The JSON formatted metadata for each violation (plate number, car id, encoded best-frame images, encoded short video clips) (see figure **5.5**)


```
{
  "_id": {
    "$oid": "68643035c1a204685338455f"
  },
  "car_ID": "21",
  "plate_text": "63883H",
  "plate_detected": "/9j/4AAQSkZJRgABAQAAAQABAAD/2wBDAAIBA(
  "video_path": "violations/car_21_violation.mp4"
}
```


Figure 5.5: Sample MongoDB Document.

- **real time SMS notification** :

  After a successful DB write, the system queries the Vehicle-Registration API with the recognised plate. If a mobile number is returned, an Arabic SMS is sent through the Jawwal REST gateway (see figure **5.6**).

Figure 5.6: SMS Message.

Chapter 6

Shared Visualization Platform (Website)

Contents

6.1 Introduction

The two computer vision pipelines developed in this project dynamic traffic signal optimization (Chapter **4**) and red light violation detection (Chapter **5**) produce complementary but disparate outputs: JSON recommendations, per approach queue length logs, best frame snapshots, OCR'd plates, and short violation clips. A unified web interface brings these artifacts together so that authorized stakeholders (traffic engineers, enforcement officers, urban planners) can:

- Consolidate all traffic management and violation data in one place
- Browse results without juggling files or custom scripts
- Correlate congestion metrics with violation events (e.g., "Did green-light reallocation upstream reduce downstream red light runs?")
- Collaborate via role aware views engineers see throughput , officers see violation snapshots and plate details

By monitoring isolated outputs into a interactive dashboard, the platform accelerates understanding, coordination, and data driven decision making.

6.2 Architecture and Design

The site follows a three-tier architecture that cleanly separates data, logic, and presentation:

6.2.1 Data Layer

- **MongoDB** stores two collections:
  - `records`: per chunk traffic counts, current/recommended signal states, throughput logs.
  - `violations`: best frame images, OCR'd plate text, timestamps, and short video clip references.

6.2.2 Backend Layer

- **FastAPI** serves as the RESTful service, exposing well typed JSON endpoints to retrieve chunks, individual records, and violations.
- **CORS middleware** allows the React frontend (running on localhost:3000 in development) to query the API seamlessly.

6.2.3 Frontend Layer

- **React with TypeScript** provides a component driven UI, ensuring type safety and predictable data flows.
- **Tailwind CSS** delivers rapid, responsive styling, including dark-mode support and utility-first layout.

6.3 Web Platform Features

The platform is divided into two core sections Traffic Management and Violation Visualization accessible via a secure login.

6.3.1 Login and User Authentication

- **Credentials Based Sign In**
  - Users enter a username/password combination.
  - On success, the API returns a short lived JWT stored in `localStorage` or an HTTP-only cookie.
- **Protected Routes**
  - React Router guard ensures only authenticated users can access dashboards.

– Unauthorized attempts redirect back to the login page.

- **Role Based Access (Future)**
  – While the MVP supports a single user role, the architecture easily extends to multiple roles (e.g., Viewer, Enforcer, Admin) with scoped API tokens.

6.3.2 Interactive Dashboard for Traffic Management

1. **Metrics:**
   - **Performance Summary:**
     Displays the overall improvement achieved by the recommendation system in reducing congestion and optimizing traffic light timing.
   - **Best Frames Card:**
     Shows four key images one for each intersection ID capturing the frame when the system took a recommended action, highlighting vehicle density and light state.
   - **Recommendation Panel:**
     Displays current vs. recommended traffic light states along with vehicle counts per approach. This helps users understand the basis for the system's decisions (see figure **6.1**).

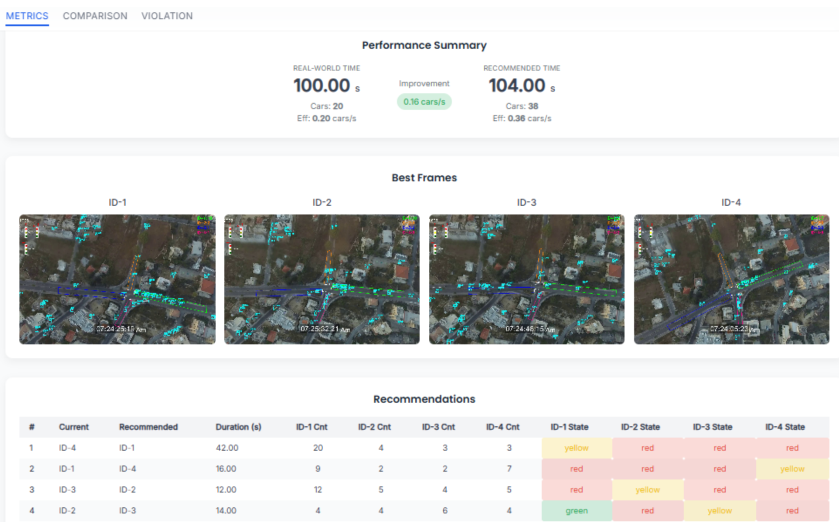

Figure 6.1: Metrics Tab

2. **Comparison:**
   - **Side-by-Side Playback:**
     Plays the original drone video alongside the simulator replay powered by our JSON recommendations, allowing visual comparison of queue reduction (see figure **6.2**).

- **Interactive Controls:**
  Provides play/pause and scrubbing options to clearly observe how dynamic control outperforms fixed timing schedules in real time.

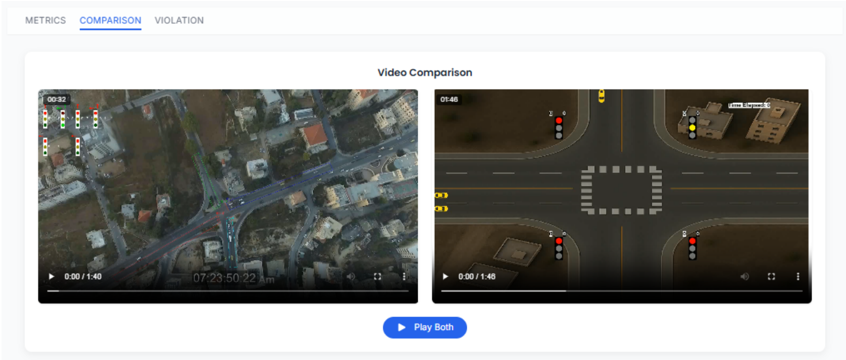

Figure 6.2: Video Comparison Drone View vs Simulation with Recommendations

6.3.3 Violation Visualization Interface

Under the Violations menu, policeman can:

- **Browse a table** of all red light events, with columns for Car ID.
- **Drill into a detail view** showing (see figure **6.3**):
  - The best frame snapshot at the moment of violation
  - A short clip (1 s before to 2 s after crossing) for context
  - OCR detection of plate showing plate's number

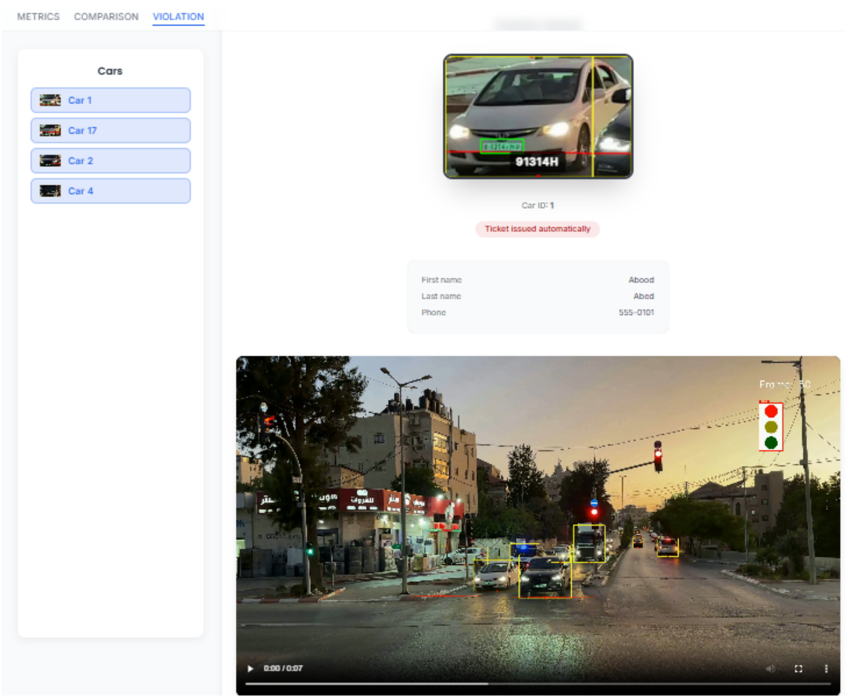

Figure 6.3: Violation Tab

Chapter 6 has presented the concepts, design, and feature set of our unified web platform. By seamlessly integrating the outputs of our traffic management and violation detection pipelines into an intuitive, role aware dashboard, we empower stakeholders to monitor, analyze, and act on real world traffic data, closing the loop from raw video to policy decisions.

Chapter 7

Conclusions and Future Works

Contents

7.1 Traffic Management System

7.1.1 Conclusion

This project presents a comprehensive framework for intelligent traffic management at intersections, particularly focusing on the "5 Stars Intersection" in Ramallah. By leveraging state of the art computer vision techniques and the YOLOv8 object detection model, the system dynamically adapts traffic light timings based on real time video analysis, thereby improving traffic flow and reducing congestion. The proposed system demonstrates the potential for significant enhancements in urban mobility through automated traffic management, providing detailed insights for authorities via a user friendly web interface.

Key Contributions

- **Robust Methodology**:

  Local data collection, preprocessing, and system testing tailored to real world traffic scenarios.

- **Integrated System**:

  Combines vehicle detection, classification, and traffic light recognition to optimize signal timing and enhance safety.

- **Responsive UI/UX**:

  A secure and user friendly interface enabling authorities to access detailed traffic reports for improved decision making and communication.

The results suggest that data driven systems like this can bridge the gap between traditional and intelligent traffic management approaches, offering scalable solutions that cater to the growing needs of urban environments.

7.1.2 Limitations and Challenges

Despite its potential, the project faced several limitations and challenges:

1. **Data Access and Collection**: Acquiring the dataset from the Ramallah Municipality required formal approvals, leading to delays that extended the project timeline. Furthermore, despite the lengthy wait, the data ultimately provided did not fully meet our requirements in terms of scope and quality, and locating a more suitable alternative dataset proved exceptionally challenging.

2. **Hardware and Resource Constraints**:

   Real time processing with YOLOv8 requires high performance hardware, which may limit deployment in resource constrained environments.

3. **Dataset Diversity**:

   The dataset used in this study represents a specific location and time of day. Broader generalization may require additional data covering varying weather conditions, times, and traffic patterns.

4. **Environmental Factors**:

   Detection accuracy may be effected by poor lighting, changing weather conditions (heavy rain or fog), and overlaying caused by larger vehicles.

7.1.3 Future Work

To address these challenges and enhance the system's effectiveness, the following directions for future work are proposed:

1. **Expanding Dataset Diversity**:

   Collect and integrate additional datasets that capture diverse weather conditions, times of day, and varying traffic densities to improve the model's robustness and generalization.

2. **Integration of Mobile Application**:

   Extend the project to include a mobile version that enables the police men to check the available intersection's status and log the violations on field.

3. **real time Deployment and Optimization**:

   Implement and test the system in a live traffic environment to evaluate real world performance. Additionally, optimize the system's processing pipelines for lower-latency detection.

This future work aims to further scale the system, making it more adaptive, accessible, and impaction for intelligent urban traffic management.

7.2 Red Light Violation Detection System

7.2.1 Conclusion

This work presents an automated red light violation detection pipeline operating at street level, beneath the “5 Stars Intersection” light. By implementing a virtual signal in each frame and leveraging YOLOv8 for vehicle and plate detection, SORT for consistent tracking, and OCR for license plate reading, the system reliably flags any vehicle whose bounding box crosses the stop line during a red phase. Upon each infraction, the clearest evidence frame is extracted, metadata (plate text, timestamp, stop line position) is stored in MongoDB, and immediate SMS notifications are dispatched via the Jawwal gateway. The proposed system demonstrates the potential for significant safety enhancements in urban mobility through automated red light violations detection, providing detailed insights for authorities via a user friendly web interface for enforcement use.

Key Contributions

- **Integrated Detection Pipeline:**

  Fuses YOLOv8 vehicle/traffic light/plate detection, SORT tracking, and OCR into a single workflow for robust violation flagging.

- **Best-Frame Evidence Selection:**

  Dynamically selects the sharpest, most informative frame for each infraction, maximizing OCR readability and legal defensibility.

- **real time Alerting & Logging:**

  Automates SMS notifications and stores violation records (images + metadata) in MongoDB for immediate enforcement and audit.

- **user friendly Interface:**

  Exposes a clean web dashboard for browsing violations, filtering by time/plate.

Together, these components form a streamlined, end to end enforcement workflow that not only detects and documents red-light violations with high accuracy but also delivers real time alerts and centralized record keeping, empowering traffic authorities to act swiftly and efficiently through an intuitive web interface.

7.2.2 Limitations and Challenges

Despite its potential, the project faced several limitations and challenges:

1. **Dataset Scope:**

   Our dataset consists of a single 29 seconds, high resolution street level recording under one traffic phase limiting generalization across other intersection lanes also the dataset supports a specific condition and doesn’t include versions for the same intersection under weather, and lighting conditions.

2. **Data Acquisition Delays:**

   Securing formal police approval and filming at the intersection, led to delays that extended the project timeline.

3. **Hardware Requirements:**

   Real time inference with YOLOv8 and high resolution plate models demands GPU class hardware, hindering deployment on lightweight or legacy edge devices.

4. **OCR Robustness:**

   Plate recognition accuracy degrades under low contrast, oblique angles, or motion blur leading to false negatives or misreads. Pedestrians, parked cars, or weather (rain/fog) can partially obscure stop line crossings or plates, causing missed or spurious detections.

7.2.3 Future Work

To address these challenges and enhance the system's effectiveness, the following directions for future work are proposed:

1. **Expand Dataset Diversity:**

   Capture multi hour, multi weather, multi lane, and multi angle street level videos including left turn and right turn approaches to improve model generality.

2. **real time Deployment and Optimization:**

   Implement and test the system in a live traffic environment to evaluate real world performance. Additionally, optimize the system's processing pipelines for lower latency detection.

These enhancements will advance the system's pipeline, user usability, and overall effectiveness in smart urban traffic control.

Bibliography

[1] T. A. et al., “Review of deep learning applications in traffic management,” *Applied AI Journal*, 2023. Accessed: March 4, 2025.

[2] A. A. et al., “Weight-based braking distance for red-light violations,” *International Journal of Traffic Management*, 2023. Accessed: March 4, 2025.

[3] Palestinian Businessmen Association, “Economic news,” 2023. Accessed: March 4, 2025.

[4] G. J. Chowdary, “Machine learning and deep learning methods for building intelligent systems in medicine and drug discovery: a comprehensive survey,” *arXiv preprint arXiv:2107.14037*, 2021. Accessed: March 4, 2025.

[5] A. C. et al., “Speeding violation detection with yolov8,” *Journal of AI and Traffic Safety*, 2024. Accessed: March 4, 2025.

[6] O. Diyamantha, “Automated traffic law enforcement system,” *ISSN*, 2023. Accessed: March 4, 2025.

[7] M. F. et al., “Mobile traffic violation detection using yolov5 and mobilenet ocr,” *IEEE Transactions on Intelligent Transportation Systems*, 2023. Accessed: March 4, 2025.

[8] H. Gehani, “Traffic signal violation detection system using computer vision,” 2024. Accessed: March 4, 2025.

[9] A. G. et al., “Enhanced traffic light detection using tldm,” *Urban AI Systems Review*, 2024. Accessed: March 4, 2025.

[10] U. Jilani, M. Asif, and M. Y. I. Z. et al., “A systematic review on urban road traffic congestion,” *Wireless Personal Communications*, 2023. Accessed: March 4, 2025.

[11] S. K. et al., “Traffic flow optimization using yolo-based detection,” *Journal of Transportation Analytics*, 2024. Accessed: March 4, 2025.

[12] W. L. et al., “Yolov5s and alexnet for low-light traffic detection,” *AI in Transportation Journal*, 2024. Accessed: March 4, 2025.

[13] K. L. et al., “Big data platforms for traffic violation detection,” *Traffic Engineering Review*, 2022. Accessed: March 4, 2025.

[14] T. M. et al., “Efficient estimation of word representations in vector space,” in *ICLR*, 2013. Accessed: March 4, 2025.

[15] The Miners, "The miners," 2023. Accessed: March 4, 2025.

[16] B. N., P. S., and K. V. K., "Ai-powered traffic lights," in *2023 Intelligent Computing and Control for Engineering and Business Systems (ICCEBS)*, 2023. Accessed: March 4, 2025.

[17] Z. Nie, "Traffic light detection algorithm based on improved yoloxs," in *IOP*, 2024. Accessed: March 4, 2025.

[18] A. P. et al., "Real-time multi-violation detection with yolo and crnn," *Journal of Computer Vision Applications*, 2023. Accessed: March 4, 2025.

[19] J. Q. et al., "Ai-powered adaptive traffic light systems," in *International Conference on Smart Cities*, 2022. Accessed: March 4, 2025.

[20] F. Q. et al., "Automated traffic law enforcement using anpr and deep learning," *Journal of Intelligent Systems*, 2022. Accessed: March 4, 2025.

[21] JESIT, "Ai in traffic safety systems," 2023. Accessed: March 4, 2025.

[22] "Ai applications in transportation," 2023. Accessed: March 4, 2025.

[23] "A survey on deep learning for intelligent transportation systems," *ScienceDirect*, 2023. Accessed: March 4, 2025.

[24] "Deep learning-based object detection in smart city applications," *MDPI Healthcare*, 2023. Accessed: March 4, 2025.

[25] V7 Labs, "Yolo object detection blog post," 2023. Accessed: March 4, 2025.

[26] zaryam Sadeghi, "Yolo v4 vs v5." YouTube video, 2022. Accessed: March 4, 2025.

[27] Y. Z. et al., "Automated red-light violation detection using yolov5s," *Computer Vision in Transportation*, 2024. Accessed: March 4, 2025.

[28] Ultralytics Inc., "Ultralytics – revolutionizing vision ai with yolo," 2025. Accessed: March 4, 2025.

[29] M. Z. Khan, "Automatic license plate recognition using yolov8," 2024. Accessed: March 12, 2025.