Adaptive Traffic Signal Control using Image Analysis and AI Techniques

by

Tanjeemul Islam
19141010
Adnan Karim
18101592
Imrose Murshed
23341029
Nazmul Ahsan
20301368

Final project submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
October 2024

Declaration

It is hereby declared that

1. The project submitted is our own original work while completing a degree at Brac University.
2. The project does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The project does not contain material that has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Tanjeemul Islam
19141010

Adnan Karim
18101592

Imrose Murshed
23341029

Nazmul Ahsan
20301368

Approval

The project titled "Adaptive Traffic Signal Control using Image Analysis and AI Techniques" Submitted by

1. Tanjeemul Islam (19141010)
2. Adnan Karim (18101592)
3. Imrose Murshed (23341029)
4. Nazmul Ahsan (20301368)

Of Summer, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on October 17, 2024.

Examining Committee:

Supervisor:
(Member)

____

Dr. Jannatun Noor Mukta

Assistant Professor
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

____

Dr. Golam Rabiul Alam

Thesis Coordinator
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

____

Dr. Sadia Hamid Kazi, PhD

Associate Professor and Chairperson
Department of Computer Science and Engineering
Brac University

Acknowledgement

We start by praising and thanking Almighty Allah for protecting our physical fitness and mental well-being as well as for creating the opportunity for us to present our project "Adaptive Traffic Signal Control using Image Analysis and AI Techniques" within due time. We are grateful for this opportunity that is presented to us

We are very grateful to our dedicated supervisor, Dr. Jannatun Noor, Ph.D., Assistant Professor in Computer Science and Engineering for her support, motivation, and useful advice throughout this journey of our project. She has been consistently helpful in the further elaboration of our project and ideas on all levels.

The successful completion of this study does not justify individual effort alone but rather efforts from various people. We would like to thank our parents and mentors for having faith in us and supporting us throughout the journey.

Abstract

Traffic jam is a significant obstacle that makes travel within the city very inconvenient. The systems, such as traffic lights, that are currently in place do not solve this problem. In usual scenario, traffic lights have a fixed pattern of changing from red to green to yellow with fixed timing. But, traffic in roads can be very unpredictable, which is a key point that we must introduce automated traffic signals. In this project, we proposed and designed a system where traffic lights change according to vehicle density on each road. We created an interface to demonstrate the simulation. At first, we trained our data with the YOLOv7 model but the results were not satisfactory so later on we trained with later YOLOv8 model. YOLOv8 model is a deep learning model to recognize objects for computer vision. Then we implemented OpenCV with Streamlit to complete the simulation. Streamlit is a Python based library to build web applications. We used Streamlit to create the simulation interface which takes video sources and the video feeds are handled by OpenCV. The YOLOv8 object detection model is an object detection model for real-time applications. The model has three primary sections input, process and output. If the proposed system is applied in real life then it can reduce problems of fixed timed traffic signal system.

Keywords: Adaptive Traffic Control, Image Analysis, Real-Time Traffic Management, YOLOv7, YOLOv8 Object Detection, Traffic Density Analysis, Streamlit.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction

Urban traffic congestion is a significant global problem that results in massive economic losses, environmental degradation, and negative impacts on public health and overall well-being [29]. Green lights on vacant roads when nearby junctions are busy result in inefficient traffic management. For instance, traffic congestion in the United States alone costs about $87 billion annually due to wasted time and fuel [13]. Conventional traffic management systems, which rely on fixed-timing signals, exacerbate these issues in crowded urban regions by not adjusting to fluctuating traffic flow, leading to avoidable delays, increased automobile emissions, and elevated commuter stress [9]. Studies have shown that fixed-time signal systems can increase travel time by up to 20% during peak hours [5]. These systems operate based on predetermined schedules and lack adaptability to respond to real-time traffic variations caused by accidents, road maintenance, or daily fluctuations, resulting in ineffective traffic control. For example, fixed systems have been shown to contribute to a 15% increase in fuel consumption and a 20% increase in CO2 emissions [7]. In this project, we proposed and designed a traffic signal control system with the help of YOLOv8, OpenCV, and Streamlit. The YOLOv8 model was used to process traffic datas which was collected from roboflow. This model was trained for real-time vehicle detection so that the traffic signal timings can be adjusted accordingly. At first YOLOv7 model was used but later on we switched to YOLOv8 as it is the latest model which has better accuracy. We implemented OpenCV to handle essential image processing functions. The traffic signal control simulation interface was created using Streamlit which is an open-source Python library for building web applications.

1.2 Research Objectives

The primary objective is to propose a system to alleviate traffic congestion by implementing an adaptive traffic signal system.The goal of establishing a more reliable and efficient traffic signal system is to enhance overall traffic management. In the simulation, video feeds will be used by our system to assess the traffic intensity and adjust the signal timings in response. Furthermore, if the system is implemented in real life it will improve the synchronization of traffic signals. This simulation will show how the adaptive traffic signal system works under different circumstances of traffic flow. It will mimic real life by using videos to feed it data on traffic flow and then adjust the signals accordingly in the process.

1.3 Research Questions

- Do the proposed system prioritize effective traffic flow and ensure timely traffic light system?

- How much improvement in traffic flow can be achieved by implementing a dynamic traffic light control system?

- What is the level of effectiveness of the proposed system in accurately assessing traffic conditions using image processing and AI techniques?

Chapter 2

Literature Review

Traffic congestion has a substantial influence on urban efficiency and safety, especially due to the absence of real-time adaptation in conventional traffic signal systems. This literature review explores the use of image analysis and artificial intelligence (AI) into adaptive traffic signal control (ATSC) systems in order to improve traffic management. ATSC systems represent a transition from fixed timed signals to adaptable, smart controls that modify themselves according to up-to-date traffic information, using technologies such as Python, OpenCV, and YOLOv8 for detecting objects. These systems enhance the flow of cars and provide priority to emergency vehicles, so enhancing the efficiency of urban transportation. The talk examines significant progress in ATSC, with a specific emphasis on the use of image processing and AI to evaluate and react to traffic situations. Instances of research and practical applications illustrate the progression from simple automated systems to advanced solutions that include adaptive and predictive capabilities. This study attempts to demonstrate the progress made in traffic control technology, the improvements they have brought to operations, and the obstacles and future research areas in this rapidly evolving sector.

2.1 Overview of Traditional Traffic Control Systems

Traffic congestion is a growing concern in metropolitan areas worldwide, with significant implications for the mental health of commuters and residents. Conventional methods, such as fixed-time traffic signals and manual monitoring, are becoming progressively inadequate because they cannot adjust to real-time traffic situations. These approaches frequently result in inefficiencies, such as heightened congestion and pollution [34]. Over the past century, metropolitan traffic control systems have continuously evolved to address the increasing demands of urban transportation and the intricate objectives of public policy. These systems play a vital role in reducing traffic congestion, boosting economic efficiency, and enhancing road safety and environmental quality. With advancements in vehicle detection and communication technologies, traffic control systems have progressed from basic, fixed-time signals to sophisticated, adaptive solutions capable of responding to real-time traffic conditions. This paper explores both popular and lesser-known traffic control systems, highlighting their distinct features, strengths, and limitations. It assesses whether the current capabilities of these systems are sufficient to meet modern policy requirements, such as environmental sustainability and efficient urban mobility.

Additionally, the paper considers how future technological innovations, such as enhanced data accessibility and connectivity, may further transform traffic control. These advancements are expected to usher in a new era for traffic management, where data-driven solutions play an increasingly central role in urban transport planning [6]. A prototype traffic signal control system was developed, demonstrating the potential to reduce traffic congestion by nearly 60%, achieving 98% vehicle detection accuracy, enabling automatic signal control with cyclic order, offering manual override for emergencies, and providing road condition monitoring without the need for an operator [15]. Even in countries with well developed infrastructure the traffic jams persist in large cities and the requirement for advanced intelligent control system became critical; the proposed AI-based system for image processing of live feeds from cameras should increase car throughput by 32%, and more can be achieved if the AI is trained on real CCTV footage [32]. A new strategy to address the problem of congestion has been described in terms of new approaches to congestion detection as well as signal control through AI and these findings will show how traffic problems can be solved by AI technologies to improve flow conditions [40]. The paper entitled "Traffic Equilibrium with Responsive Traffic Control" gives derivation of a theory of traffic equilibrium that includes the relation of signal control policies and drivers' route choice decisions as twin faces of the traffic flow regulation problem. The main concern here is an iterative optimization assignment algorithm which computes equilibria under responsive control policies. This algorithm mimics the daily fluctuation process of the driver's route decisions as well as traffic signal timings [2].

2.2 Integration of Image Analysis and AI Techniques

The integration of image analysis and AI techniques in traffic signal control systems has garnered significant attention in recent years. The merging of advanced computer vision with deep learning technologies is exemplified in the work conducted by Aniruddha Tonge and others,titled "Traffic Rules Violation Detection using Deep Learning. This research makes a significant contribution by employing advanced techniques such as YOLO (You Only Look Once), Convolutional Neural Networks (CNN), Mask R-CNN,and Optical Character Recognition (OCR) to automatically detect traffic regulation breaches [17]. The use of image analysis and AI in the adaptive Traffic Signal Control Systems has enhanced the flow of traffic around cities. They demonstrate how the system delivers better results than the fixed-time signal techniques based on many parameters. This technology employs the use of artificial intelligence to help in the control of the traffic sign which include traffic lights so as to reduce the jam [38]. The findings of the study show that Internet of Things based traffic control systems provide plausible solutions of controlling traffic surveillance in real-time. With the help of these state-of-the-art technologies, the system is designed for detecting traffic jam and then adjusting traffic flow and offering smart transportation solutions for requirements of cities [38]. Reddy et al., employed machine learning solution to design an automated system for traffic rules violation detection. This detection system notes signal violations, categorises objects into vehicles, and extracts licence plate numbers for the subsequent actions. The suggested image processing algorithms are grey scaling and Gaussian blurring, background subtraction, binary thresholding, dilation for contour detection, and dilation for the improvement of image illumination and segmentation [21].

Wankhede and Bajaj used Particle Swarm Optimization (PSO) to develop a behavioral detection model for the purpose of monitoring and identifying traffic violations in Tier 2 cities in India, with a specific focus on driver social behavior. Their approach is based on the use of the soft computing technology known as Particle Swarm Optimization (PSO). Section on red light violation and mobile phone detection models implemented using Particle Swarm Optimization (PSO). The process of red light violation detection utilizes matching and Particle Swarm Optimization (PSO) to analyze video data and accurately detect instances of red light running [22]. The mobile phone detection model employs same techniques to detect driver mobile phone use, with the addition of PSO to enhance accuracy [22]. One of the critical image processing methods highlighted in paper "Detection of Traffic Rule Violations Using Machine Learning: An Analytical Review" is the Scale Invariant Feature Transform (SIFT), which is used for license plate recognition. SIFT is robust to scale and rotation, making it highly effective for identifying license plates under various conditions [28]. Moreover, for the solution of the problem of further detection the Hough Transform is used for detection of lanes which are important for the correct following of traffic lane rules by vehicles [28]. This integration is well explained in a study by Franklin and Mohana (2023) focusing on a study titled "Traffic Signal Violation Detection using Artificial Intelligence and Deep Learning". The primary usefulness of their study rests on the improved traffic violation detection system based on YOLOv3 (You Only Look Once, version 3). This method is characterized by high accuracy in counting instances of multiple traffic violations such as vehicle throughput and speed measurement [13]. Image analysis, also in conjunction with the use of artificial intelligence in control of traffic signals foster progress in traffic control in urban areas. This integration analyze and assess real time traffic data from image analysis using AI, specifically reinforcement learning [5]. The Cerebellar Model Articulation Controller (CMAC) retains and generalizes Q-estimates in single-agent and multiagent test beds; it demonstrates how Artificial Intelligence operates over vast data set and makes decision within short span of time. Such a machine learning method like deep learning and reinforcement learning hold the capacity to adapt to the changing traffic conditions. The approach proposed by Gao et al [7] effectively manages real-time traffic data. Experience replay and target networks are two sophisticated tools that improve both the stability and the performance of the system. By making use of experience replay, there are more number of encounters being provided to the system to learn, and thus reducing the impact of linked data samples. The target network technique improves learning through the use of the target network parameters which results in improved estimations of the target Q-values [7]. Researchers at Carnegie Mellon University implemented computer vision techniques for vehicle detection and tracking for denser predictions on traffic flow and congestion. In the same way, Deep Learning Algorithm has implemented for the prediction of the traffic flow by using the historical records with excellent accuracy. Reinforcement learning is also applied to control signal timings for traffic lights to reduce travelled distance and time with a success in the simulation environment [30].

The research conducted by Balci et al. [8] named "NIR Camera Based Mobile Seat Belt Enforcement System Using Deep Learning Techniques" is a good practical example of applying deep learning when it comes to object identification. This research also highlights the efficacy of combining image analysis with artificial intelligence. The use of Convolutional Neural Networks (CNN) for the goal of deep learning classification is a significant contribution to this study. More particularly, the utilization of RGB images received from a surveillance camera that was installed on a highway is highlighted as an essential addition. Because of its ability to independently and dynamically collect spatial hierarchies of information from input photographs, CNNs have proved to be indispensable in the field of image analysis. CNNs are well suited for complex object recognition tasks, such as the identification of drivers, seat belts, and parts of the dashboard located by side windows [8]. By combining IoT and image analysis, Asoba et al. [12] created an advanced traffic infraction control system. Their method tracks vehicles via RFID. As cars approach traffic signals, RFID scanners scan their unique IDs on RFID tags. This configuration enables effective vehicle recognition and monitoring, giving the system its basis. An AI-powered YOLO-based traffic management system by Channi et al. [20] reduced junction congestion. The system identifies cars using deep learning image processing. Traffic management requires YOLO (You Only Look Once) convolutional neural network object detection in real time,CCTV video analyzes congestion and informs traffic management. The prototype framework integrates various sensors and technologies to enhance traffic management capabilities. According to Mampilayil and Rahamathullah [11], an effective framework utilizing image analysis and artificial intelligence has been developed for the identification of three-wheeler vehicles and violations of one-way traffic regulations. This framework, which does not rely on conventional sensors, is considered cost-effective and straightforward to implement, making it well-suited for one-way traffic management. The proposed strategy incorporates advanced object detection and categorization techniques, including extraction and Gaussian Mixture Model (GMM) methods, both of which demonstrate robustness in complex traffic environments [11]. Additionally, the authors highlight that the elimination of shadow-related ambiguity through shadow removal contributes to enhancing the accuracy of object detection. Using a single-shot detector (SSD), de Goma et al. [14] identified cars in video recordings. Using the SSD model and CNN features, the system can identify red-light runners up to 100% and speeding crimes up to 92.1%. CNN architectures like MobileNet, Inception V2, and ResNet 50 extract features for reliable object recognition. The study also examines background registration techniques for moving automobiles, which increase vehicle identification. The system uses machine learning to predict red light infractions, showing how image analysis and AI work together [14].

The paper, "Artificial Intelligence-Based Adaptive Traffic Signal Control System: A Comprehensive Review:" explores a range of AI-based methods used in the development of adaptive traffic signal control (ATSC) systems. These methods are applied in both single-intersection (SI) and multiple-intersection (MI) settings. Advanced approaches, such as reinforcement learning (RL), deep reinforcement learning (DRL), and hybrid models, have shown considerable promise in improving traffic flow. Specifically, these AI techniques have proven effectiveness by reducing travel times and lowering down the emission probability, contributing to more efficient traffic management [36]. Artificial intelligence techniques were applied to optimize traffic signal timings on an urban road network using a two-stage approach, incorporating multilayer neural networks and Kohonen feature maps for training, and Cauchy machines along with genetic algorithms for optimization, which demonstrated superior results compared to conventional methods in minimizing delay time and stop frequencies [3]. The method of traffic control based on reinforcement learning, namely deep Q-learning, was introduced for the case of partial vehicle detection, and it was shown that the usage of such approach provides high efficient traffic management under partially detected situations, including DSRC technology [23]. A move to intermodal transportation systems that include traditional, connected, and autonomous vehicles requires the use of data-driven, feedback-controlled control systems that are precise and intelligent to adapt to the dynamic traffic systems being expected in the future [10]. "Adaptive Look-ahead Optimization of Traffic Signals" paper provides a new branch-and-bound optimization framework for efficiently finding delay-optimizing signal policies; the method is found to be tractable when optimizing signals according to several criteria such as the number of stops; further, the performance of the search algorithm is established through simulation while investigating the implementation of active as well as passive bus priority using decentralized controllers and network-wide optimization for arterial networks [4]. To evaluate DQF, therefore, the proposed mathematical model undergoes extensive simulation analysis to test for both efficiency and comparison with existing algorithms based on indices such as average delay, offered throughput and emissions. According to the discoveries, DQF is much more effective than basic algorithms in reducing the emission level while maintaining effective traffic flow, thus helping to create environment-friendly urban traffic control tactics employing new approaches based on artificial intelligence [37]. HEIMDALL system is an AI based video surveillance system for the sensing the environment of urban cities, integration and coordination of communication along with training of AI models The system employs YOLO system for tracking and Faster R-CNN for identifying anomalies [19]. To establish CTSC as a centralised system, a study was carried out to demonstrate the use of comprehensive traffic information in the COordinated management of inter-TSs, hence enhancing CTSC operation and outcompeting traditional systems which led to decreased queue lengths and travel time despite improving vehicle flow on arterial roads than the systems which lead to delays on collector roads [16].

Malfunctioning and risky driving by some drivers often leads to car accidents at roundabouts The study "AI-Based Adaptive Signaling for Traffic Control Around Roundabouts" holds new techniques for minimizing such occurrences. The proposed artificial intelligence based roundabout traffic control system (ARTCS) was able to identify left approaching traffic through the use of a camera and a trained neural network. The paper titled, "AI-Based Dynamic Traffic Signal Control Systems for Autonomous Vehicle Navigation" establishes that DRL can successfully learn traffic control policies that are adaptive in the case of AV navigation and perform better than rule-based methods concerning real-time traffic conditions. This advancement comes in the same line with continuous developments of active traffic control strategies for urban traffic conditions that have largely been in the development phase [41]. In the recent past, various deep learning techniques such as SSD, R-CNN and YOLO have been used in detection of vehicles. From these, the YOLO models have been proven to be the fastest with better precision and are therefore suitable to be used in real time operations. These are extended in this thesis by using YOLOv8 for vehicle detection and of making fast signal changes based on traffic situations [33].

2.3 Case Studies and Real-World Applications

Image analysis and artificial intelligence in adaptive traffic signal control have enhanced the urban traffic control especially under real traffic environment [25]. Maadi and colleagues used reinforcement learning to improve a real-time adaptive traffic light control system. The rationale of the project was to reduce traffic congestion in cities and to improve mobility with linked and autonomous cars. Signal planning was optimized, and a speed guidance system for fully autonomous vehicles was introduced using reinforcement learning. With these technologies integrated, then the system may change the traffic signal timings in real time, in a bid to shorten the queue length and cross standstill time. In urban contexts, this method was more effective than the fixed-timing and the actuated control [25]. The work established that the method helps decrease mean time loss of stop and waiting, not only in high traffic condition but in diverse other traffic conditions. This strategy optimised traffic and proved that AI and image analysis technologies can help in real-time adaptive traffic signal management. The versatility, effectiveness as well as the possibilities offered by adaptive traffic signal management systems can be perhaps best observed with real life examples and case studies these supply a great deal of valuable information [5]. The practical application of Q-learning for solving congestion problem was established by Abdulhai and his colleagues in 2003. As a first step, the study was done on single junction and then it was generalized across multiagent system [17]. The outcomes show that the accuracy of Q-learning algorithms is promising when it comes to interpreting traffic circumstances in the demonstration process, and applying exploration strategies is vital when seeking optimal outcomes. In addition, the authors find that the proposed method of extending the base system by incorporating weighted global incentives can potentially improve the system's capability to converge toward an efficient policy faster. This discovery shows the effectiveness of using reinforcement learning in the practical field of traffic control [5]. To identify opportunities for synchronizing neighboring intersections according to the observed traffic behavior, the authors gathered observational data on vehicular platoons in four sites in London towards controlling signal phasing to eliminate delay; the authors noted that the spread of platoons as they go downstream can be used to determine appropriate time offsets with optimal offset expressed as a linear function of the distance downstream from the originating traffic signal [1]. One good example is the use of artificial intelligence in traffic control during pilgrimage in Mecca known as Hajj. These millions of pilgrims cause enormous logistical problems with traffic and crowds each year at this event. Researchers Gazzawe and Albahar explain how smart cameras and AI algorithms are used to control the traffic flow of the approximately 250000 vehicles. These systems utilize machine learning for real-time data analytics, enabling the prediction and alleviation of traffic congestion by suggesting alternative routes and improving vehicular flow. The case study conducted in Makkah during the Hajj pilgrimage demonstrates the effectiveness of AI technology in handling traffic congestion during large-scale events. This study demonstrates the positive impact of intelligent information and communication technology (ICT) [26] infrastructure and the implementation of a smart city model on the management of crowds and traffic [29]. They used artificial intelligence and machine learning techniques to address the unique challenges associated with the Hajj pilgrimage, which attracts millions of pilgrims. This study included a combination of qualitative and quantitative research methodologies, including interviews, focus group discussions, and observations, to gather valuable insights from various parties involved in Hajj crowd management. Individuals with talents relevant to the research were chosen via targeted sampling [29]. The study titled "A Gain With No Pain: The research paper entitled 'Intelligent Street Traffic Signal Configuration for Emergency Vehicles: Possibility of Implementation' discovered that the application of deep reinforcement learning procedure for traffic signaling system results to a swift response of emergency service while at the same time reducing the disruption of conflicting rush in other directions [24].

2.4 Current Technological Challenges and Limitations

Even though we have come so far with technology yet several challenges and limitations still exists in adoption of integrated image analysis and AI techniques in traffic signal control systems. Thus, the key challenge that arises because of deep learning models like YOLOv3 that require graphic card computation and other specialized equipment are required is a major barrier particularly in regions with low technological advancement [13]. Also, the need to obtain annotated data for training and testing the model can be considered as the other big challenge. The combination of image analysis and artificial intelligence approaches in traffic signal control systems has been developed thoroughly, but it still presented some issues. The processing of data has to be real-time because that is how the system manages large traffic loads. Independent features such as the accurate detection and classification of objects on the roadway as well as their lighting and weather variations are also important. While runtime optimizing is needed for the program's scalability and for avoiding inefficiencies, meeting computational needs with hardware limitations is still a problem. Using the penalty point system to add difficulties and including Optical Character Recognition for the number plate increases the overall complexity and should be timely considered [7].

In future research works further extensions of this adaptive image processing system should be made, including the integration of more sophisticated artificial intelligence as well as more deep and refined computer vision algorithms for traffic control applications. Another considerable issue is precise identification and recognition of objects located on streets, and their classification when the situation is foggy and at night. This is due to the problem of over fitting, as well as nuanced differences in the specific kind of inference that neural network prediction models can cause fidelity losses in range of applicability [14]. Further, it is required to create stronger algorithms that would be tolerant to certain number of conditions for more accurate detection [21]. The techniques like YOLO, Mask R-CNN, and CNNs take a lot of computational power, which may hinder with the system's scalability, and therefore with its availability, particularly in the less technologically developed countries [17]. Another major problem is the assurance of item identification and subsequent tracking. The study underscores the need to position the camera effectively to minimize cases of the object overlap in the images, that may otherwise lead to false positives for detection. In addition, the performance in detecting might be influenced by different illumination conditions where it is challenging to maintain dependable result [38]. In essence, it is possible to register the issues and challenges associated with the related problems and limits, thus focusing on the need to address all the obstacles in the context of proper implementation of image analysis and artificial intelligence methods with regard to adaptive traffic signal management [28]. This is because the adoption of these approaches offers great benefits since it is a multiplex of the single strategies. Despite the fact that the proposed system includes acceptable accuracy and recall rates, there is insufficient experimental testing to suggest that the system is viable with regard to numerous traffic conditions. The environment during the detection process can be challenging, so it is possible that image-based detection systems will be negatively impacted by the weather, including rain, fog or snow. This is something which might occur. There is the likelihood that some conditions may have a detrimental impact on image quality and the task of sorting items using artificial intelligence algorithms will become much more challenging. Controlling for high performance has remained to be a key issue with respect to weather extremity, which demands further innovation [8]. The identification of the transition probabilities of intersection states is difficult due to the nature of traffic situations. Traffic is dependent on several factors such as weather conditions, accidents, construction works and consequently, modeling intersection behaviors is a challenge [7]. There is also a need to pay much attention to carefully managing this to avoid adverse effects on traffic flow or some other drive factors due to activities of this system. This underscores the importance of future studies and advancement to enhance the robustness, as well as applicability of sophisticated adaptive traffic signal control systems [5]. While experimental results reveal that integrating image analysis with AI for traffic management can work and that the framework provides a good start, many questions and challenges are still present [9]. Another limitation peculiar to most intelligent computing methods is the inability to learn on its own.A phase optimization module needs improvement in flexibility to make it flexible in use, although it poses high computational problem to the network as in the use of neural networks and fuzzy mode [9]. Because of low economic status, it can be costly applying high density of cameras and application of AI system in such Nation as Bangladesh. Hence, financial factor must be carefully analysed when deciding on adoption of one or another advanced technology [28]. The goal of the project is to reduce time delay in traffic signal control in real-time by incorporating AI with IoT devices [39]. It involves the computational use of Artificial intelligence to assess and interpret picture information with a view of addressing present traffic conditions. In addition, present traffic data are used with machine learning algorithms to improve the accuracy of the system's projections. This technology blends the use of image analysis together with machine learning to increase traffic signals and reduce traffic jam. This serves to illustrate the honoring synergistic interaction between artificial intelligence and image analysis in the construction of complex Traffic control systems that are able to adapt to Urban traffic functions in real-time [39].

The paper on "Detecting and Counting Different Vehicles in Real-Time Traffic Signals Using Deep Learning" points out that YOLO is a fully convolutional network with 75 convolutional layers; skip connections; and upsampling layers. It uses a convolution layer that learns features having a stride of 2 to reduce the size of feature maps maintaining low-level features. While YOLO does not depend on the size of images as input, fixed sizes of height and width of the images are helpful during batches processing. In this work, YOLOv3 is used with Nvidia Tesla K80 as a GPU in order to improve computational ability [31]. The paper entitled Automatic Planning for Generating and Simulating Traffic Signal Schemes investigates how AI approaches can be used for traffic management with a focus on signal control in urban environments. It illustrates that model-based approaches, including automated planning, can handle unpredictable traffic conditions and common traffic patterns in parallel. The work discussed here brings out and experiments traffic signal control strategies by feeding the low level sensor data collected from the Kirklees region in United Kingdom into its generated framework [27]. The study titled "Intelligent Vehicle Pedestrian Light (IVPL): In the paper titled "A Deep Reinforcement Learning Approach for Traffic Signal Control" the authors propose a deep reinforcement learning (RL) adaptive traffic signal model that shows high efficiency in the control task for both vehicle and pedestrian traffic streams. From the findings of the study it is clear that the IVPL method developed here is intended to provide fair distribution of green time to vehicles and pedestrians with the objective of minimizing the total user time instead of merely the vehicle time. The model uses an extended reward function that captures delays induced by interactions between different road users, and assesses the effects both of pedestrian jaywalking and interruptions to traffic flow in different conditions. Computer simulation discussed in this paper utilizing real traffic signal parameters confirmed the superior performance of the IVPL model over fully actuated traffic signal system in terms of minimizing intersection delays effect [35]. Firstly, it is imperative to note that this study is largely a systematic review, which by design means that it cannot incorporate all the various forms of intricate analysis, which most of the papers reviewed offered diagrams and other conceptual models and not compendiums of intricate quantities, so the best method – a meta-analysis – is impossible to perform. With only 35 papers under review, the authors noted that there is paucity of research in some of the critical areas regarding AI based ITS and the need for a much larger study in this fast growing field. Literature review on actual design, implementation and assessment of traffic parameters with real-application is scant with few specific cases matching with current frameworks/models in operational smart cities. The review did not devote enough attention to the evolution of the traffic control systems such as dynamic traffic light which in the model by Nellore and Hancke (2016) was discussed in passing without further discussion of the idea nor other sources. Furthermore, the topics on adaptive traffic control systems and other related discussions could not receive further expounded analyses due to the paper length restriction of journal publication [18].

Chapter 3

Methodology

3.1 Description of the model

YOLOv8 is one of the best model for computer vision right now. It is known for its fast architecture, efficiency and accuracy. YOLO had many versions, YOLOv8 is the latest version. YOLO8 makes it ideal for real time processing where speed is of the essence. One of the most profound applications of YOLOv8 is demonstrated in controlling traffic lights, and its reliability cannot be overstated. This allows the system to make optimum decisions in the timing of the signals depending on present traffic-update patterns. The fast image processing is desirable in order to adapt rapidly to new conditions in terms of traffic flow that is especially significant in densely-populated areas wherein the traffic intensity can dramatically change.

3.1.1 Architecture Overview

To quickly and effectively detect objects, YOLOv8 follows the neural network called YOLO architecture, and it is advanced. The model's structure can be visualized as a flow diagram, comprising three primary sections: input, process, and output. .
Input: The architecture takes input data in a form of images. Which later has to be preprocessed to make them standard and favorable for object detection.

Process: The Head and the Backbone are the two primary processes that drive YOLOv8. The P1 through P5 layers that create out the Backbone are assigned for feature extraction. Each layer somewhat covers different levels of depths where lower layers try to identify edges and textures of a given image while higher layers try to identify patterns. The Head that is aligned with the Backbone work in parallel to this process, and they interpret the features that have been retrieved as well as guess works. This is accompanied by extra parts named as 'Concat' and 'Detect' to indicate feature concatenation and object identification processes, correspondingly. The layers of the Head, which are likewise designated P2 through P5, reflect the Backbone's layers and create a link between object detection and feature extraction.

Output: The culmination of this process is the output, represented by two scales: 13x13x255 and 26x26x255. These dimensions are actually the resolution and depth of the prediction maps that include bounding boxes and their class probabilities.

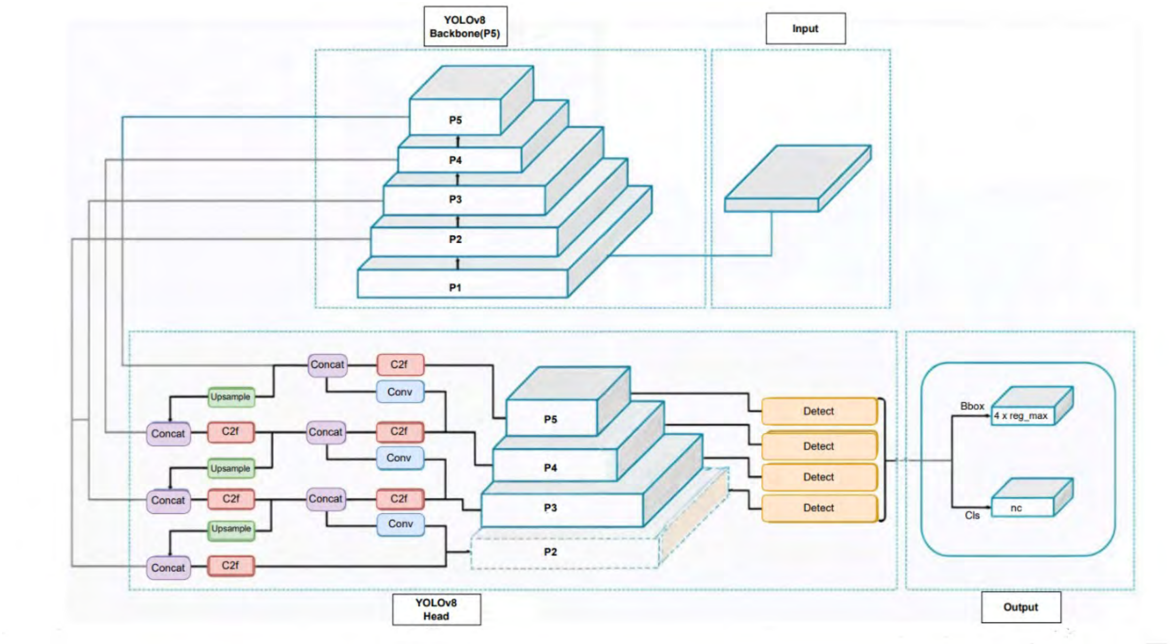

Figure 3.1: YOLOv8 architecture

3.2 Description of the data

The dataset utilized in this study was manually assembled from CCTV footage and included a variety of urban traffic components. Recordings include different types of vehicles like automobiles, bicycles, trucks, and convertible vehicles, etc. It helped the model to identify diverse road patterns and types of vehicles. Specific preprocessing methods, for example, data enhancement, provide differences that emulating real-world conditions like variations in light, climate, or partial occluding. This increases the model's stability and its ability to perform efficiently in other conditions with relative ease.

Dataset and Preprocessing

Dataset quality and preparedness are critical determinants of the model being developed. Proper labeling means that the model will recognize the right boundaries of objects and their type; preprocessing adds to better generalization as to new cases.

Labeling Process: The labeling process was done by using CVAT, It is a computer vision annotation tool. Every object was included in a bounding box and labeled with an appropriate name. This part is very important for the model's capability to pinpoint and recognize specific objects. .

Data Augmentation Process: The images that was used were changed and modified in a way that the model could encounter in real life scenarios. Camera footage of CCTV are not always crystal clear due to weather conditions or lighting's. So, it was enhanced by using techniques like random cropping, rotation, and flipping, variations were introduced.

hsv_h=0.015, # Slight hue shift to simulate color change at night

hsv_s=0.7, # Increase saturation

hsv_v=0.2, # Adjust value (brightness)

flipud=0.5, # Vertical flip

scale=0.5, # Scale the image

translate=0.2; # Translate the image for robustness

Improving the flexibility of Models Using Parameterized Augmentation

As such, we can see that applying specific custom augmentation firmly enhances the performance of YOLOv8 in the context of various visual settings. These parameters mimic real-world environment changes to make the model optimize the object detection accuracy. In the model, we have added the GaussNoise tried to mimic the real-world scenario where the camera detects a grainy nose using 30 percent probability. In the real world, scenes have some amount of noise and hence distortion noise helps the model to familiarize it with images that have some degree of impairment. Then we have added Hue shift (hsv_h=0.015). This small modification makes small color variations, emulating changes in lighting conditions for example from daytime to nighttime to help the model perform objects in a different light. We also have increased the saturation (hsv_s=0.7). This increased saturation makes the subsequent color appear even more conspicuous, which helps the model to learn objects with different intensities of hues shaded from thin to thick. In the brightness adjustment (hsv_v=0.2) turning a little brightness gives the model a different level of light which makes its reading accurate both in bright areas and dark areas. We have also used the flipped parameter which makes the images flipped by 50 percent handling the images with objects either mirrored or upside down. For the size variability as well as testing the ability to mimic camera zoom we used random scaling to make the image larger or smaller to train the model. And lastly, we have added translation changing the object coordinates by 20% making the model recognize objects in other frame locations across various scenes.

Normalization: The images were kept to similar uniformed size of 640 x 640 px. This helped to make the training stable and efficient. This process is important to mitigate problems with image sizes and scales, If all images are consistent then the model is efficiently trained.

3.3 The impact of preprocessing on the performance of the model

The goal of such preprocessing steps was to ensure that the input set is robust and transformable enough for training purposes of the YOLOv8 model. Besides, the augmented and normalized dataset also used in this research was purposely to imitating the real-world situation, which helps the model to perform well on the new samples that it had never encountered during data training. This generalization is very important in real-time applications to allow the model to adapt quickly to the rapidly changing conditions as might be observed in traffic signal control.

The detailed and very careful way of naming and with a lot of pre processing, contributed to a proper foundation for YOLOv8 model getting very high efficiency and reliability for the tasks of object detection. Thus, the problems that may occur during the usage of the dataset for training and testing were considered beforehand, as well as differences of traffic situations in different cities.

Chapter 4

Evaluation of the Model

4.1 Performance Metrics

To evaluate the YOLOv8 model's performance and reliability during the training phase, a comprehensive set of metrics and visualizations were employed. These tools are instrumental in identifying the model's strengths and areas for improvement.

4.1.1 Normalized Confusion Matrix

The normalized confusion matrix reveals that the model excels at identifying 'background' with high accuracy, while the 'bicycle' and 'motorcycle' categories show some confusion, likely due to visual similarities. Detection accuracy for 'bus', 'car', and 'person' is moderate. Overall, the matrix suggests a proficient model, especially in discerning heavily represented classes and distinguishing them from the 'background', a crucial aspect for traffic analysis applications.

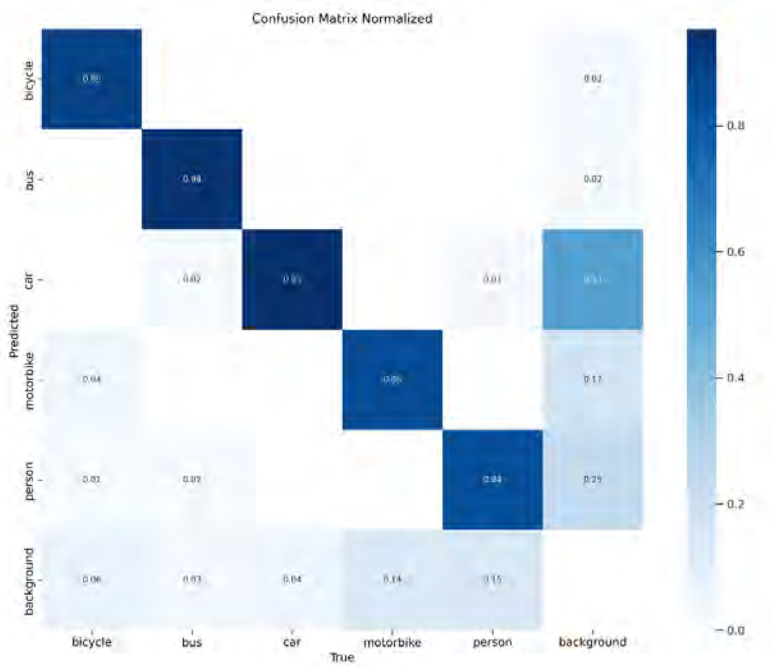

Figure 4.1: Normalized Confusion Matrix

4.1.2 F1-Confidence Curve

The model's performance in different classes is displayed on the "F1-Confidence Curve" graph, which has an ideal F1 score of roughly 0.87 at a confidence level of 0.403. This peak suggests that the model is accurately calibrated because it shows a great balance between precision and recall. Even while the curves for each class—bicycles, buses, automobiles, motorbikes, and people—are different from one another, they all show the same general pattern of increasing to a peak and then decreasing, which is typical for F1 scores as confidence thresholds rise. This illustrates how consistently accurate the model is at detecting objects across several traffic-related categories.

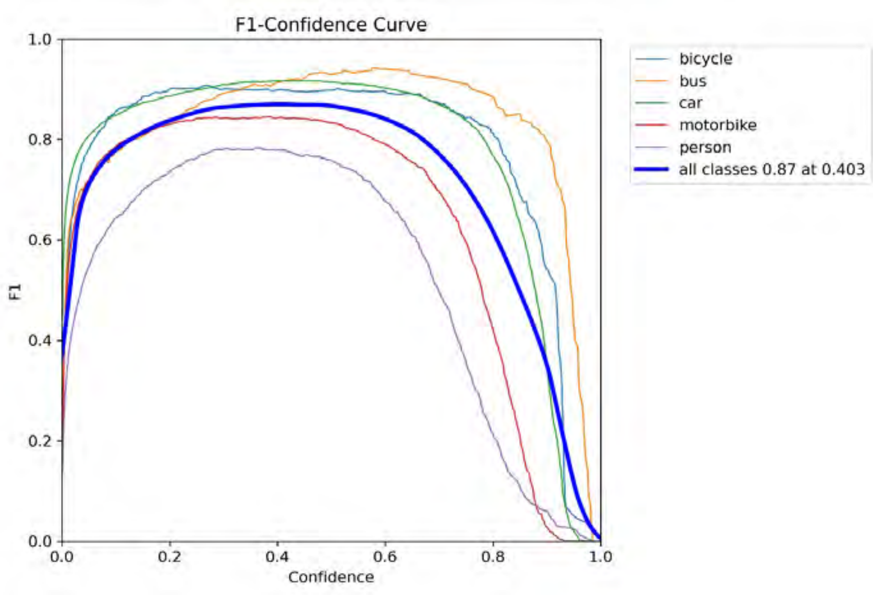

Figure 4.2: F1-Confidence Curve

4.1.3 Precision-recall curve

The model performs exceptionally well in a variety of classes, as evidenced by the accuracy-recall curve; average precision (AP) scores range from 0.833 for "person" to 0.971 for "bus." With a mean Average Precision (mAP) of 0.915 at an Intersection over Union (IoU) of 0.5, object detection accuracy is high overall. This implies that the model is very good at detecting pertinent items with a low false positive rate, which is important for traffic signal management applications where accuracy is critical.

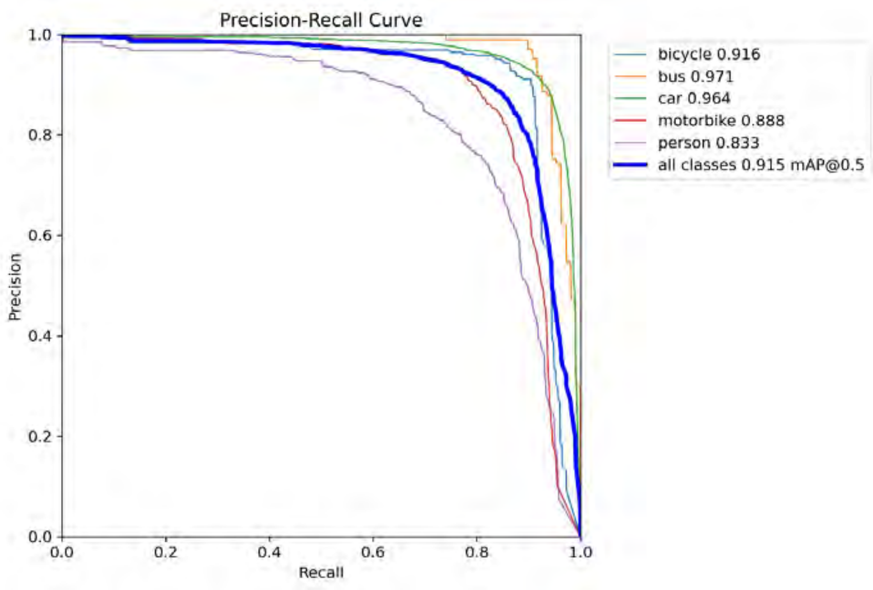

Figure 4.3: Precision-recall curve

4.1.4 Precision-Confidence Curve

For the classes of bus, bicycle, person, vehicle, and motorcycle, the Precision-Confidence Curve graph shows that the model retains a high degree of precision across all confidence levels. Remarkably, the 'all classes' curve stays precisely on target the entire time, suggesting that the model predicts with extraordinary accuracy, independent of the confidence level. This consistency indicates a well-trained model that ensures strong performance in a variety of settings and minimal false positives when utilized for object recognition in traffic control systems.

4.1.5 YOLOv7

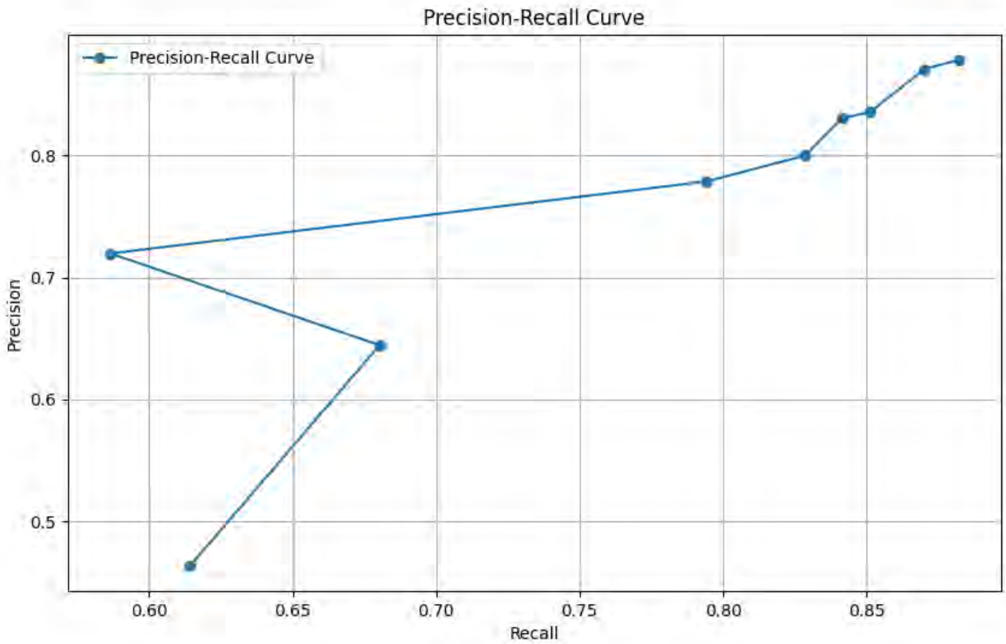

Figure 4.4: Your caption for the YOLO V7 precision-recall curve.

The graph of Precision-Recall curve for the vehicle detection YOLOv7 model shows the rate of increase in precision as well as recall throughout the detection process. The model also show a more selective behavior at the lower right quadrant, which corresponds to the initial point of the curve with recall about 0,6 and precision about 0,7. Here, approximately 70 percent of the identified vehicles are valid and true positives, while it checks only approximately 60 percent of all the actual vehicles out there. This means that the model is somewhat selective with its detections to minimize on false predictions in an effort to improve on the accuracy of the detections. At the point where recall is slightly higher to 0.62, precision reduces drastically to 0.50. This fall suggests that while the model tries to recognize more vehicles, it brings in higher false positives component thus lowering the total system efficiency. This phase might be caused by changes in the detection thresholds where carrying capacity is traded against recall at the expense of precision. Still, precision improves as recall starts to level up to 0.67, to about 0.60. This improvement reveals a relative gain in parameter setting that gains more instances of vehicles while decreasing the ratio of wrong detection. The model is now less conservative but remains realistic to a reasonable degree. As more recalls are still performed the precision of the results increases and reaches the 0.85 level which is still higher than a 0.75 to 0.85 recall level. The model is at best balanced for this stage, it catches most of the vehicles while the precision is high. The upward trend observed here indicates that architects have been tuning every factor, which defines an actual model, such as the confidence threshold and the Intersecting over Union (IoU threshold), to capture more instances with the least possible trade off on the precision level.

4.1.6 Comparison Metrics for the YOLOv7 and YOLOv8 Model (Without Augmentation):

Metrics	YOLOv8	YOLOv7
Precision	0.90254	0.8702
Recall	0.84165	0.8707
F1-Score	0.8710	0.8704
Map50	0.91456	0.9130

Table 4.1: Benchmarking YOLOv7 and YOLOv8 Model (Without Augmentation)

Precision:

YOLOv8: The precision score obtained is 0.90254. This means that out of all the cases that the model tagged as positive (which includes true positive and false positive) 90.25% were true positive. This high precision implies that YOLOv8 will rarely give false alarms that is, identify signs of an object when it is not – and therefore is more effective in alerting its user when it picks signs of an object's presence. YOLOv7: The precision accuracy is 0.8702, though it is a little lower than YOLO standardized in YOLOv8. This suggests that about 87.02% of this positive detection is accurate though still strong. This indicates that YOLOv8 is somewhat less prone to false detections than YOLOv7 is. Judging by both, YOLOv8 has a 3.2% more improvement in precision compared to YOLOv7. This difference could be useful most of all in those cases where the number of false detections is highly undesirable, for instance, in the case with the automatic traffic control systems, where the improper identification of vehicles can lead to improper actions.

Recall:

YOLOv8: The recall score is 0.84165 meaning that more than 84.17% of the actual positive samples are accurately recalled by the corresponding criteria. It can be seen that the more objects it detects, the larger the number of actual instances that it is likely to have missed. YOLOv7: By comparing the recall metrics of the detectors, YOLOv7 has a marginally better sensitivity compared to that of YOLOv8 whose sensitivity is 0.8707. This implies YOLOv7 recognizes a larger number of true positives with only 87.07% implying a lower False Negative rate suggesting that our model is better at not missing objects that are in the images. Judging by this The paper shows that the recall of YOLOv7 is higher by about 2.9 percent, which means that, in the case of identification of a greater number of objects, YOLOv7 is more efficient. This characteristic makes them suitable where an object that may be out of sight should be quickly located, for example, in real-time security surveillance or traffic analysis where all vehicles must be seen.

F1-Score:

YOLOv8: The F1-score is 0.8710, which is the balanced version of the precision coefficient and the effectiveness coefficient. This score implies the model has a reasonable trade-off for both completeness and accuracy of object detection. YOLOv7: F-measure is 0.8704, which is equal to the YOLOv8 algorithm with a small difference. Same way, it shows the ratio between precision and recall whereby YOLOv7 also ± profiles a good recall-precision measure which also means a good tangle between false positive errors and missed errors. By comparing both F1-score comparison between YOLOv8 and YOLOv7 is also very close with a difference of 0.0006. This implies that both models give almost similar overall performance with respect to the measurement of precision and recall. In a practical sense, both models seem to be effective in equal measure in terms of reliability, nonetheless, the effectiveness in the precision and recall skills of each separately may determine their appropriateness in specific applications.

mAP@0.5 (Mean Average Precision at IoU=0.5):

YOLOv8: The mAP@0.5 is 0.91456, meaning the model gains a mean average precision of averagely 91.46% over ./844 words er at the moment of the Intersection over Union (IoU) threshold of 0.5. This shows how well the model is in predicting the ground truth box compared to the size and location of the boxes it has generated. YOLOv7: However, the mAP@0.5 of this model is slightly less – 0.9130 which is slightly less than the YOLOv8 model. It shows similar performances for discovering objects of two or more classes with a similar degree of accuracy. So from the both comparison, we observe that the value of mAP@0.5 is much closer, with YOLOv8 slightly ahead of YOLOv3 by 0.15%. This means that in terms of assessments in the detection of objects using different methods with varied IoU thresholds, the two models are quite efficient. However, it can be seen that YOLOv8 might have a slight advantage for those applications that would much rather deal with precise detection.

Overall analysis:

YOLOv8 and YOLOv7 are, respectively, fast and very fast object detection models suitable for different tasks due to their features. As for the precision, it is higher (0.90254) and the mAP@0.5 is a bit better (0.91456), which is more useful in cases where false positives can be a problem, especially in the utilizing of highly accurate detections. On the other hand, the recall of YOLOv7 is higher (0.8707), which means it will detect a range number of objects and may be preferable to not detect any instance at all, as in surveillance. Thus, the F1 scores given in both models varied only in two decimal points, which means that the two models were balanced between precision and recall. In the end, it all comes down to what kind of application is in use, if the requirement is for a program that must detect something with high sensitivity then linear search would be used, whereas if the requirement is that the program must generate results with high accuracy then binary search would be used.

4.1.7 YOLOv8 Model With Augmentation vs Without Augmentation:

Metrics	YOLOv8 (Augmentation)	YOLOv8 (Without Augmentation)
Precision	0.87019	0.90254
Recall	0.81516	0.84165
F1-Score	0.8418	0.8710
Map50	0.88211	0.91456

Table 4.2: Benchmarking YOLOv8 with and without augmentation

Precision : Precision of the model trained without augmentation is 0.90254 while the one trained with augmentation is 0.87019. Precision reflects how accurately the model indicates the presence of objects. The lack of change with augmentation suggests that, while the model was more precise at a higher level, it was doing so conservatively and had fewer false positives. This may have been due to augmentation adding variations that made the model slightly less confident of its detections.

Recall: The model without augmentation got 0.84165 and the model with augmentation got 0.81516. Recall analyses whether the model captured all the existing objects in the dataset and recognized all objects. Judging by the numbers higher recall without augmentation can identify more actual objects since there are fewer variations in the training data. This means that the augmentation applied could have included noise or new patterns which in general has slightly affected the recognition capability of the model for every object of interest.

The F1-Score: The F1-Score which equals the harmonic mean between Precision and Recall, adds to these findings. While the model trained with augmentation has an F1-Score of 0.8418 the F1-Score for the model trained without any augmentation is 0.8710. This metric balances Precision and Recall to provide a single measure of overall performance. The calculation for the F1-Score was verified as accurate for both cases, confirming that the model without augmentation outperforms the augmented model in terms of balanced object detection.

mAP50 (mean average precision at 50% Interesection over Union) : 0.91456 where augmentation is not used and is 0.88211 where augmentation is used. The mAP50 score measures how accurate the model is when it comes to positioning and shape of objects they are attempting to detect. The mAP50 is slightly higher for the non augmented model indicating that the model had slightly better object localization as opposed to the model that was augmented but contained localization variations. It also suggests that the initial dataset might have had sufficient variability, and the attracted augmentation slightly deteriorated the performance due to unknown patterns.

4.1.8 Results of YOLOv8 Model

The spreadsheet image shows different metrics that are tracked across several epochs when the machine-learning model is being trained. These measures include validation losses for losses like 'rpn_cls_loss', 'rpn_bbox_loss', 'roi_cls_loss', 'roi_bbox_loss', and 'total_losses'. Based on observable patterns, it seems that losses diminish with time, indicating that the model is learning and convergent. To avoid overfitting, it appears that the model is being assessed for generalization when both training and validation losses are present. Effective learning is shown by the steady decline in "total losses" throughout epochs, and strong generalization skills are suggested by the equilibrium between training and validation losses. As it suggests the model is learning to recognize and classify things effectively without memorizing the training data, this is a desired outcome in model training. All things considered, the findings point to a successful model that is getting better at prediction over time.

epoch	train/box_loss	train/cls_loss	train/dfl_loss	metrics/precisior	metrics/recall(B)	metrics/mAP50(I	metrics/mAP50-!	val/box_loss	val/cls_loss	val/dfl_loss	lr/pg0	lr/pg1	lr/pg2
1	1.4208	1.6675	1.1063	0.69697	0.60687	0.65572	0.41165	1.1628	1.0122	1.0195	0.00036931	0.00036931	0.00036931
2	1.2794	1.0341	1.0439	0.73238	0.68711	0.71211	0.45184	1.156	0.88209	1.0215	0.00070303	0.00070303	0.00070303
3	1.2614	0.9447	1.0366	0.755	0.70466	0.75705	0.50024	1.118	0.79107	0.9965	0.0010001	0.0010001	0.0010001
4	1.2238	0.88102	1.0255	0.77499	0.71983	0.78086	0.50844	1.1014	0.77129	0.98299	0.00094602	0.00094602	0.00094602
5	1.1946	0.82397	1.0126	0.76274	0.74334	0.79615	0.53871	1.0864	0.71897	0.98565	0.00089102	0.00089102	0.00089102
6	1.1704	0.78605	1.0068	0.8219	0.74426	0.82088	0.55483	1.0687	0.67642	0.9824	0.00083603	0.00083603	0.00083603
7	1.1597	0.7609	0.99911	0.82183	0.77174	0.84652	0.57114	1.0587	0.68092	0.97604	0.00078103	0.00078103	0.00078103
8	1.1339	0.72981	0.99006	0.81637	0.79628	0.85257	0.58979	1.0357	0.64035	0.96927	0.00072604	0.00072604	0.00072604
9	1.111	0.71089	0.98123	0.82033	0.80855	0.85343	0.593	1.0221	0.64284	0.9606	0.00067104	0.00067104	0.00067104
10	1.1051	0.69784	0.97968	0.82376	0.79899	0.86069	0.59671	1.0196	0.63057	0.95749	0.00061605	0.00061605	0.00061605
11	1.0709	0.66006	0.97487	0.8539	0.79979	0.87249	0.61262	1.0035	0.6055	0.95076	0.00056105	0.00056105	0.00056105
12	1.0454	0.63341	0.96588	0.86377	0.81829	0.87778	0.6181	0.99178	0.6009	0.9498	0.00050606	0.00050606	0.00050606
13	1.0263	0.6142	0.95791	0.87829	0.82133	0.88967	0.63043	0.98774	0.58038	0.94978	0.00045107	0.00045107	0.00045107
14	1.0138	0.59756	0.95285	0.84715	0.83757	0.8933	0.6378	0.96852	0.57508	0.94069	0.00039607	0.00039607	0.00039607
15	0.99287	0.58101	0.94463	0.89353	0.82838	0.89843	0.64669	0.96358	0.56055	0.93922	0.00034108	0.00034108	0.00034108
16	0.97724	0.57025	0.94023	0.89916	0.83089	0.90076	0.65253	0.95142	0.54855	0.93136	0.00028608	0.00028608	0.00028608
17	0.95734	0.55204	0.93528	0.88166	0.83482	0.89705	0.65419	0.94286	0.54686	0.93187	0.00023109	0.00023109	0.00023109
18	0.94323	0.53889	0.92936	0.88676	0.85906	0.91016	0.66603	0.93189	0.53219	0.92715	0.00017609	0.00017609	0.00017609
19	0.93025	0.53003	0.92508	0.89861	0.85173	0.9145	0.67256	0.92109	0.52282	0.92505	0.0001211	0.0001211	0.0001211
20	0.91439	0.51943	0.92091	0.90254	0.84165	0.91456	0.67649	0.91332	0.51886	0.9234	6.61E-05	6.61E-05	6.61E-05

Figure 4.5: Results

Chapter 5

Implementation of the Traffic Signal Management System

The implementation of the Traffic Signal Management System is a crucial part of this project, demonstrating the operational aspect of the proposed adaptive traffic management system. The application is web-based application built with Streamlit. The simulation is designed to dynamically adjust traffic lights based on real-time vehicle detection data obtained from video inputs, simulating a traffic management system for an urban setting.

5.1 System Architecture of the application

The traffic control system operates as a simulation tool that processes multiple video inputs to monitor the density of vehicles at various intersections. The system later changes traffic light's indication from red, through yellow and to green based on the number of cars identified in each video stream. This mechanism reflects a real world adaptive traffic control system designed to control the traffic congestion by adjusting the signal based on current traffic conditions.

5.1.1 Traffic Light Simulation Pseudo code

Algorithm 1 Adaptive Traffic Signal Control

```
Initialize signal states for each road with:
   Red light
   Zero transition count
   Vehicle count buffer for the last 30 frames
Wait until the start button is pressed
while stop button is not pressed do
   for each road do
      if the road has a green light then
         if maximum green light duration has passed then
            Change the light to yellow
            Display yellow light for the specified duration
            Change the light to red
            Move to the next road in line
         end if
      else
         Increment transition count for roads with red light
         if any road has not received a green light for 4 transitions then
            Prioritize these roads based on vehicle count
            if multiple roads have the same vehicle count then
               Choose the road that has waited the longest
            end if
            Set the chosen road to green
         else
            Compare vehicle counts of all red roads
            if multiple roads have the same vehicle count then
               Choose the road that has waited the longest
            end if
            Set the chosen road to green
         end if
      end if
      Update the display with the current traffic light state for each road
   end for
end while
End the traffic management when the stop button is pressed
```

5.1.2 System Flowchart

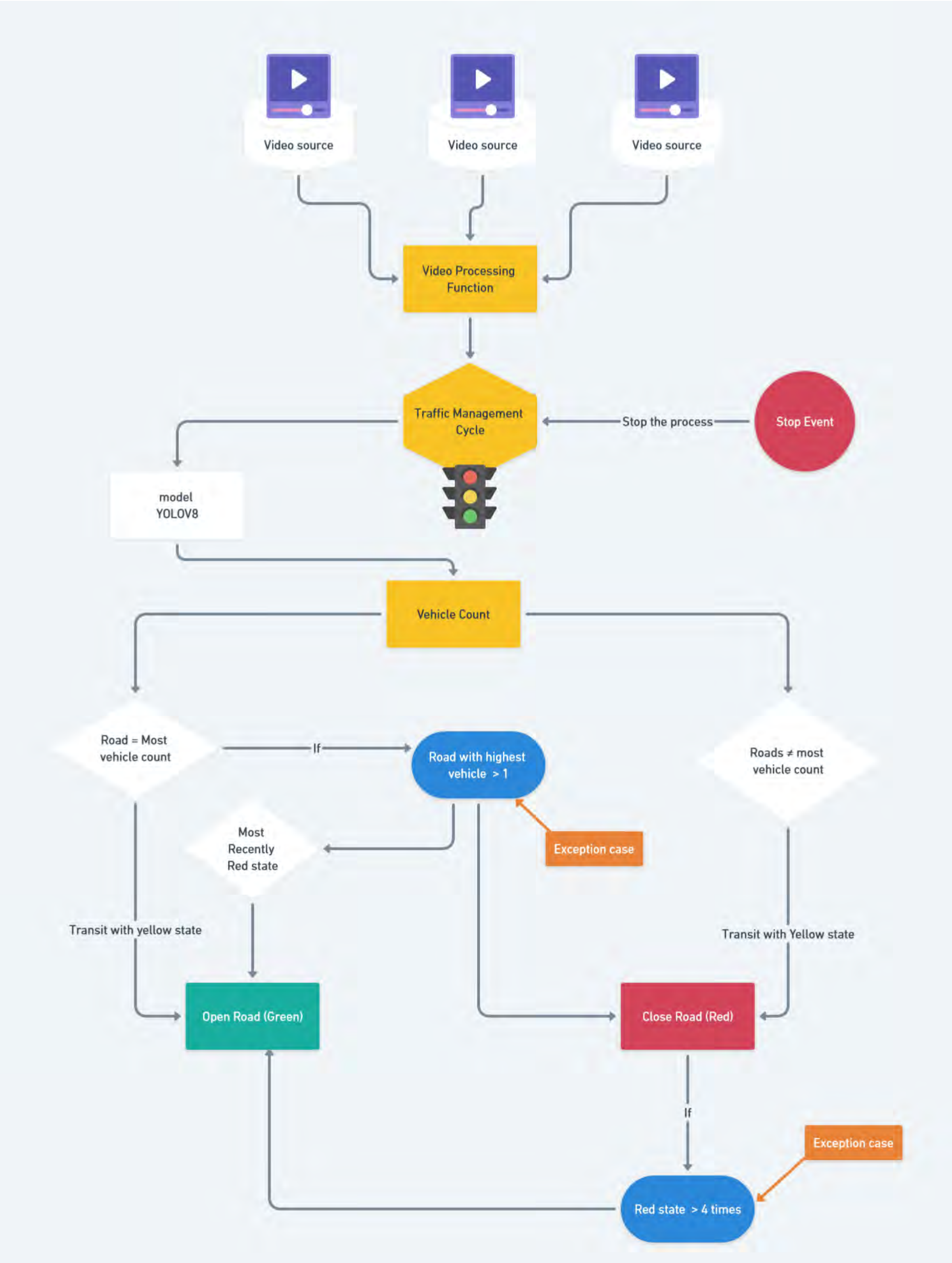

Figure 5.1: System Flowchart

Implementation:

The code that is shown below represents Traffic Signal Management System that now employs a YOLOv8 for object detection and Streamlit for interaction and Visualization. The system provides view from several cameras and based on the detected vehicle data controls traffic signals in real time. This efficient smart traffic control system lowers traffic jams and enhances free flow of traffic at junctions.
The system's core functionalities are divided into the following key components:

Streamlit(st): Streamlit is a Python library for creating a user-friendly website interface. Users can copy and paste video files, set input signal lengths, and preview the output video in real time.

OpenCV (cv2): A computational tool of complex character and high performance, which is used for managing video stream, frames' reading, and possible image processing.

Numpy (np): Assists in performing mathematical calculations; in creating arrays utilizing which images are processed.

YOLOv8 of the ultralytics library: This is presumably the current best practice in object detection. Real-time will be especially effective for detecting vehicles in video feeds.

Tempfile: Tempfile is particularly useful in handling of temporary storage space for the videos that users upload so that the management of large video data can be done effectively without leading to the need for permanent storage.

Time: Required for controlling cycle time related to traffic lights such as red time, yellow time or green time.

OS: An application can delete temporary files after processing so that all extra data is not kept by the system.

Threading.Event: Provides synchronization control to pause or stop the processing loop, allowing the user to experience continuity of start, stop, or pause for the simulation.

Video Input Handling: The system can accept video inputs from multiple roads, which simulate various intersections. Each video file, uploaded via the application interface, represents real-time traffic data collected from cameras placed at these intersections. Minimum 3 video have to be added as intersections contains at least three different roads.

Vehicle Detection: The detection of vehicles (such as cars, buses, motorcycles, etc.) in the video frames is handled using the YOLOv8 model. The model has been optimized for traffic-related objects, these include cars, buses, motorcycles, and bicycles. Detection occurs on a frame-by-frame basis, with bounding boxes drawn around the identified vehicles to demonstrate the real time actions.

Signal State Management: Traffic light signals are simulated using three states: red, yellow, and green. These states control the flow of traffic at each intersection based on the vehicle density. The application calculates the optimal signal state by evaluating the number of detected vehicles on each road. The signals are timed dynamically to allow the road with the highest traffic density to have a green light, ensuring smoother traffic flow.

Dynamic Signal Control: The system continuously processes the video feeds and adjusts the signal state based on the real-time traffic density. The process is managed in a loop where each intersection's traffic density is compared, and the signal light is adjusted accordingly. After a specific green light duration, the system switches to yellow for a brief period before changing to red imitating the real time traffic signals.

Streamlit-Based Interface: The user interface is built using Streamlit, and deployed on Streamlit, which allows for seamless interaction with the system. Users can upload video files, set the duration for red and yellow lights, and start or stop the traffic simulation through a user-friendly interface. The video feeds are displayed in a grid format, each corresponding to a different intersection, along with the signal states.

5.2 Functionality and Workflow

The whole application follows these steps during the usage:

1. **Video Upload**: The user uploads at least three video files, each representing a different road. These videos are processed in real-time to detect the number of vehicles on each road.

2. **Object Detection and Processing**: The YOLOv8 model processes each frame from the uploaded videos to detect vehicles. The detections are processed in real time, and bounding boxes are drawn around the identified objects (vehicles).

3. **Traffic Light Simulation**: A dynamic algorithm determines which road should receive the green light based on the vehicle count. Roads with higher vehicle counts get longer green light durations, while less congested roads wait on red or yellow lights. The light state is updated in real-time, with the green light duration being adjustable through a user-configured slider.

4. **Visualization and Control**: The Streamlit interface provides a real-time visual of the ongoing simulation, displaying both the video streams and the current traffic light state for each intersection. The application offers control buttons for starting, pausing, or stopping the simulation.

5. **Data Cleanup**: After processing each frame, the system releases the temporary resources associated with the video files to maintain efficiency. If the simulation is stopped, the system clears any ongoing processes to ensure smooth operation.

5.3 Code and System efficiency

The system's design is optimized for handling multiple video streams simultaneously by using Python's threading and event management system, enabling efficient concurrent processing of traffic data. Essential libraries such as like OpenCV, Streamlit, and YOLOv8 are implemented to ensure best real-time performance.
Core Libraries

1. **OpenCV**: Used for video input and processing. It handles real-time frame reading, video playback, and drawing of traffic lights and vehicle detection boxes.

2. **Streamlit**: Provides an interactive web interface for video uploads, control buttons (start, stop), and dynamic adjustments of traffic signal timing.

3. **YOLOv8**: is used for detecting objects (vehicles) in real-time within each frame of the video streams. The model performs inference on the frames to detect various vehicle types, providing bounding box coordinates and class labels.

5.3.1 Handling Edge Case:

Sequential Green Assignment or Equal Count Resolution: Handling Equal Number of Vehicles on two or more Roads:

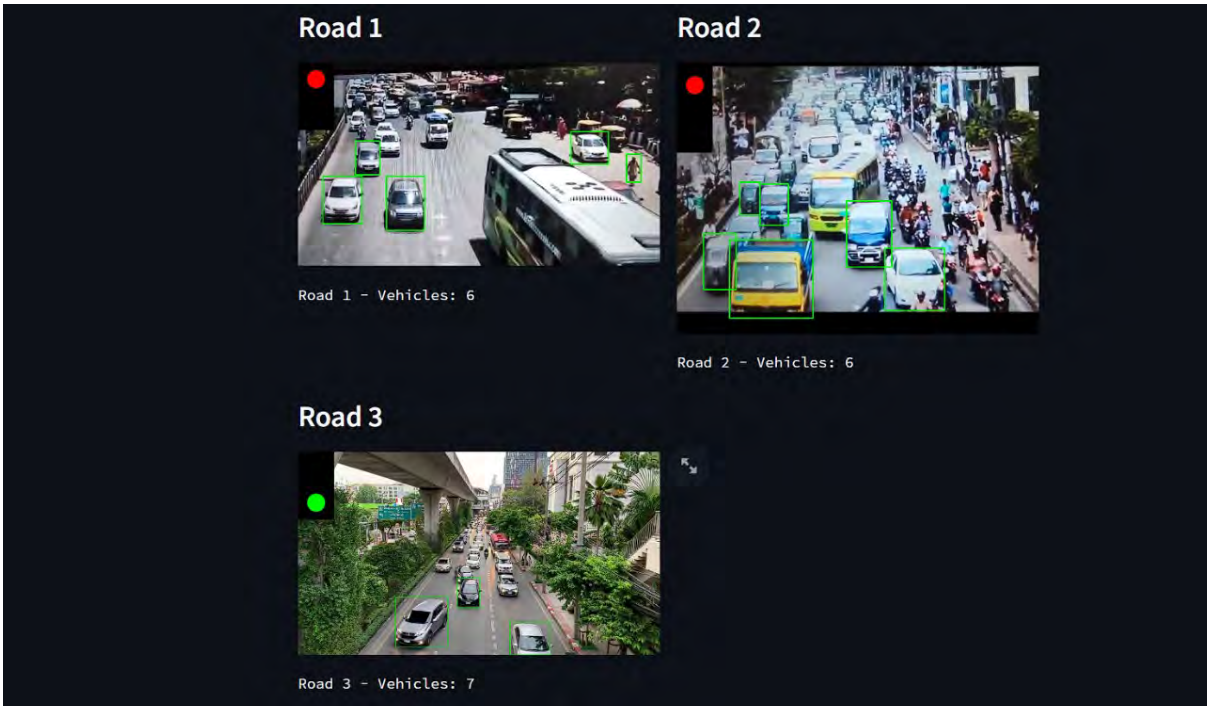

Figure 5.2: Same Number of Vehicles

In this traffic system, when two or more roads have the same number of cars, there's a special rule to decide which road gets the green light first. The system is organized by road numbers in particular order. For example, if road 1 and road 2 each have the same number of cars then road 1 will be given green light before road 2. This method is called Sequential Green Assignment or Equal Count Resolution. In other words, the system keeps all the vehicles on the roads treated equally, when multiple roads are equally busy. By following this rule, the system avoids delays that could happen if roads with the same car count all tried to go at once. This helps traffic lights switch more efficiently, making the overall flow of traffic better.

Equal Opportunity Transition Rule:

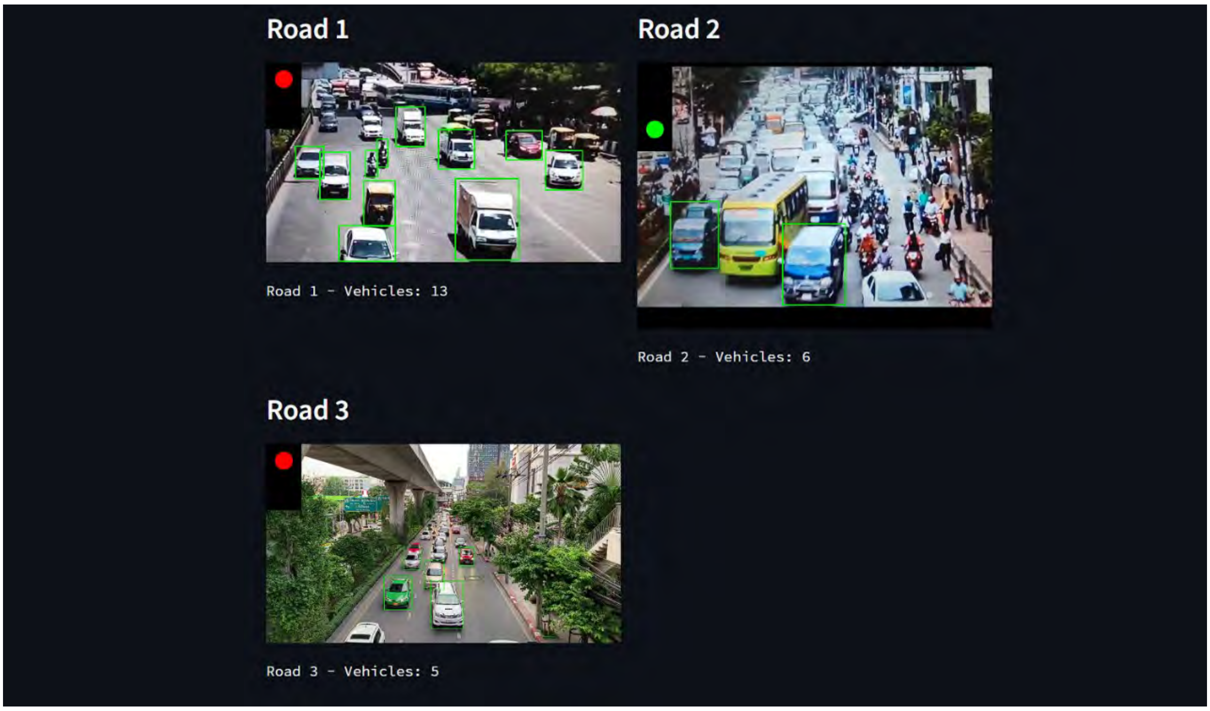

Figure 5.3: Equal Opportunity Transition Rule

In this system, if a road with fewer cars doesn't get a green light for a while, there's a rule to fix that. It's called the Equal Opportunity Transition rule. After three green lights for other roads, the road with fewer cars will get a green light on the fourth turn. This all means that all roads have a fair chance to move, keeping traffic fair and balanced. By making sure that even less busy roads get attention regularly, the system prevents long waits and gets you smooth traffic flow across the entire network. This approach makes the whole traffic system more efficient and equitable for all drivers.

Chapter 6

Conclusion

6.1 Conclusion

In conclusion, the system we proposed for automating traffic singnals can be very useful if it's utilized in real life scenario. In this project we proposed and designed an adaptive traffic signal control system, the system's simulation was created using YOLOv8 and OpenCV on Streamlit. Traffic data obtained from Roboflow was im- plemented to the YOLOv8 model, trained for real time vehicle detection to optimize the traffic signal timings. OpenCV was incorporated to address image analysis tasks, while the simulation traffic signals control interface was developed using Streamlit. This innovative approach tackles problems in traffic flow for congested lanes by properly scheduling the traffic light and vehicle flow. In any case, the vision-based strategy that we proposed can help to enhance the control of traffic flows and increase the safety of roads significantly. Our core aim was to design a system that would allow changing traffic lights depending on the existing conditions. We showed the demonstration in a simulation. If this is implemented properly in real life, this will help to alleviate traffic congestion and improve overall trafic flow.

6.2 Future Work

Our work on adaptive traffic signal management system can be scaled and improved. The system can be upgraded to favour emergency vehicles like fire brigades and ambulances, to give them green lights to help them travel faster. We can make the technology sustainable by implementing sustainable strategies like dimming street lights during low-traffic hours. Solar-powered traffic lights can be used. The system must be validated to be able to sustain different traffic levels, road configurations and peak hour conditions in urban areas. With its use of artificial intelligence, the system will be able to identify vehicle license plates, and thus enforce traffic regulations more effectively. To enhance traffic efficiency and safety, this system should be linked to smart city, smart parking and other intelligent transportation technologies, to build up an advanced and interactive urban transportation network.

Bibliography

[1] J. A. Hillier and R. Rothery, “The synchronization of traffic signals for minimum delay,” *Transportation Science*, vol. 1, no. 2, pp. 81–94, 1967. DOI: 10.1287/trsc.1.2.81. eprint: https://doi.org/10.1287/trsc.1.2.81. [Online]. Available: https://doi.org/10.1287/trsc.1.2.81.

[2] M. J. Smith and T. van Vuren, “Traffic equilibrium with responsive traffic control,” *Transportation Science*, vol. 27, no. 2, pp. 118–132, 1993. DOI: 10.1287/trsc.27.2.118. eprint: https://doi.org/10.1287/trsc.27.2.118. [Online]. Available: https://doi.org/10.1287/trsc.27.2.118.

[3] T. Nakatsuji, S. Seki, S. Shibuya, and T. Kaku, “Artificial intelligence approach for optimizing traffic signal timings on urban road network,” in *Proceedings of VNIS'94 - 1994 Vehicle Navigation and Information Systems Conference*, 1994, pp. 199–202. DOI: 10.1109/VNIS.1994.396841.

[4] I. PORCHE and S. LAFORTUNE, “Adaptive look-ahead optimization of traffic signals,” *ITS Journal - Intelligent Transportation Systems Journal*, vol. 4, no. 3-4, pp. 209–254, 1999. DOI: 10.1080/10248079908903749. eprint: https://doi.org/10.1080/10248079908903749. [Online]. Available: https://doi.org/10.1080/10248079908903749.

[5] B. Abdulhai, R. Pringle, and G. Karakoulas, “Reinforcement learning for true adaptive traffic signal control,” *Journal of Transportation Engineering*, vol. 129, May 2003. DOI: 10.1061/(ASCE)0733-947X(2003)129:3(278).

[6] A. Hamilton, B. Waterson, T. Cherrett, A. Robinson, and I. Snell, “The evolution of urban traffic control: Changing policy and technology,” en, *Transp. Plan. Technol.*, vol. 36, no. 1, pp. 24–43, Feb. 2013.

[7] Y. Shen, J. Liu, M. Ito, and N. Shiratori, “Adaptive traffic signal control: Deep reinforcement learning algorithm with experience replay and target network,” May 2017.

[8] B. Balci, B. Alkan, A. Elihos, and Y. Artan, “Nir camera based mobile seat belt enforcement system using deep learning techniques,” in *2018 14th International Conference on Signal-Image Technology Internet-Based Systems (SITIS)*, 2018, pp. 247–252. DOI: 10.1109/SITIS.2018.00045.

[9] Y. Wang, X. Yang, H. Liang, and Y. Liu, “A review of the self-adaptive traffic signal control system based on future traffic environment,” *Journal of Advanced Transportation*, vol. 2018, pp. 1–12, Jun. 2018. DOI: 10.1155/2018/1096123.

[10] Y. Wang, X. Yang, H. Liang, and Y. Liu, "A review of the self-adaptive traffic signal control system based on future traffic environment," *Journal of Advanced Transportation*, vol. 2018, no. 1, p. 1 096 123, 2018. DOI: https://doi.org/10.1155/2018/1096123. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1155/2018/1096123. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1155/2018/1096123.

[11] H. R. Mampilayil and R. K., "Deep learning based detection of one way traffic rule violation of three wheeler vehicles," in *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, 2019, pp. 1453–1457. DOI: 10.1109/ICCS45141.2019.9065638.

[12] S. Asoba, S. Supekar, T. Tonde, and J. A. Siddiqui, "Advanced traffic violation control and penalty system using iot and image processing techniques," in *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, 2020, pp. 554–558. DOI: 10.1109/ICIMIA48430.2020.9074949.

[13] R. J. Franklin and Mohana, "Traffic signal violation detection using artificial intelligence and deep learning," in *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, 2020, pp. 839–844. DOI: 10.1109/ICCES48766.2020.9137873.

[14] J. C. de Goma, R. J. Bautista, M. A. J. Eviota, and V. P. Lopena, "Detecting red-light runners (rlr) and speeding violation through video capture," in *2020 IEEE 7th International Conference on Industrial Engineering and Applications (ICIEA)*, 2020, pp. 774–778. DOI: 10.1109/ICIEA49774.2020.9102059.

[15] A. Shaikat, R. Tasnim, F. Mahbub, R. Saleheen, R. Mahmud, and T. Islam, "An image processing and artificial intelligence based traffic signal control system of dhaka," Dec. 2020. DOI: 10.1109/APCoRISE46197.2019.9318966.

[16] T. V. Toan and T. Van Hoai, "An ai-based approach for arterial signal control in heterogeneous traffic," in *2020 International Conference on Advanced Computing and Applications (ACOMP)*, 2020, pp. 42–48. DOI: 10.1109/ACOMP50827.2020.00014.

[17] A. Tonge, S. Chandak, R. Khiste, U. Khan, and L. A. Bewoor, "Traffic rules violation detection using deep learning," in *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 2020, pp. 1250–1257. DOI: 10.1109/ICECA49313.2020.9297495.

[18] A. H. Ahmed and L. Z. Fragonara, "Adaptive intelligent traffic control systems for improving traffic quality and congestion in smart cities," 2021.

[19] A. Atzori, S. Barra, S. Carta, G. Fenu, and A. S. Podda, "Heimdall: An ai-based infrastructure for traffic monitoring and anomalies detection," in *2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, 2021, pp. 154–159. DOI: 10.1109/PerComWorkshops51409.2021.9431052.

[20] S. Channi, S. Reddy, P. Totad, and A. H. Nalband, "Ai powered yolo based traffic management system through application advancement," Jun. 2021.

[21] P. Reddy, T. Nishwa, R. S. K. Reddy, C. Sadviq, and K. Rithvik, "Traffic rules violation detection using machine learning techniques," in *2021 6th International Conference on Communication and Electronics Systems (ICCES)*, 2021, pp. 1264–1268. DOI: 10.1109/ICCES51350.2021.9488998.

[22] S. S. Wankhede and P. Bajaj, "Traffic violation detection model using soft computing tools," in *2021 6th International Conference for Convergence in Technology (I2CT)*, 2021, pp. 1–5. DOI: 10.1109/I2CT51068.2021.9417887.

[23] R. Zhang, A. Ishikawa, W. Wang, B. Striner, and O. K. Tonguz, "Using reinforcement learning with partial vehicle detection for intelligent traffic signal control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 1, pp. 404–415, 2021. DOI: 10.1109/TITS.2019.2958859.

[24] M. Cao, V. O. K. Li, and Q. Shuai, "A gain with no pain: Exploring intelligent traffic signal control for emergency vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 17 899–17 909, 2022. DOI: 10.1109/TITS.2022.3159714.

[25] S. Maadi, S. Stein, J. Hong, and R. Murray-Smith, "Real-time adaptive traffic signal control in a connected and automated vehicle environment: Optimisation of signal planning with reinforcement learning under vehicle speed guidance," *Sensors*, vol. 22, no. 19, 2022, ISSN: 1424-8220. DOI: 10.3390/s22197501. [Online]. Available: https://www.mdpi.com/1424-8220/22/19/7501.

[26] T. Singh, V. Rajput, Satakshi, U. Prasad, and M. Kumar, "Real-time traffic light violations using distributed streaming," *The Journal of Supercomputing*, vol. 79, no. 7, pp. 7533–7559, Nov. 2022. DOI: 10.1007/s11227-022-04977-4.

[27] S. Bhatnagar, R. Guo, K. McCabe, T. L. McCluskey, F. Percassi, and M. Vallati, "Automated planning for generating and simulating traffic signal strategies.," in *IJCAI*, 2023, pp. 7119–7122.

[28] K. Biswas, N. Paul, D. Saha, T. Ahmed, and R. Mahmud, "Detection of traffic rule violations using machine learning: An analytical review," *Malaysian Journal of Science and Advanced Technology*, vol. 3, pp. 37–47, Mar. 2023. DOI: 10.56532/mjsat.v3i1.146.

[29] F. Gazzawe and M. Albahar, "Reducing traffic congestion in makkah during hajj through the use of ai technology," *Heliyon*, vol. 10, e23304, Dec. 2023. DOI: 10.1016/j.heliyon.2023.e23304.

[30] R. Kumar, *A research paper on ai for traffic management*, Mar. 2023. DOI: 10.13140/RG.2.2.21119.69287.

[31] S. A. Kumar, "Detecting and counting different vehicles in the real time traffic signal using deep learning," Ph.D. dissertation, Dublin, National College of Ireland, 2023.

[32] G. R. P. Kumari, M. Jahnavi, M. Harika, A. Pavani, and C. V. Lakshmi, "Smart traffic signal control system using artificial intelligence," in *Intelligent Communication Technologies and Virtual Mobile Networks*, G. Rajakumar, K.-L. Du, and Á. Rocha, Eds., Singapore: Springer Nature Singapore, 2023, pp. 829–838, ISBN: 978-981-99-1767-9.

[33] M. A. Ottom and A. Al-Omari, "An adaptive traffic lights system using machine learning.," *Int. Arab J. Inf. Technol.*, vol. 20, no. 3, pp. 407–418, 2023.

[34] P. V, A. R, R. D, and S. G, "Ai-driven urban traffic optimization to assess complex traffic patterns for public traffic control and mobility," *International Journal for Research in Applied Science and Engineering Technology*, vol. 11, pp. 794–798, Nov. 2023. DOI: 10.22214/ijraset.2023.56630.

[35] M. Yazdani, M. Sarvi, S. A. Bagloee, N. Nassir, J. Price, and H. Parineh, "Intelligent vehicle pedestrian light (ivpl): A deep reinforcement learning approach for traffic signal control," *Transportation research part C: emerging technologies*, vol. 149, p. 103 991, 2023.

[36] A. Agrahari, M. M. Dhabu, P. S. Deshpande, A. Tiwari, M. A. Baig, and A. D. Sawarkar, "Artificial intelligence-based adaptive traffic signal control system: A comprehensive review," *Electronics*, vol. 13, no. 19, 2024, ISSN: 2079-9292. DOI: 10.3390/electronics13193875. [Online]. Available: https://www.mdpi.com/2079-9292/13/19/3875.

[37] C. Ashokkumar, D. A. Kumari, S. Gopikumar, N. Anuradha, R. S. Krishnan, and S. I, "Urban traffic management for reduced emissions: Ai-based adaptive traffic signal control," in *2024 2nd International Conference on Sustainable Computing and Smart Systems (ICSCSS)*, 2024, pp. 1609–1615. DOI: 10.1109/ICSCSS60660.2024.10625356.

[38] H. Khan and J. Thakur, "Smart traffic control: Machine learning for dynamic road traffic management in urban environments," *Multimedia Tools and Applications*, pp. 1–25, May 2024. DOI: 10.1007/s11042-024-19331-4.

[39] A. Phand, S. Bagade, N. Bandgar, and P. Wayal, "Real-time traffic light optimization using ai and iot," *International Journal for Research in Applied Science and Engineering Technology*, vol. 12, pp. 3534–3539, Apr. 2024. DOI: 10.22214/ijraset.2024.60686.

[40] G. V. Priya, B. Gowri, B. Rajalakshmi, L. Dilli Kumar, U. Janagan, and R. Sudhir, "A novel approach to mitigating traffic issues through advanced congest detection and adaptive signal control using ai," in *2024 4th International Conference on Intelligent Technologies (CONIT)*, 2024, pp. 1–5. DOI: 10.1109/CONIT61985.2024.10627587.

[41] D. E. Topal, "Ai-based dynamic traffic signal control systems for autonomous vehicle navigation," *Journal of Bioinformatics and Artificial Intelligence*, vol. 1, no. 2, pp. 38–51, Jun. 2024. [Online]. Available: https://biotechjournal.org/index.php/jbai/article/view/31.