TRAFFIC SIGNAL DETECTION USING AI BASED OBJECT DETECTION ALGORITHMS

THESIS SUBMITTED

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE AWARD OF THE DEGREE OF

MASTER OF TECHNOLOGY

IN

INTELLIGENT AUTOMATION AND ROBOTICS

(DEPARTMENT OF ELECTRONICS & TELECOMMUNICATION ENGINEERING)

By

SUPRATIM BISWAS

Registration No: 160224 of 2021 - 2022

Examination Roll No.: M6IAR24007

Under the guidance of

Prof. SHELI SINHA CHAUDHURI

Department of Electronics & Telecommunication Engineering

Jadavpur University

Kolkata – 700032

West Bengal, India

May 2024

DECLARATION OF ORIGINALITY AND COMPLIANCE OF ACADEMIC ETHICS

I hereby declare that this thesis contains a literature survey and original work by the undersigned candidate, as part of my Master of Technology in Intelligent Automation and Robotics in the Department of Electronics and Telecommunication Engineering. All information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that I have thoroughly cited and referenced all material and findings which are not original to this research, as provided by these rules and conduct.

Name: Supratim Biswas

Exam Roll No: M6IAR24007

Thesis Title: "TRAFFIC SIGNAL DETECTION USING AI BASED OBJECT DETECTION ALGORITHMS"

Signature of Candidate

FACULTY OF ENGINEERING & TECHNOLOGY JADAVPUR UNIVERSITY

CERTIFICATE OF RECOMMENDATION

This is to certify that the thesis entitled **—** "**TRAFFIC SIGNAL DETECTION USING AI BASED OBJECT DETECTION ALGORITHMS**" has been carried out by Supratim Biswas (Class Roll No: 002110704007, Examination Roll No.: M6IAR24007 and Registration No: **160224** of **2021-22**) under my guidance and supervision and be accepted in partial fulfillment of the requirement for the degree of Master of Technology in Intelligent Automation and Robotics offered by the Department of Electronics and Telecommunication Engineering.

Prof. Sheli Sinha Chaudhuri
Supervisor
Department of Electronics &
Telecommunication Engineering
Jadavpur University

Prof. Amit Konar
Course Co-ordinator
Department of Electronics &
Telecommunication Engineering
Jadavpur University

Prof. Manotosh Biswas
Head of the Department
Department of Electronics &
Telecommunication Engineering
Jadavpur University

Prof. Dipak Laha
Dean
Faculty Council of
Engineering & Technology
Jadavpur University

FACULTY OF ENGINEERING & TECHNOLOGY JADAVPUR UNIVERSITY

CERTIFICATE OF APPROVAL

The forgoing thesis titled − "**TRAFFIC SIGNAL DETECTION USING AI BASED OBJECT DETECTION ALGORITHMS**" is hereby approved as a creditworthy study of an engineering subject conducted and presented satisfactorily to warrant its acceptance as a precondition to the degree for which it was submitted. It is understood that the undersigned does not automatically support or accept any argument made, opinion expressed, or inference is drawn in it by this approval, but only approve the thesis for the reason for which it was submitted.

Committee on Final Examination for

Evaluation of the Thesis

Signature of External Examiner

Signature of Supervisor

ACKNOWLEDGEMENT

This thesis entitled "**TRAFFIC SIGNAL DETECTION USING AI BASED OBJECT DETECTION ALGORITHMS**" is the result of the work whereby I have been accompanied and supported by many people, my guide, my friends, and lab seniors. It is a pleasant aspect that now I have the opportunity to express my gratitude to all of them.

First and foremost, I would like to express my sincere gratitude to my thesis supervisor **Prof. Sheli Sinha Chaudhuri**, Professor of Department of Electronics and Telecommunication Engineering, Jadavpur University for her valuable guidance, insightful suggestion and support while conducting this thesis work as well as in the writing of this thesis. I have been very fortunate to have a guide like her. Her positivity, confidence, and ideas help me to complete my thesis and she guides me as a guardian.

I am thankful to **Prof. Manotosh Biswas**, Head of the Electronics and Telecommunication Engineering Department, for granting me access to the laboratory and other departmental facilities essential for the completion of this thesis. I would also like to express my gratitude to **Prof. Amit Konar** for his valuable advice and support. I would also like to express my gratitude towards all the respected professors of the Department of Electronics and Telecommunication Engineering, Jadavpur University for their valuable guidance and advice.

I am thankful to my fellow project mates, friends, technical and non-technical staff of Jadavpur University who has helped me directly or indirectly during the tenure of my thesis work.

I want to express my gratitude to my parents and family also, as, without their sacrifices, I can't do anything. And their invaluable love, encouragement, and support make me, whatever I am today.

Supratim Biswas

Jadavpur University

Kokata-32, West Bengal

ABSTRACT

Traffic accidents continue to be a major problem on a global scale, resulting in fatalities, property damage, and high financial expenses. The frequency of traffic accidents is still high even with improvements in road infrastructure and car safety features. The ongoing problem of traffic signal violations, which frequently result from human mistake such as intentional non-compliance or distraction, is one of the contributing factors. As a result, there is an urgent need for creative solutions that might improve traffic signal compliance and thereby increase road safety. The proposed Traffic Signal Detection using AI based Object Detection Algorithm represents a forward-thinking approach to mitigating road accidents by focusing on traffic signal compliance. The system is designed to identify traffic signals in real-time and alert drivers to their status, particularly in high-traffic situations where the risk of accidents is elevated. This proactive alert mechanism aims to reduce instances of signal violation by capturing drivers' attention and encouraging timely and appropriate responses to traffic signals. At the heart of the system is the integration of the You Only Look Once (YOLO) algorithm within a Convolutional Neural Network (CNN) framework. The YOLO algorithm is renowned for its ability to perform object detection tasks with high accuracy and speed. By employing YOLO, the system can rapidly and accurately identify traffic signals from real-time video feeds. By enhancing the accuracy and reliability of traffic signal detection, the system holds promise for significantly reducing road accidents. Additionally, the system's capability to monitor and record traffic signal violations provides valuable data for traffic authorities, which can be used to identify patterns of non-compliance and develop targeted interventions. Despite improvements in infrastructure and car safety, traffic signal violations remain a major cause of accidents, necessitating innovative solutions for better compliance. The proposed AI-based Traffic Signal Detection System uses advanced object detection to improve signal adherence, enhancing road safety and providing valuable data for traffic management.

Keywords: Traffic Signal Detection; Deep Learning; Convolutional Neural Network; Object Detection Algorithm; You Only Look Once.

Table of Contents

Declaration of originality and compliance of academic ethics i

Certificate of recommendation ii

Certificate of approval iii

Acknowledgement iv

Abstract v

Table of contents vi-x

Chapter 1: Introduction **1-4**

Chapter 2: Literature Survey **5-30**
- Traditional Methods 6-9
  - Gaussian Distribution 6
  - Hough Transform 6
  - Fast Radial Symmetry Transform 8
- Machine Learning Methods 9-12
  - Support Vector Machine 9
- Deep Learning Methods 12-30
  - Convolutional Neural Network 12
  - Region-based Convolutional Neural Network 13
  - Faster Region-based Convolution Neural Network 14
  - Single Shot Detector 16
  - You Only Look Once 19

Chapter 3: Study on Methodological Advances in Traffic Signal Identification **31-40**
- Dataset Preparation 33

Dataset Training 36
Environment Setting 37
Signal Detection 38
Database Synchronization 39

Chapter 4: Results and Observations 41-57
Metrics 41-42
Metrics Precision 41
Metrics Recall 41
F1 Score 41
Mean Average Precision 42
Training Results of YOLOv5 42
Training Results of YOLOv7 45
Training Results of YOLOv8 48
Performance Comparison between the Models 52-54
Graphical Representation of the Model's Performance 52
Database Results 54
Detection Results of the 3 YOLO models 57

Chapter 5: Architectural Analysis of the YOLO Models 58-67
YOLOv5 58
YOLOv7 61
YOLOv8 64
Comparison based on the Results 66-67
Backbone Network 66
Neck Design 66
Detection Head 66
Optimization Techniques 67

Chapter 6: Conclusion 68

Chapter 7: Bibliography 70-73

List of Figures

Fig. No.	List of Figures/ Tables	Page
Fig 2.1	Various Object Detection Method	5
Fig 2.2	Traffic Sign/ Light Detection using HOG and SVM	10
Fig 2.3	Architecture of Faster R-CNN	15
Fig 2.4	Architecture of SSD	18
Fig 2.5	YOLOv3 algorithm for Traffic Sign Detection	22
Fig 2.6	YOLOv5 architecture	27
Fig 3.1	Basic Architecture of YOLOv8 Object Detection Model	31
Fig 3.2	Bar chart of no. images having a class	36
Fig 3.3	Bar chart of no. of objects per class	37
Fig 3.4	An illustration of the flowchart of proposed method	39
Fig 3.5	Flowchart of syncing the database to the proposed system	40
Fig 4.1	Precision curve of YOLOv5s	43
Fig 4.2	Recall curve of YOLOv5s	43
Fig 4.3	Precision-Recall curve of YOLOv5s	43
Fig 4.4	F1 curve of YOLOv5s	44
Fig 4.5	Confusion Matrix of YOLOv5s	45
Fig 4.6	Training Results of YOLOv5s	45
Fig 4.7	Precision curve of YOLOv7	46
Fig 4.8	Recall curve of YOLOv7	46
Fig 4.9	Precision-Recall curve of YOLOv7	46

Fig 4.10	F1 curve of YOLOv7	47
Fig 4.11	Confusion Matrix of YOLOv7	48
Fig 4.12	Training Results of YOLOv7	48
Fig 4.13	Precision curve of YOLOv8s	49
Fig 4.14	Recall curve of YOLOv8s	49
Fig 4.15	Precision-Recall curve of YOLOv8s	49
Fig 4.16	F1 curve of YOLOv8s	50
Fig 4.17	Confusion Matrix of YOLOv8s	51
Fig 4.18	Training Results of YOLOv8s	51
Fig 4.19	(a) Precision (b) Recall (c) mAP@0.50 (d) mAP@[0.050-0.95]	52
Fig 4.20	(a) Box Loss (b) Class Loss (c) DFL Loss	53
Fig 4.21	Car Details	55
Fig 4.22	Owner Details	55
Fig 4.23	RTO Details	56
Fig 4.24	Fine Details	56
Fig 4.25	Defaulters Details	56
Fig 5.1	Basic Architecture of YOLOv5	59
Fig 5.2	Basic Architecture of YOLOv7	62
Fig 5.3	Basic Architecture of YOLOv8	65

List of Tables

Table No.	List of Figures/ Tables	Page
Table 3.1	Data Sample of proposed dataset	34-35
Table 4.1	Results obtained while testing with YOLOv8, YOLOv5 and YOLOv7	52
Table 4.2	Comparison of Signal Detection results using YOLOv5, YOLOv7 and YOLOv8	57

CHAPTER 1: INTRODUCTION

The issue of road safety is a significant concern worldwide, especially in rapidly developing countries like India, which has become the most populous nation. The high incidence of road accidents in India can be attributed to various factors, including but not limited to the following:

The dire state of road infrastructure in India is a pivotal factor contributing to its high road accident rates. A substantial portion of the road network suffers from neglect, showcasing surfaces ridden with potholes, insufficient lighting, and a lack of essential road markings. Moreover, pedestrian crossings and traffic signal systems, where they exist, often fail to meet the necessary standards of visibility and efficiency. This not only endangers vehicle occupants but also puts pedestrians at significant risk. The inadequate infrastructure fails to accommodate the diverse needs of all road users, including cyclists and motorcyclists, leading to chaotic and unsafe road conditions.

Overloading vehicles is a common practice in India, not just in commercial transportation but also in private vehicles, compromising the safety and manoeuvrability of these vehicles. Exceeding the weight limit puts additional strain on the vehicle's brakes and tires, making it difficult to handle, especially in emergency situations. Simultaneously, the lack of adherence to maintenance protocols leads to vehicles being driven in suboptimal conditions. Worn tires, faulty brakes, and malfunctioning headlights are just a few examples of maintenance issues that significantly increase the likelihood of accidents.

The role of human error in road accidents cannot be overstated. Reckless driving behaviours, such as speeding, not yielding the right of way, and erratic lane changes, are rampant. Driving under the influence of alcohol or drugs further impairs judgment and reaction times, escalating the risk of catastrophic accidents. Moreover, the disregard for traffic rules — like running red lights or not wearing seatbelts and helmets — exemplifies a widespread culture of non-compliance and contributes to the severity and frequency of road accidents.

The lax enforcement of traffic laws in India emboldens drivers to ignore regulations without fear of repercussions. This is compounded by systemic issues such as corruption, where violators can often evade penalties through bribes, and a lack of

resources, including insufficient numbers of traffic police and inadequate technology to monitor and enforce laws. The result is a road environment where violations go unpunished, encouraging a cycle of dangerous driving behaviours that endanger all road users.

A significant challenge in improving road safety is the widespread lack of awareness about safe driving practices and pedestrian behaviour. Educational initiatives on road safety are limited and fail to reach a broad audience. Many drivers and pedestrians are either unaware of the rules or choose to ignore them, leading to risky behaviours like jaywalking, ignoring pedestrian signals, and unsafe crossing practices. The absence of a strong culture of road safety, starting from a young age, perpetuates these issues.

Addressing these challenges requires a comprehensive approach that includes upgrading and maintaining infrastructure, enforcing vehicle load and maintenance standards, educating the public on safe road practices, strengthening law enforcement mechanisms, and building widespread awareness about the importance of road safety. Only through concerted efforts across these domains can significant progress be made in reducing the alarmingly high rates of road accidents in India.

One of the primary road accidents that occur is due to the traffic signal violation. The pervasive issue of traffic signal violations on Indian roads significantly contributes to the country's alarmingly high rate of road accidents and traffic congestion. This problem is deeply intertwined with broader road safety challenges in India, stemming from a confluence of factors including inadequate infrastructure, enforcement deficiencies, and cultural attitudes towards road use.

In India, the roads are often scenes of chaos, marked by congestion where vehicles, pedestrians, and sometimes animals are forced to share the same cramped space. In such conditions, the urge to save time or simply move ahead compels many drivers to attempt beating the traffic signals, a practice that has become all too common. This behaviour not only disrupts the orderly flow of traffic, exacerbating congestion, but also significantly increases the risk of accidents, putting numerous lives in danger.

This issue of signal violation is emblematic of the broader challenges facing road safety in India. The inadequate road infrastructure often lacks the necessary

signage, proper lane markings, and well-functioning traffic signals, making compliance with traffic rules a challenging endeavour even for the well-intentioned motorist. Addressing this problem requires a holistic approach that not only focuses on enhancing road infrastructure and strengthening law enforcement but also on changing cultural attitudes towards road use. This includes improving driver education, raising public awareness about the consequences of traffic violations, and fostering a more respectful and law-abiding road use culture.

For the escalating concern over road safety in India, highlighted by a marked increase in traffic offenses and accidents attributed to traffic signal violations or jamming, underscores the urgent need for innovative technological solutions. The reported 23% surge in traffic violations, as noted by The Times of India, along with the authorities' challenges in tracking down violators due to technological limitations, amplifies the critical need for advanced systems to aid in mitigating these issues. In response to this dire situation, the proposed Smart Traffic Signal Detection System emerges as a promising solution aimed at enhancing road safety by leveraging the latest advancements in deep learning and computer vision.

The Smart Traffic Signal Detection System represents a forward-thinking approach to tackling the persistent issue of traffic signal violations in India. By harnessing the power of YOLOv8 and CNNs, this system provides a robust, real-time solution capable of enhancing driver compliance with traffic signals, thereby promising a significant reduction in road accidents and traffic offenses. Its implementation could mark a pivotal step towards safer and more regulated roadways in India, showcasing the transformative potential of AI and deep learning technologies in addressing critical public safety challenges.

The rest of the paper is organised as follows: Chapter 2 provides a comprehensive survey of the existing literature in the field of object detection, with a specific focus on traffic signal detection. It details various methodologies and techniques that have been developed and employed, highlighting their unique contributions and effectiveness. Key theories, models, and empirical studies will be reviewed to establish a solid foundation for the research. This section also discusses the evolution of technology in this area and identifies gaps in the current knowledge that the proposed study aims to address. Chapter 3 the proposed model for detecting traffic

signals is introduced. The section begins with an overview of the conceptual framework, followed by a detailed description of the model's architecture and components. It elaborates on the algorithms and computational processes involved in the model, explaining how each part contribute to the overall functionality. This chapter also discusses the rationale behind choosing specific technologies and methods for the model development. Chapter 4 is dedicated to assessing the performance of the proposed model. It presents the methodology used for the evaluation, including the criteria and metrics for measuring effectiveness. The results of the model testing are displayed and thoroughly analysed, comparing them with the performance of existing methods. A comprehensive comparative analysis is conducted to highlight the strengths and limitations of the proposed model in the context of real-world applications. Chapter 5 presents a comprehensive architectural analysis of the models utilized for detection, emphasizing their respective strengths and the differences that influence their performance outcomes. Chapter 6 summarises the finding and discusses the theoretical and practical implications of the results, considering what they contribute to the field of object detection and traffic signal recognition. The chapter also explores potential avenues for future research, suggesting how the gaps identified in the literature review could be further explored and how the model could be enhanced or adapted for other applications. Chapter 7 lists all the bibliographic references used throughout the research. It provides detailed citations for all sources, including books, journal articles, conference papers, and online resources, ensuring that the research adheres to academic integrity and copyright norms.

CHAPTER 2: LITERATURE SURVEY

With the increase in road fatalities, it is very important to develop a system that can help the drivers to stay alert while driving. So, the research to develop such a system is in demand in many countries, and many algorithms have been proposed to solve the problem. In this chapter we will be focusing on the various methods used in this field. Now the focus will be on looking at the various conventional methods and progress accordingly.

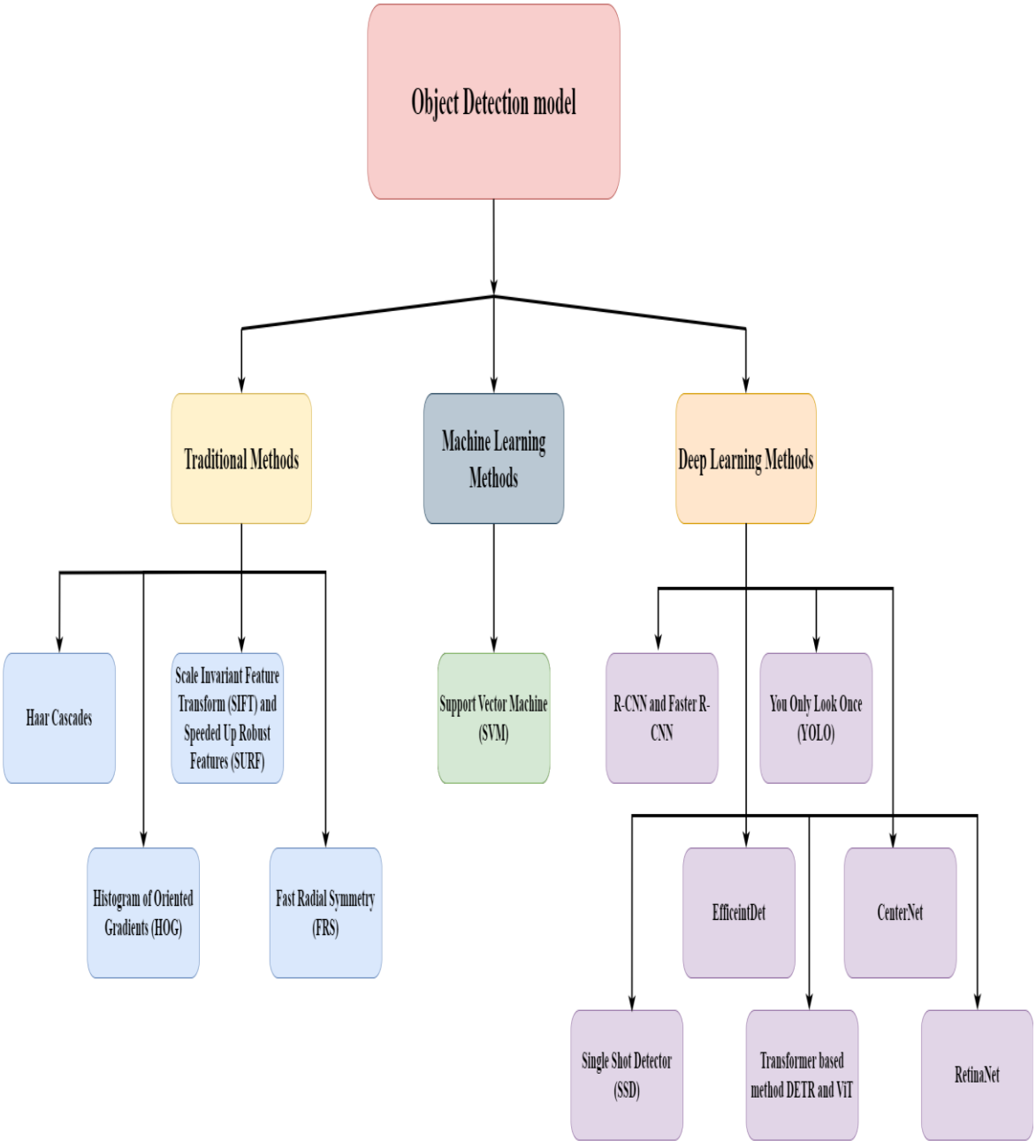


Figure 2.1: Various Object Detection Methods

2.1 Traditional Methods

2.1.1 Gaussian Distribution

One of the conventional methods used in detection is modeling the hue and saturation of traffic lights according to the Gaussian distribution. This approach is based on the observation that the colors of traffic light can be distinctly represented within the hue and saturation channels of HSL (Hue, Saturation, Lightness) or HSV (Hue, Saturation, Value) color spaces. So, by modeling the hue and saturation it becomes robust to identify the traffic lights in images or video streams under varying lighting conditions. The Gaussian distribution provides a statistical framework that accounts for the natural variation in the appearance of traffic light colors due to factors like sunlight, shadows, and reflections. A Gaussian distribution is defined by its mean and standard deviation, which represents the color's central tendency and the spread of color values around this mean. During traffic light detection, the hue channel captures the color information, and each traffic light color has a typical hue range. So, modeling these ranges allows the algorithm to filter out colors that don't match the expected traffic light colors. And the saturation channel measures the intensity of the color, as traffic lights have high saturation values, the Gaussian distributions in the saturation channel helps to differentiate the traffic lights that are bright, from other dull objects in the background. Yehu Shen et al. *[1]* have proposed a robust and efficient algorithm to detect traffic lights based on video sequences captured by a video camera. They have used this modeling of hue and saturation based Gaussian distribution to train their images. So the region containing traffic lights in the test images are extracted using the learned models. Then a post-processing method is applied to these regions and the shape information is taken into the account, and finally the results are obtained. As the algorithm works using a low-end video camera the image quality and the resolution may have limitations, also the algorithm may struggle to detect traffic lights that are at a large distance; also it might be difficult in complex traffic light scenarios containing arrow-based traffic lights.

2.1.2 Hough Transform

A commonly used and a traditional method to detect the traffic lights can be using the Hough Transform. The Hough transform helps in extracting the feature

extraction technique in the field of image analysis and digital image processing. It is an effective method as it can detect simple shapes like lines, circles, and ellipses. So, in traffic light detection the Hough transform can be applied to detect the geometric shapes corresponding to the actual structure of traffic lights. As a result, we can pinpoint the exact location and the boundaries of traffic lights within the region. Masako Omachi et al. *[2]* have utilized this concept for traffic light detection in an image by converting the color space from RGB to normalized RGB, then applying the Hough Transform on the region of interest that detects the shapes, especially circles to detect exact region of the traffic light. However the proposed method has some limitations like, it cannot address the issue of detecting both filled circles and open circles that can lead to false detection and also affect the accuracy of detection. Also, the paper does not provide a detailed analysis of the computational complexity which is important for real-time applications.

Chunhe Yu et al. *[3]* have proposed an algorithm for traffic light detection using image processing technology, specifically based on RGB color space. Their algorithm includes using a CCD camera, an image acquisition card, and a PC, and is based on the RGB color space, then the red, green and yellow objects in the image is extracted and it is verified using the identity features of traffic light. The paper focuses on various traffic light shapes which also includes the arrow shaped traffic lights which wasn't taken into consideration by the former author. The algorithm is also said to eliminate disturbances in the environment like vehicle lamps, vehicle bodies, and road signs however there is no thoroughly evaluated results. Although the approach in paper is said to provide a better advanced driver assistance system by recognizing traffic signals, it does not discuss the performance of the algorithm in challenging weather conditions of heavy traffic congestion.

Another approach in the field of detecting the traffic lights includes the mapping of the traffic signals. Traffic light mapping becomes crucial when it comes to the development of an autonomous driving systems and advanced driver assistance systems (ADAS). This method includes creating a precise and detailed map of the traffic light based on their location and orientation. Nathaniel Fairfield et al. *[4]* have used this method for the detection of traffic lights. Their work involves estimating the 3D position and orientation of the traffic lights by driving a car equipped with cameras, GPS, IMU, and lasers through intersections and collecting precisely time-stamped camera images. Then

the position of traffic light is estimated by triangulating from multiple views. The accuracy of the traffic light map depends on the accuracy of the position of the car. However, the camera calibration is important for the accuracy of the mapping pipeline. Although the system performs well, but it has few limitations like the performance is affected by heavy rain or suns glare that can obscure the camera's view and affect the accuracy. There is latency in transmission of images from the camera to the computer which impacts the real-time performance.

2.1.3 Fast Radial Symmetry Transform

Fast Radial Symmetry transform (FRS), is a powerful technique in computer vision for detecting symmetrical features within images. The FRS algorithm first scans the image to find out the radial symmetry, it is done by evaluating the image's gradient to find out those points. It then identifies the local maxima and minima which help in distinguishing the traffic lights from other urban elements. Then the radial strictness factor controls the strictness of the symmetry detection, i.e., a high strictness factor indicates that the algorithm will only consider structures that closely match the perfect radial symmetry. And finally, the algorithm is tuned to detect objects by setting the range of the radii, like for traffic signals or lights the radii parameter would be set accordingly. So, the FRS algorithm can be considered in detecting objects with symmetrical features, and its efficiency becomes crucial for real time applications, such as autonomous vehicles and ADAS system. In *[5]* George Siogkas et al. have proposed the use of a monocular video camera for the detection of traffic lights, which incorporates a color pre-processing module to enhance the discrimination of red and green regions. This improves the visibility and separability of the traffic lights. Then they have used the fast radial symmetry transform (FRS) to get the detection done. Although they have tested their algorithm in various conditions like day and night, but their algorithm does fail in heavy dense traffic conditions and also there are instance where there are false positives that affect the accuracy of working.

Ying Jie et al. in *[6]* have discussed an idea of developing an electronic aid for the visually impaired persons helping them in detecting the traffic light when on road. Their idea proposes to use a camera mounted on the hat of the use along with the embedded control unit and a speaker that can notify the use regarding the signal. So, for traffic signal detection they have used the basic method of detection that is by

recognizing the shape of the object. Their method includes recognizing the rectangular shape of the traffic signal first and then checks for the circular object that resembles the traffic light. It is basically done on the combination of shape and color-based features, once the shapes are extracted, the color distribution in HIS and RGB color spaces are analyzed, and finally the position of the circular shape within the rectangle is considered for recognition. This has helped them to achieve 97% recognition rate on an average. But the system has one concern i.e., the false detection of the circular shapes, and also the detection of high gray level in the environment.

2.2 Machine Learning Methods

2.2.1 Support Vector Machine

For the detection of objects like traffic lights, the Support Vector Machine can also be integrated with the image processing techniques that can help in automatic detections. This is a comprehensive approach to detect the signals accurately. So, if we look at the working process of this, we can see that the image goes through some pre-processing techniques like noise reduction and enhancement, followed by the feature extraction process where the Color Space Transformation and Shape detection method like Hough transform is used. This helps to separate the colors of traffic light from the background and helps to identify the geometric properties of traffic lights. Finally, the training is done with the help of SVM classifier, and the feature vectors determine whether the object is traffic light or not. In *[7]* Zhilu Chen et al. have used this SVM method for automatic detection of traffic lights. In corporation to the SVM, the authors have used the Hidden Markov Model (HMM) to improve the accuracy and stability of the detection results. The HMM enhances the performance by considering the sequential nature of traffic light states, and incorporating from previous frames. And one added advantage to this is, this is well suited for scenarios where the system's state transitions over time is not directly observable but it can be inferred though observable events. This proposed system has provided better performance compared to the previous system. However, their paper focuses only on the detection of red traffic lights and not the yellow and green lights. Also, the author has not discussed the computational complexity while detecting the performance real time.

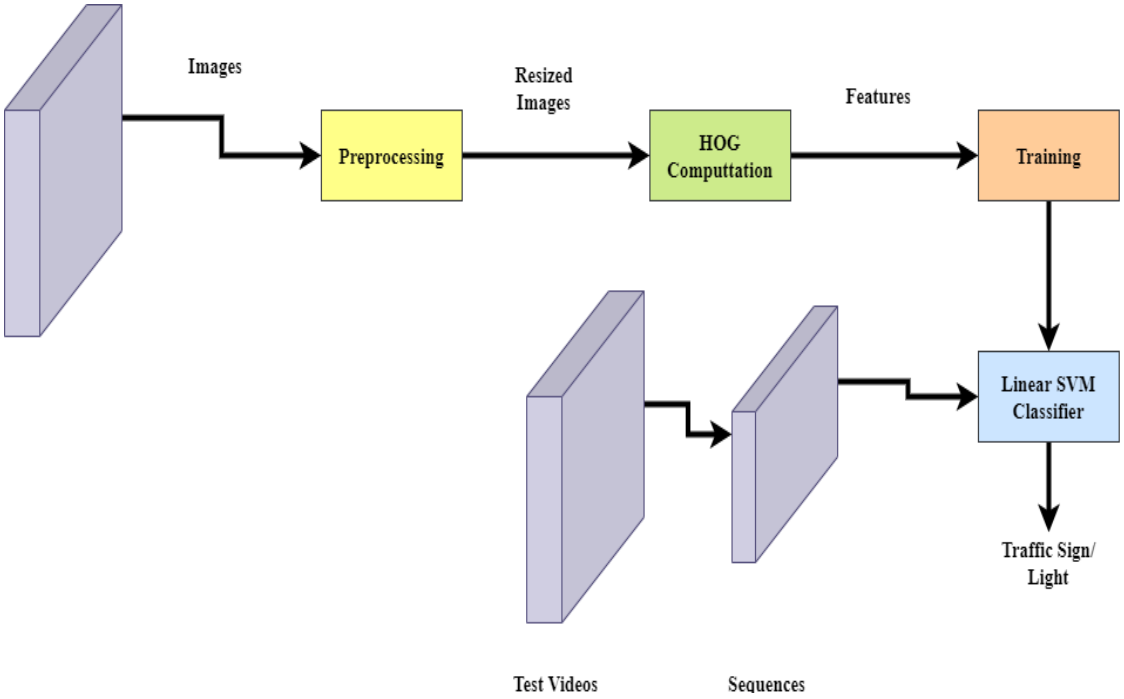


Figure 2.2: Traffic Sign/ Light detection using HOG and SVM

Florin Oniga et al. in *[8]* have discussed an algorithm for traffic light detection using the capabilities of a mobile device. For this they have used the previously mention technique of Gaussian distribution for hue, saturation, and lightness, by which they build the candidate region by clustering the connected candidate points. Then they filter out candidate regions based on the shape followed by a fast circularity check improves the accuracy of detection. Although their system achieves detection accuracy of 80-85% but it requires refinement for use in real traffic scenario. And another drawback to their system is use of high-resolution images which stresses on the computation power of the mobile phone.

An approach to integrate the existing methods like, Color Segmentation and Circle Hough Transform to detect the traffic lights. The color segmentation algorithm will separate the pixels in the image on the basis of its color, like for traffic light had red, yellow and green color. Then the Hough Transform will detect the circles in the image and highlight them. So, there is a potential benefit of making the system less computationally expensive compared to deep learning approaches. So, Dwi H. Widyantoro et al. in *[9]* have discussed the potential of using this approach, along with some extra algorithms like Gaussian filtering to remove noise from the image and smoothen the image, continuously adaptive mean shift method to avoid false negatives

and tracking loss in detection and recognition of the traffic lights, then normalized RGB color space and the RGB components for handling undesired effects of illumination changes. However, they have faced some limitation in their approach like, miss-detection of pedestrian signal as traffic lights and miss-detection of construction lights as yellow-red traffic lights or the flashing lights. Also, their system has no provisions for recognizing traffic lights with different shapes, and there are possibilities of facing challenges with complex lighting and background.

The next approach is about using the multi class learning and multi class object tracking. In the context of traffic light detection, multiclass learning would be used to classify detected objects based on the state of the traffic light. The multi class learning includes SVM, decision trees and ensemble methods like random forests. Now coming to the multi object tracking it basically monitors the movement and changes in the state of multiple objects over time. In traffic light detection system, tracking is done to detect the traffic light across successive frames of video. We will be focusing on this approach in the later chapters. Zhilu Chen et al. in *[10]* have presented this approach for accurate detection and classification of the different types of traffic lights. They have used the color extraction and blob detection to locate the candidates or regions for the traffic light detection. Then a pre-trained PCA network is used as a multiclass classifier to obtain the frame-by frame results. Also, they used an online multi-tracking technique to avoid the occasional loss. And finally, a forecasting method is used for filtering the false positives and to stabilize the outputs of the detection. Overall, it can be said that the proposed method is effective for detecting the green and red traffic lights of various types, but it has few limitations like the performance compromises in real world scenarios due to light conditions, motion blur, color distortion and variation in the scene. Also, it lacks research on detection during severe weather or night conditions. And the reliability of the system is affected by the video quality and camera configurations.

In *[11]*Asaad F. Said et.al have proposed a real-time approach for detection and classification of traffic light signals in complex traffic scenes by using a Lookup Table which is independent of lighting conditions. They have used an unsupervised classifier to accurately segregate the traffic signals for example circular, semi-circular, and arrow-shaped, so this does not require a training dataset. The primary advantage of this approach is the independence from labeled training data, which can be expensive or

time consuming. And the use of an unsupervised classifier enhances the versatility and adaptability of traffic light detection systems, making is reliable in diverse conditions. Contrasting on the limitations the system needs to be optimized more and also the addition of GPS would be and added advantage to it.

2.3 Deep Learning Methods

2.3.1 Convolutional Neural Network

Now moving on to the modern object detection techniques, the focus will on the development and the use of the techniques in the object detection field. With the recent development in this field the introduction of convolutional neural network started to dominate the object detection. The convolutional neural network is a powerful and effective for processing the data with a grid-like topology, like images. The CNNs contribution in field of computer vision, which has a significant advancement in the tasks like image classification, object detection, segmentation and beyond it. The architecture of CNNs is inspired from animal visual cortex which is designed as to automatically adapt to learn spatial hierarchies of features from the input images. Next looking into the core components of CNN, these include the convolutional layers, activation function, pooling layers, and fully connected layers.

Convolutional layers are the primary building blocks of a CNN. These are responsible for performing the convolution operations on the input data; this is done with a set of filters or kernel that produces a feature map. This convolution process helps the network to capture details like edges, textures, and patterns. This as a result preserves the spatial relationship between the pixels by learning the image features.

After every convolution operation the activation function is applied, which introduces non-linearity into the model, which enables it to learn complex patterns. Some commonly used activation functions are Sigmoid, Hyperbolic Tangent (Tanh), SoftMax, Rectified linear unit (ReLU) function. But out of these the Rectified Linear Unit (ReLU) is most preferred activation function in CNNs, as it helps to accelerate the convergence of training and solves the problem of vanishing gradient.

After the layers are reduced to the spatial dimensions of the input for the next convolutional layer, pooling operation is done. This decreases the computational load,

memory usages, and the number of parameters. The most commonly used filter for the pooling operation is the Max Pooling that selects the maximum element from the region of the feature map covered by it.

Finally at the end of the network, convolutional and pooling layers are followed by one or more fully connected layers. Then the layers are flattened and the high-level features are learned by the previous layers into a vector and perform classification based on these features. And this final layer uses a SoftMax activation function for the multi-classification tasks.

Coming to the advantages and the key features of CNN, the parameter sharing and the local connectivity is an added advantage, so in CNN, each filter is applied across the entire input. And due to parameter sharing, the total number of parameters is reduced; this makes the CNN less prone to over-fitting and makes it more efficient. The use of CNN helps to recognize the objects in an image irrespective of its position. So, this ability to handle translation invariance makes the CNN very effective to handle object detection and recognition tasks. Another added advantage of using CNN is the ability of automatic feature detection and extraction, thereby reducing the need for domain expertise.

2.3.2 Region-based Convolutional Neural Networks (R-CNN)

With the growing advancement in the CNN, a lot of new development and improvement is seen, this includes the development of region-based Convolutional Neural Networks (R-CNN), and this combines the convolutional neural network with the traditional object localization methods. The fundamental idea behind this concept is first a set of candidate region is proposed that are likely to contain the objects, then a CNN is used to extract features from this proposed region, and classify each region into different object categories. So, the key components of R-CNN include the proposal of regions, the algorithm generates around 2000 regions from an input image. These regions have variable-sized bounding boxes that may contain objects. Then the feature extraction takes place where every region is resized to a fixed size and then passed through a pre-trained CNN to extract a feature vector. The extracted feature vectors are fed into a set of class-specific linear Support Vector Machines (SVM), classifying the regions into object. This SVM is responsible in distinguishing an object from the

background. And to improve the precision of the bounding boxes, a linear regression model is used to predict those adjustments in the proposed region.

2.3.3 Faster Region-based Convolutional Neural Networks (R-CNN)

Now moving to the developments and advancements in R-CNN, the introduction of region proposed network (RPN) has allowed almost a cost-free region proposal, this significant enhancement has been introduced in Faster R-CNN. This is one of the major developments in the R-CNN architecture and it has led to improved reduction speed and accuracy. Focusing on the key components of Faster R-CNN includes proposed network (RPN) which is responsible for generating high quality region proposals which is used by the detection network, this replaces the select search method used in R-CNN. Thus, speeding up the proposal generation and the system is allowed to be trained end-to-end. The next feature is the shared convolution layers, for detection, the RCNN uses a set of shared conversion layers for both RPN and the detection network as a result the deficiency also improves and the overall performance is also enhanced. The next feature is the anchor box, the anchor boxes are nuclear defined boxes of various skills and different aspect ratio and it serves as a reference for the location of objects in the image. And finally, the integration between RPN and faster R-CNN, After the region is proposed by the RPN, the fast R-CNN detector utilizes this region to extract the features using the region of interest (RoI) pooling. Then these regions are classified by the system and the bounding boxes are refined. Thus, this integrated process is efficient in detecting the objects with high accuracy in real time.

Zhongrong Zuo et al. in *[13]* have discussed the use of Faster R-CNN to detect traffic signs. Out of the two-training method i.e., alternative training (alt-opt) and approximate joint training (end to end) the authors have chosen alt-opt method to train their model. And have considered the VGG16 to train the model as it is used for large datasets, modifying the network as per their need. On successfully testing they achieved a mAP of 0.3449 which is not the best, and requires more improvement. Also, they have faced the issue related to the accurate positioning of the bounding boxes in the detected traffic signs. This might be due to the less no. of epochs used while training.

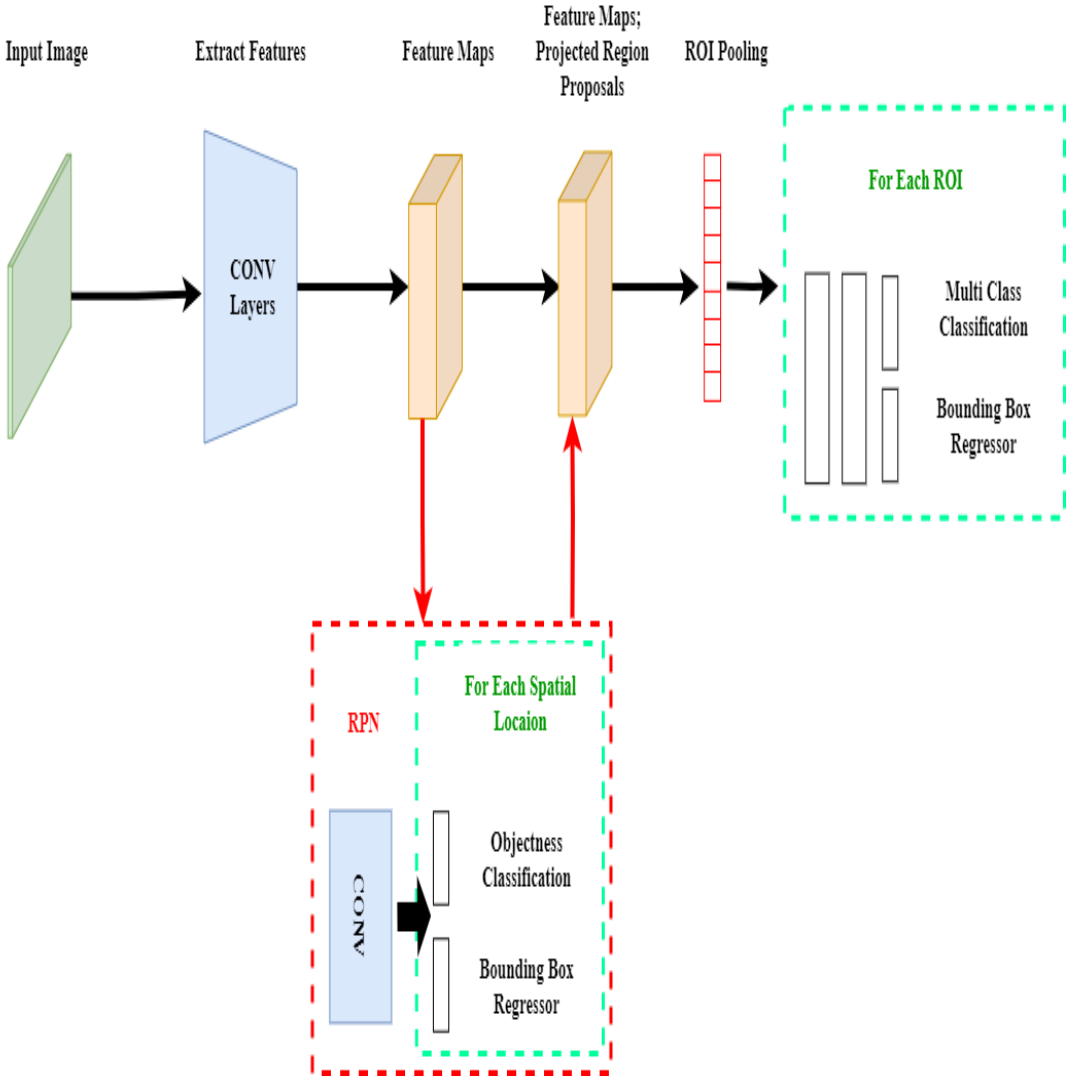


Figure 2.3: Architecture of Faster R-CNN

In *[14]* Fei Wang et al. have proposed an improved Faster R-CNN algorithm for Traffic sign detection. They have used a new sampling method to optimize the network, that is done be considering high quality region proposals and a post processing scheme for re-sampling the false samples. Their method also involves tasking ResNet into consideration as the backbone of their network, as it provides multi-scale features at the ground or the small level, and it increases the range of the receptive fields for the networks. The authors have used the Tsinghua-Tencent 100K dataset for their evaluation, and got a recall of 87%. However, the research doesn't provide a detailed analysis of the limitations.

Emin Güney et al. in *[15]* have implemented Faster R-CNN to detect traffic signs and Road objects for the safety of the driver. They have prepared the dataset of 1880

images from the roads of Turkey and Germany, which was divided in to 80:20 ratios for training and validation process. On comparison to other models, they achieved an accuracy of 88.99%. Their model has been able to detect 22 different traffic signs, people, bicycles, and vehicles. Apart from the traffic sign detection, the authors plan to add lane violation and other traffic scenarios to their model.

In *[16]* P. Kanchanamala has proposed a traffic light detection system for self-driving cars. The authors have discussed the various challenges faced during the traffic signal detection, they have studied various algorithms used to detect the traffic signal like Support Vector Machines (SVMs), Convolutional Neural Network (CNN), Faster R-CNN, YOLOv3, and Adaboost Algorithm, out of which they found SVM to be complex while dealing with high dimensional inputs, and time consuming annotation process in YOLOv3, feature extraction in Adaboost algorithm is also time consuming, and in R-CNN the requirement of large amount of data, as it is sensitive to the variations in image quality, making a difference in the performance, thus makes it challenging to be used in real-time applications.

2.3.4 Single Shot Detector

The next object detector framework is a single shot multi box detector It was designed to perform object detection efficiently as it is executed on a single, deep neural network pass and. This technique combines the strategies that allow it to predict the object class and the location of the bounding box directly from the images in one step. The working principle of SSD is discussed as follows:

At first, the SSD starts with the convolution network at the base which extracts the feature maps from the input image. The base network is generally pre-trained network like the VGG 16 which is modified to remove the fully connected layers and is adjusted to improve Performance of detection for small objects. The low-level features are captured by the early network layers, and the high-level features are captured by the deeper layers.

The SSD adds several conclusion layers of decreasing size on the top of this base network. These are the additional layers which enable the object detection at multiple scales. As a result, of these layers a feature map of different resolution, is produced

which allows the model to detect objects of various size just by looking at the proper feature map scale.

Now in each feature map, SSD defines a set of default bounding box or the anchor box of various aspect ratios. These anchor boxes are important for capturing the objects of different shape and size across different classes. So, for each anchor box the class probabilities which include the class for the background and the adjustment to the bounding box coordinates is predicted by this SSD.

Now the SSD uses a small convolution filter to predict the bounding boxes and class probabilities associated with it. These predictions are made for every bounding box in the image or the feature map which results into a large set of potential reductions of various scales and aspect ratios.

Now as the SSD is capable of generating multiple predictions per object which may also include many overlapping boxes for the same object a new technique called non maximum suppression is used in such cases. So only the confident prediction of each object is kept by the non-maximum suppression technique and the rest weak overlapping boxes is removed. Thus, this is a crucial step in reducing the number of proposals while focusing on the main object location and enter

Now the training is done during which the SSD deals with significant imbalance between the positive and negative objects. So, for this the SSD utilizes a hard mining technique where all the negative examples are sorted using their confidence loss and the positive ones are selected and the ratio of fixed size is maintained between them. So, the model learns from this challenging situation.

Now to improve the performance and the robustness of the model and various augmentation techniques like cropping flipping coma scaling of input images is used by the SSD. These data augmentation techniques help the model to detect objects under various conditions.

The main advantage of using SSD is the use of a single forward pass to perform object detection enhancing its efficiency and effectiveness in the detection. So, the combination of deep neural network for feature extraction, and the systematic approach to manage the class imbalance and model training makes SSD a versatile and powerful object detection framework suitable for real time applications.

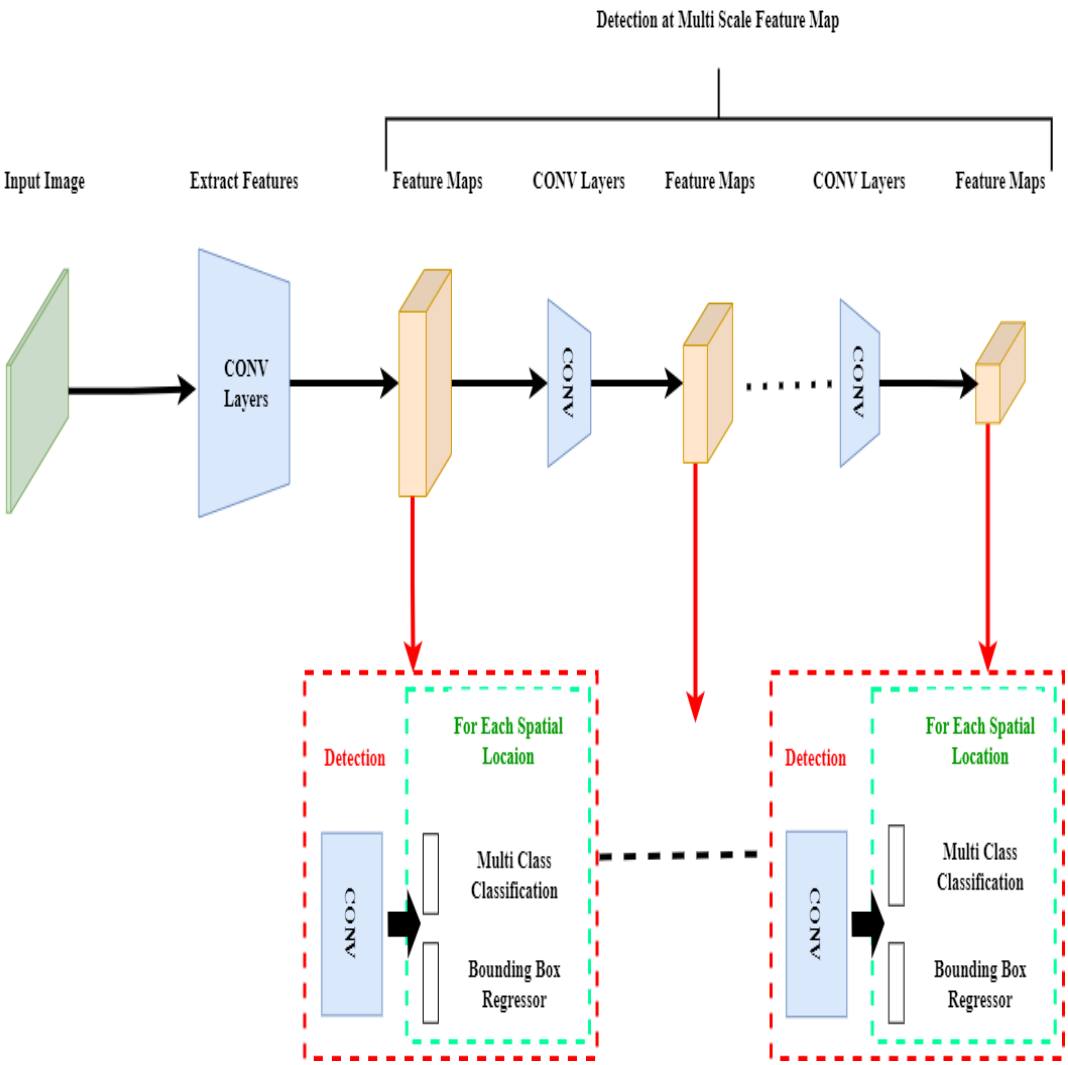


Figure 2.4: Architecture Design of SSD

In *[17]* Benhe Gao Et Al. have discussed their SSD algorithm for traffic sign detection, where they have trained and tested their model on German Traffic sign dataset which is based on PascalVOC. They have modified the SSD network in such a way that the feature map of the last six layers is considered for the detection, and the convolution Colonel helps them to gain the confidence value and offsets of the prior box. While testing their model, they achieved a precision of 91.09% and recall of 88.06% on their data set. As their evaluation was based on a German dataset the results may vary across the other data sets or the real-world scenarios. They have also not discussed the competition complexity while detecting making it difficult to access it relative effectiveness.

Shuai You et al. *[18]* have proposed and algorithm to detect the traffic signs on the roads using a lightweight SSD network. The authors have used TT100k dataset for their work, their algorithm firstly detects the locations and marks it as an alternative area, and then the marked coordinates are cropped and resized to 240x240 pixels. Then a color detection algorithm analyzes the cropped images to detect the primary colors of the Chinese traffic sign. Finally, a component analysis is done on the binary images, and the largest connected component's area is considered, and on surpassing the threshold, the detection is labeled as accurate else considered as a misdetection. The proposed method yields a mAP of 0.73, but it does not discuss the performance in complex scenarios where the traffic signs are partially occluded.

R. Krishna Paavani et al. in *[19]* have proposed a traffic sign board reduction using single shot detection, where they have discussed the object detection using SSD and have achieved better results than the previous models like R-CNN and Faster R-CNN in terms of detecting small objects with high accuracy and speed, and achieving a mAP of 78%. However, they have not mentioned any comparison or benchmark with other traffic sign detection models that makes it difficult to judge the performance of their system.

2.3.5 You Only Look Once (YOLO group)

In the field of object detection, one of the best algorithms developed is You Only Look Once (YOLO). It is a revolutionary approach in this field, and is significantly different from the region proposed based methods like R-CNN. The YOLO has been developed by Joseph Redmond et al. *[20]* and is a single step approach that allows it to achieve high speed and making it suitable for real time application. Now looking at the core principles of YOLO, YOLO uses a single neural network which is used to predict the multiple bounding boxes and the class probabilities simultaneously. This is a better approach than those two-stage detectors that first proposed the region and then classifies them. Yolo also has the feature of spatial division, where the input image is divided into a grid. And the grid cells are responsible for predicting the bounding boxes and the probabilities of the objects that fall within the cell. And the entire region is considered for making the prediction which helps in reducing false positives. In YOLO each bounding box prediction includes coordinates like center, width, and height, then confidence score and class probabilities.

Now moving on to the evolution of the YOLO algorithm, there has been significant development with time, the very first version of YOLO is the YOLOv1, that was revolutionary for framing object detection as a single regression problem and directly predicting the bounding boxes and the class probabilities from full images in a single network pass. The YOLOv1 used a simple convolution network where an input image was divided into grid with each cell responsible for predicting the bounding boxes and probabilities for objects whose center falls within that cell. However, YOLOv1 struggled with detecting small objects and had a lower recall compared to two stage detectors, and it also was prone to localization errors.

The next model follows is the YOLOv2 or YOLO9000, it aimed to improve upon the YOLOv1's accuracy while maintaining high speed. It introduced batch normalization, higher resolution inputs, and anchor boxes to better predict the bounding box shapes. An ambitious aspect of YOLOv2 was its ability to detect over 9000 object classes by combining data from multiple sources, a significant leap in class detection capability. In terms of performance YOLOv2 achieved better recall and precision especially for smaller objects, without sacrificing the real time performance that defined the original YOLO.

The next model is the YOLOv3 which further refine the detection accuracy, particularly for small objects, by making predictions at three different scales using deeper and more complex network architecture. The YOLOv3 introduced the Darknet-53 backbone, replacing the Darknet-19 which is used by the YOLOv2, which provided a good balance between the speed and accuracy. However, with the improvements in the layer architecture and the training techniques, YOLOv3 was more versatile and accurate across a broad range of object size and types.

In [21] Ruben J Franklin et al have proposed the use of YOLOv3 for traffic signal violation detection. It aimed at scenarios like signal jump, vehicle speed, and count detection. The YOLOv3 provides three features that class of the object, bounding box, and semantic segmentation that is used for determining the traffic violation type. For calculating the speed of the vehicles, the system uses the pixel calculation area of the bounding box and the anchor points so; the rate of change of the object frame by frame and the actual distance covered by the vehicle is taken into consideration to determine the speed. The authors have trained their model on MSCOCO dataset and have

achieved 97.67% accuracy for vehicle count detection and 89.24% accuracy for vehicle speed detection. However, the authors did not address the issue of class imbalance that might affect the performance of the detector Also, the computational time for detecting the vehicle speed is 5.4 times slower than the actual speed and the authors also did not discuss the robustness of the proposed system under various environmental conditions.

Research carried out for detecting and localizing the traffic lights using YOLOv3 and Stereo Vision is proposed by Wael Omar et al in *[22]*. The research includes an indicated traffic light distance measurement system using YOLOv3 or detection and Stereo Vision for 3D traffic light localization. The authors have used HSV classification method for classifying the traffic light color. The authors have experimented and found out better results within 20 meters from the sensor while getting some misdetections and classification errors at further distance. However, the authors faced some errors in the map generation by stereo vision so the use of unscented Kalman filter (UKF) is preferred as it would improve the prediction by combining the stereo camera disparity map and lighter readings.

Branislav Novak et al in *[23]* have proposed the YOLOv3 algorithm with some additional convolution neural network for traffic sign recognition. They have utilized the YOLOv3 algorithm for detecting objects like car, trucks, pedestrians, traffic signs, and traffic lights and used the CNN for traffic sign recognition, combined together into a single component. The authors have used Belgium TS dataset and the German traffic sign recognition benchmark combined into a single dataset for their proposed algorithm. As the YOLOv3 detects the object from the video frames, the CNN processes the output to recognize the traffic sign, and they have achieved a classification confidence of 99.2% accuracy. The paper however does not provide the details of handling complex scenarios and have also not provided a detailed evaluation of the performance of the proposed solution.

As it is very challenging to detect the traffic sign with better precision Huibing Zhang et al in *[24]* have proposed a real time detection method for small traffic signs based on YOLOv3. For this, they have used the MSA_YOLOv3 which is a modified version of the traditional YOLOv3 that supports multi scale architecture which is helpful in detecting small objects such as traffic signs. They have tested the detection performance of MSA_YOLOv 3 on the TT100K dataset, and have found to have a

mAP of 0.863 whereas the general YOLOv3 had a mAP of 0.804. As they focus on detecting small traffic signs, they missed out on testing their performance on medium and large sized traffic signs. Also, the detection speed and computational complexity for MSA_YOLOv3 is more than the YOLOv3.

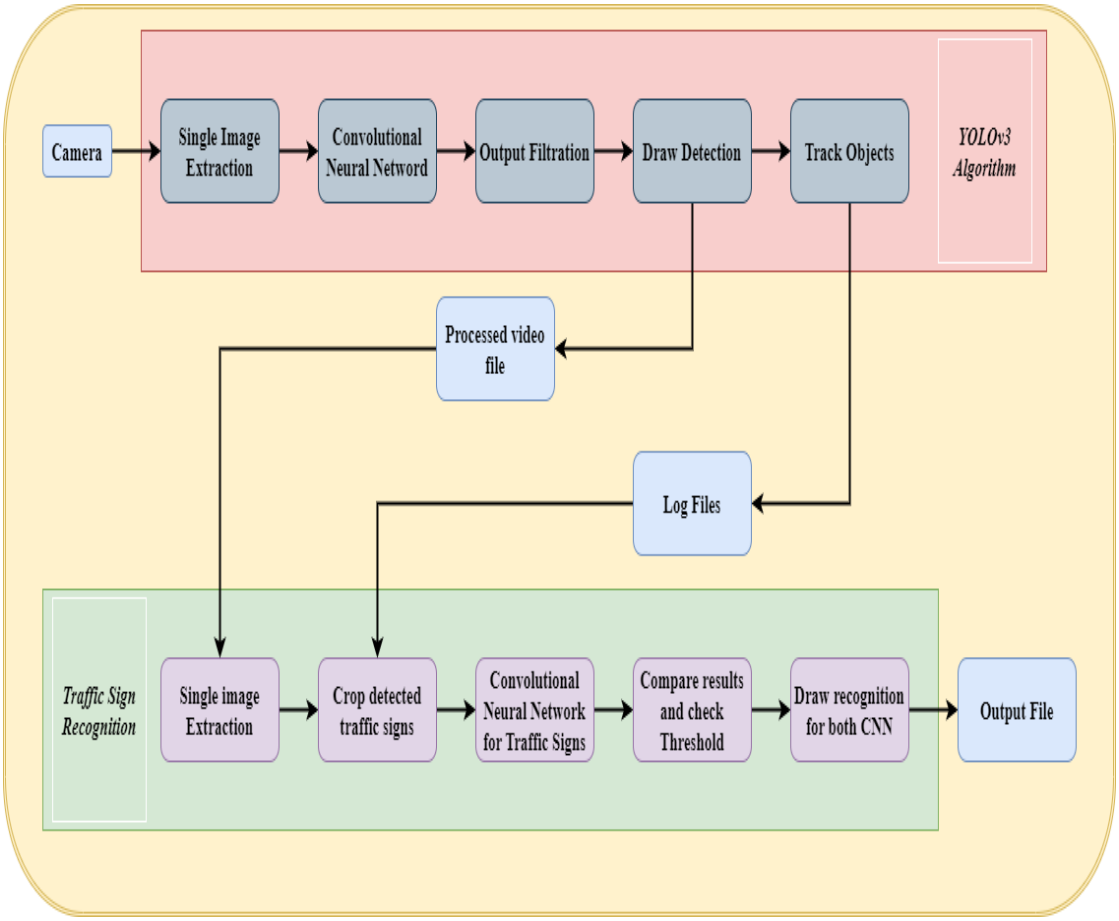


Figure 2.5: YOLOv3 algorithm for Traffic Sign Detection

Another work carried out to detect the traffic signs are done by Wan Noorshahida Mohd-Isa et al in *[25]*. For this they have modified the YOLOv3 framework by introducing the SPPNet which allows multi size training and identification of far and small objects. They have implemented their proposed algorithm for the Malaysian traffic signs and have achieved map of 82.5%. However, the proposed algorithm has a few limitations to it, like the integration of SPPNet with YOLOv3 has a chance of over-fitting their model if not properly tuned, so it requires careful architectural adjustment so that it does not affect the overall detection performance. Another limitation can be that generalizability to the findings of the traffic signs and other countries as they are restricted to only Malaysian traffic signs.

Another modification in the research of traffic sign recognition and detection using YOLOv3 has been proposed by Ao Qin et al in *[26]*. The modification that they introduce is based on the Feature Pyramid Network (FPN). The use of FPN is basically to enhance the object detection models by leveraging the inherent multi scale, pyramidal hierarchy for deep convolutional networks. The FPN generally uses two processes that is top-down process and the lateral connection. With the integration of Yolo V3 and FPN, the YOLv3 uses the concatenation method that combines the up sampled, high-level features with the low-level features instead of using the element wise addition. The author used the German traffic data set for testing their algorithm, and they also tested on other benchmarks algorithm like Faster R-CNN, improved Faster R-CNN, YOLOv3+FPN and improved YOLOv3+FPN where they have found a mAP of 94.24% for the later, and it performed well when compared to the other mentioned frameworks.

An improved traffic regional light recognition algorithm based on YOLOv3 has been carried out by Feng Yu et al. in *[27]*, where the author proposes to use the SimAM attention mechanism for enhancing the features and to improve the accuracy of detection. The paper focuses on adding the SimAM attention mechanism to the FPN network of YOLOv3 which helps in redistributing the weights of the featured layer so as to emphasis on the main features while suppressing the secondary features. In addition to this the algorithm adds the SSH Network to enhance the receptive field and strengthen the feature extraction. The proposed algorithm was tested on the TT100K dataset and on testing the YOLOv3-SimAM resulted in a mAP of 67.67% which is better than the general YOLOv3 by 1.2%, SSD300 by 34.54%, but it was outperformed by the Faster R-CNN by 4.64%. According to the experimental results, it can be seen that the algorithm requires more study and more refinement to be considered for real time applications.

In *[28]* Chunpeng Gong et al. have discussed the traffic sign recognition based on the YOLOv3 algorithm. They propose a spatial pyramid pooling structure into the YOLOv3 network for fusing the local and the global features together. This helps in solving the problem of extracting same image feature space which ultimately improves the detection efficiency. They have also utilized the DIOU loss for stable target fame regression that considers the distance between the target and anchor, the overlap rate and the scale of the image. They have tested the algorithm on the TT 100K data set and

have achieved map of 77.3% that is higher than the general YOLOv3 by 8 .43%. However, their research requires more improvement as they have not focused on the real time performance and the computational complexity of their algorithm.

S. Vinothkumar et al in *[29]* have proposed a hybrid network of YOLO and ResNet for traffic sign detection. They have used the YOLOv3 algorithm for object identification and traffic sign detection in real time and ResNet for extracting the features from the training data set and used for classification. So, the hybrid algorithm combines the strength of the both to improve the efficiency and accuracy of the traffic sign prediction. The test results of the proposed algorithm are said to outperform the other algorithm in terms of accuracy. However, there has been no such data found in the paper, so it is difficult to analyze the real performance of the proposed algorithm.

The goal of the YOLOv4 was to maintain accuracy while making the model faster and easier to utilize for training on traditional hardware. Numerous novel approaches were integrated into it, such as PANet for feature aggregation, spatial pyramid pooling, and CSPDarknet53 as the framework. Maintaining good detection performance while making sure the model could be effectively trained and operated on generally accessible hardware was one of YOLOv4's core objectives.

Tai Huu et al. in *[30]* have discussed novel research on Traffic Light Detection using YOLOv4 in real-time. Their proposed method includes detection, classification, optimization and main state identification. They have used YOLOv4 by controlling the various hyper-parameters, for the detection and classification of the traffic lights. After the detection and classification, they have proposed a post processing stage where the algorithm removes the irrelevant predictions like occluded traffic lights. For the proposed method they have used the dataset from Korean highways and urban areas of Seoul city consisting of 4994 images. On evaluation of their model, they achieved accuracy over 90% at 30fps. Removing the occluded images from the training and validation set is not a wise decision, as these kinds of complex scenarios are common on road.

An improved traffic lights recognition algorithm for autonomous driving in complex scenarios is proposed by Ziyue Li et al. in *[31]*. The objective behind the research aims in recognizing the traffic lights in extreme illumination and complex road scenarios. To improve the accuracy of detection the authors have integrated the deep learning and

multi-sensor data (MSDA) to create a fusion, and have employed the YOLOv4 for the feature extraction and detection purpose. They carried out their test on the data from the roads of Beijing, and achieved a mAP of 92.5%. The authors aimed at testing the model at different conditions but have not done as they wanted, i.e., testing at different weather and lighting conditions. Moreover, they have also not discussed the marking method for the traffic light locations in the map that may affect the accuracy of the map generation.

A comparison of between YOLOv3 and YOLOv4 has been done by Hui Gao et al. in *[32]* for the traffic signal detection. For the experiment they considered the TT100K dataset in which they had 182 categories, including the data enhancement methods. On training and testing the data on YOLOv3 and YOLOv4 they found out that the confidence score of YOLOv4 is better than YOLOv3, providing a good performance. They found out the model to have a recall of 75% and an accuracy of 80%. However, they missed on performing a detailed study of the performance between the two models and with other existing models. Also, they have not considered various weather conditions for testing, like bad weather, poor visibility, which might also affect the performance of the model. They have not discussed the computational complexity required by the two models.

As YOLOv4 has provided some significant results, the development in the layers were tested, which introduces the modified of the improved version of YOLOv4. Based on it, Zhang Gan et al. in *[33]* discussed the use of improved YOLOv4 in Traffic Sign recognition. Their proposed method includes adding a new cross-layer connection to the traditional YOLOv4 network that adjusts the weights of the feature map during the feature fusion, thus enhancing the feature extraction ability of the network. They carried out the experiment based on their proposed method on the TT100K dataset, and achieved a mAP of 87.85% at 31.25fps. The authors have slightly missed by not providing the details of the computational resources that is required for the training and testing the modified YOLOv4. Apart from that they also did not consider the impact of occlusion and lighting conditions on the performance of the model.

Xiru Wu et al. in *[34]* have proposed a Traffic Sign Detection Algorithm based on Improved YOLOv4. The authors have proposed an improved lightweight YOLOv4 replacing the traditional feature extraction network of YOLOv4 with MobileNetV2, this

improves the speed of detection and also reduces the parameters of the model. The MobileNetV2 also adds a focus mechanism in the backbone network making it very effective during feature extraction. The authors experimented, their model on the TT100K dataset and they got a mAP of 82.3% which is less than 2.5% as the YOLOv4 has mAP of 84.8%. They also compared the improved YOLOv4 with the SSD300 and Faster R-CNN, where the modified YOLOv4 outperforms the SSD300 by 17.4% but is slightly less than the Faster R-CNN by 4.1%. The authors aim to improving the algorithm accuracy, and also considered the complex lighting and weather conditions.

A study done by Ying Zhao et al. in *[35]* on detection and recognition of Traffic Lights is based on Improved YOLOv4. The authors have used the lightweight ShuffleNetv2 replacing the CSPDarknet53 network of YOLOv4. The K-means clustering is applied to generate the anchor boxes, so that the sensitivity issues of outliers can be avoided. Also, the introduction of CS2A mechanism fuses the spatial and the channel attention, enhancing the location information. The authors used the dataset from Shanghai Jiao Tong University having 5786 images, and on testing they achieved an increase in mAP by 4.70% and 6.05% for CS2A over CA and CBAM mechanism. Their modified YOLOv4 algorithm reduces the model size by 81.97% and the detection speed increased by 16% and the recognition effect for small targets increased by 37%. Talking of the limitations faced by the authors includes the real-world scenarios like detection in varying lighting conditions, and the complex backgrounds, that might affect the detection and recognition of the traffic lights.

The ultralytics created the YOLOv5, which has grown in popularity because of its functionality and simplicity of usage. It has enhancements to model scalability, accuracy, and speed. Advanced data augmentation approaches, automatic learning rate scheduling, and model variants of varying sizes to balance speed and accuracy are some of the enhancements brought forth by YOLOv5. YOLOv5 has emerged as a popular model for real-time object identification tasks among developers and researchers because to its focus on practical applications and usability.

The Traffic Sign Recognition Application proposed by Daria Snegireva et al. in *[36]* uses the YOLOv5 architecture to do real-time object detection of the road signs. They have used a dataset of 2000 images based on the Russian Traffic Sign. As they faced issue with wrong annotations, they used an open-source tool called the Visual Object

Tagging Tool (VoTT) to help them provide correct image tagging. They conducted an experiment on the various YOLOv5 models i.e., from YOLOv5s to YOLOv5l and they found out YOLOv5l to perform better in terms of accuracy and got a mAP of 72.29% with a training duration of 300 epochs. However, the results are not that satisfactory and have a scope of improvements.

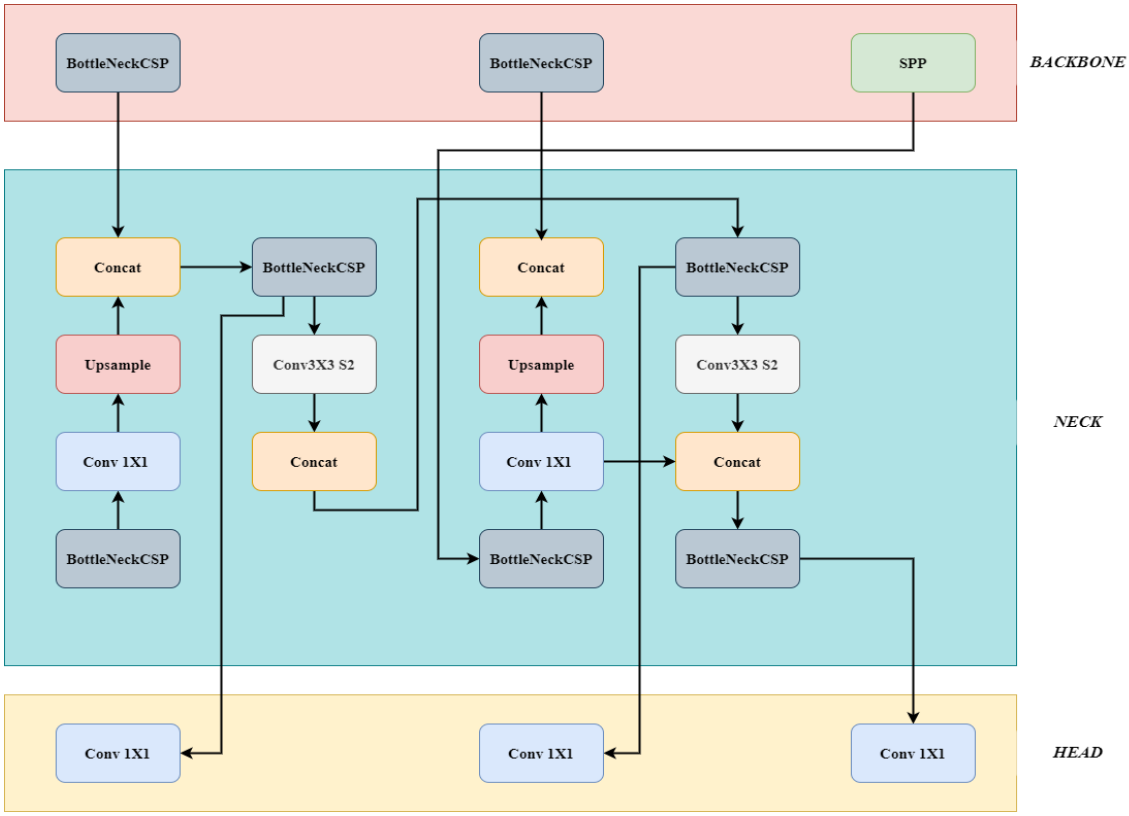


Figure 2.6: YOLOv5 architecture

Another application to the use of YOLOv5 includes the Real-time Signal Light Detection in the Railways, proposed by Wentao Liu et al in *[37]*. The idea behind this research is to improve the safety and efficiency of the Railways in terms of signal detection, as it becomes complicated to detect the signals due to the small size and complex environment. To conduct their experiment the authors collected the image samples real-time, and had 3000 images to do their task. For better performance data enhancement techniques like Adaptive anchor frame, Mosaic data enhancements were used. Finally, the training was done using the YOLOv5s for 300 epochs, and they achieved an average recall rate and accuracy of 0.972. As there has been no comparison with other existing methods it is difficult to judge the superiority of this model.

Traffic Sign recognition based on deep learning proposed by Yanzhao Zhu et al. in *[38]*, focuses on improving the performance and accuracy in detecting the traffic signs using the YOLOv5 and SSD model. To conduct their research the authors selected eight classes of traffic signs and collected them from the roads, and their dataset has 2182 images of the eight classes. On implementation they achieved 97.7% mAP with YOLOv5 and 90.14% mAP with SSD, where it can be clearly seen that the YOLOv5 performed better than the SSD in terms of the recognition speed with 30fps. The authors have missed out on mentioning about the complex scenarios like occlusions in various lighting conditions, and also did not discuss on the computational requirements.

Another research carried out by Chaunxi Niu et al. in *[39]* discusses about the used of YOLOv5s and AlexNet in detecting the Traffic light. The objective of the author is to use YOLOv5 for the target detection and AlexNet for recognition judgment. Their dataset contains 4200 images based on Traffic lights collected online, and to enhance the model performance they have optimized the dataset using the ZeroDCE low-light enhancement technique, followed by some data augmentation techniques like adding salt-and-pepper noise, and Gaussian noise. On conducting the tests using the YOLOv5 and AlexNet and successively comparing it with the results of other models like YOLOv4 and the Faster R-CNN they found YOLOv5 to perform better with an average precision of 99.46% and an average accuracy of 87.75%. YOLOv4 had the worse results in locating the traffic lights, followed by Faster R-CNN which could detect the traffic lights but deviated from the position at time, whereas the YOLOv5 detected the traffic lights accurately with high confidence. And the traffic signal recognition by AlexNet outperforms other models like VGG16 and ResNet in terms of accuracy. As they had tested on a homogenous dataset, they cannot rely on the results, moreover they did not discuss the challenges and complexity faced while using YOLOv5 and AlexNet.

Another notable work done by Hua Khanh Huy et al. in *[40]* proposes Traffic Light Detection and Recognition for Autonomous Vehicle in Robot Operating System 2 (ROS2) using modified YOLOv5. The authors used a free dataset available in the Roboflow dataset based on the data of Can Tho city of Vietnam. The data is then processed using various data augmentation methods like flipping, rotation, blur, random cropping, and noise production. The whole data is then prepared for training with the modified YOLOv5, by passing through the K-means clustering, which helped in reducing the training time and speeding up the process. The authors have then

considered testing the model in a simulated environment Gazebo of ROS2, where they found the model to have an accuracy of 78%, which is not that great. Although the use of K-means clustering for training with YOLOv5 improved the training speed by 24% but the end results need better accuracy to be considered for real-time applications.

A study carried out by Shubham Gore et al. in *[41]* describes the application of YOLOv5 algorithm for traffic sign detection, specifically focusing on the Indian roads, where the traffic conditions are quite challenging. Their research consisted of ten traffic signs commonly viewed on roads, and training them using YOLOv5 model. On evaluating their model, they achieved a mAP of 0.817. As, their research is limited to only ten traffic signs, it cannot be considered for a general case, and they have also not provided a comparison with other state-of-art methods for the detection.

YOLOv6, which outperformed its predecessor YOLOv5, achieved greater accuracy on the MS COCO dataset. On an NVIDIA Tesla T4 GPU, the fastest variant, YOLOv6-N, can achieve over 1200 FPS while keeping decent detection accuracy. Numerous architectural enhancements are brought about by YOLOv6, such as the innovative Bi-directional Concatenation (BiC) module and the Anchor-Aided Training (AAT) technique. Better detection is achieved without slowing down the model thanks to the BiC module's enhancement of localization signals. By combining the advantages of anchor-free and anchor-based detection paradigms, AAT increases effectiveness.

Among real-time object detection methods, YOLOv7 is said to reach the highest accuracy. Reaching 56.8% AP (Average Precision) on the MS COCO dataset, it shows a notable improvement over earlier versions. The "Trainable Bag-of-Freebies Sets" method used by YOLOv7 enables the model to pick up new skills while being trained, thus augmenting its capabilities. Moreover, YOLOv7 makes use of a cutting-edge compound scaling technique that preserves optimal structure even as model size is raised, making it flexible for a range of applications. Similar to YOLOv6, YOLOv7 places emphasis on swift and precise object recognition, rendering it appropriate for applications like autonomous vehicles, unmanned aerial vehicles, and robots.

Avinash Padmakar Rangari et al. in *[42]* have discussed deep learning-based traffic light system using YOLOv7. The main objective of the author is to manage the traffic density and Indian roads for which they have placed a camera at the junction of the roads to capture the data and simultaneously process them using YOLOv7. The

YOLOv7 algorithm helps them to count the vehicles on the road and then decides the color of the traffic light to be displayed. The extended efficient layer aggregation network (E-ELAN) architecture used by the YOLOv7 helps it to learn more effectively and efficiently. The authors have achieved an accuracy of 92% which was better than the previous models. However, they have not provided a detailed analysis of the performance in real world traffic scenarios such as complex traffic patterns or in harsh weather conditions.

CHAPTER 3: STUDY ON METHODOLOGICAL ADVANCES IN TRAFFIC SIGNAL IDENTIFICATION

This Chapter focuses on building a smart system that will detect the traffic lights on the roads of India. To develop such a system a novel method of object detection like YOLOv8 is being used. The YOLOv8 is one of the latest algorithms developed in the YOLO family and has some significant improvements in it object detection speed and accuracy compared to its predecessors. The architecture of the YOLOv8 is divided into three parts and is discussed as follows:

The backbone of YOLOv8 employs CSPDarknet53 as its foundational architecture, setting itself apart with an assembly of 53 convolutional layers, further enriched by the inclusion of three unique residual blocks. At the core of this architectural marvel is the Cross-Stage Partial (CSP) module, serving as the linchpin in the sophisticated process of feature extraction from the provided data inputs. The CSP module is ingeniously designed to split into two distinct pathways: the main branch and the auxiliary residual branch. This strategic division not only enhances the depth and richness of feature extraction but achieves this without the need for a substantial increase in the overall number of model parameters. A standout feature of this architecture is the deployment of focus layers, a clever tactic that effectively diminishes the spatial size of the feature maps whilst simultaneously expanding their channel density. Through the seamless amalgamation of these cutting-edge components, CSPDarknet53 significantly bolsters the efficacy of YOLOv8, propelling it towards greater speed and accuracy in detecting objects, and setting a new benchmark for performance in the realm of object detection technologies.

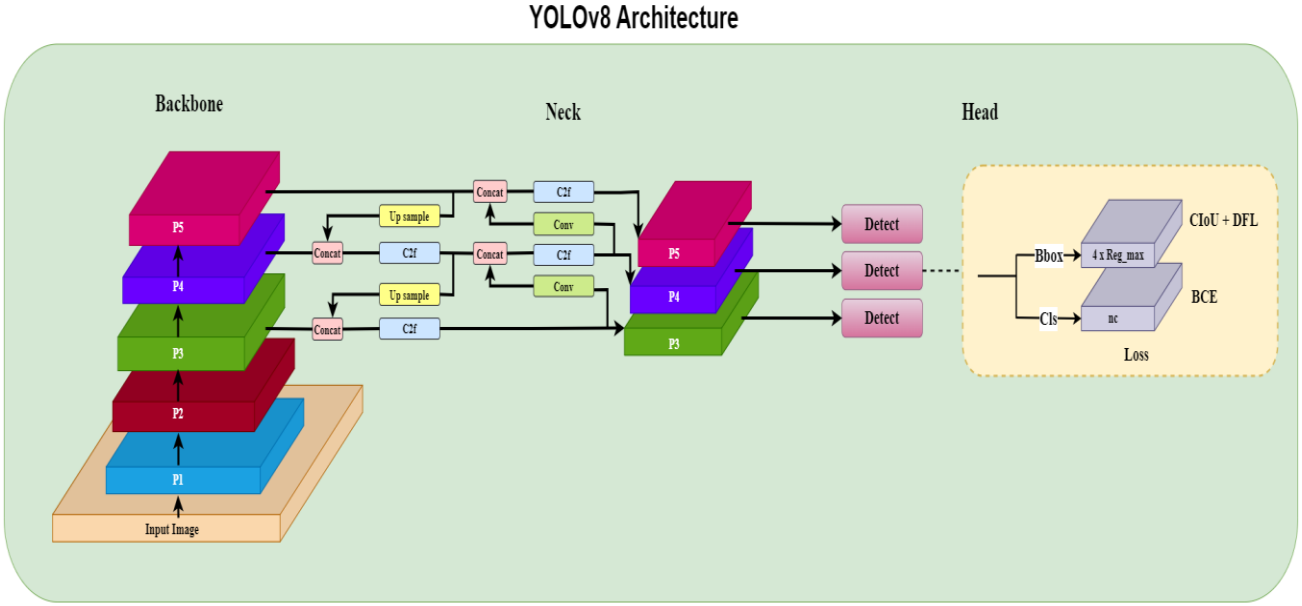


Figure 3.1: Basic Architecture of YOLOv8 Object Detection Model

Within the intricately designed framework of YOLOv8, the architecture of the 'neck' is significantly bolstered by the inclusion of the state-of-the-art C2F (Convolution to Feature) module. This indispensable element is tasked with the fusion of outputs emerging from the convolutional layers, crafting a detailed feature map in the process. This map is a confluence of critical information derived from the input image, laying a solid foundation for advanced processing and insight extraction. Subsequent to its formation, the feature map is meticulously processed through an advanced series of convolutional layers. This critical phase is instrumental in weaving together the high-level features with the relevant contextual information, thereby substantially enriching the model's comprehension and analytical capacity regarding the input.

The C2F module is particularly noteworthy for its proficient capability to harmonize and unify features from the diverse strata of the backbone architecture. This feature unification is instrumental in significantly amplifying YOLOv8's accuracy. It capitalizes on the dynamic interplay and fusion of features from various layers, propelling the model towards achieving unparalleled accuracy. This synergistic fusion of features across different levels of the architecture underpins the model's enhanced object detection proficiency, demonstrating the module's pivotal role in refining and elevating YOLOv8's operational performance.

The head component of YOLOv8 plays a pivotal role in identifying and classifying objects within an image. This segment of YOLOv8's architecture is meticulously designed to include two crucial elements: Anchor-free detection and a Decoupled head, both of which contribute significantly to the model's efficacy and flexibility.

In its approach to object detection, YOLOv8 adopts a center-based methodology, aiming to accurately pinpoint the central position of objects within the image space. This technique simplifies the prediction process by focusing on the most central aspect of an object's presence. Complementing this, the head architecture introduces a Decoupled head mechanism. This innovative design splits the head into two distinct branches: the classification branch and the regression branch. The classification branch is tasked with determining the category to which each detected object belongs, while the regression branch is dedicated to calculating the precise coordinates of the bounding boxes that encapsulate each object.

This bifurcation of responsibilities within the head architecture not only enhances the precision of YOLOv8 but also endows it with greater adaptability, allowing it to excel in a wide array of object detection scenarios. Compared to its predecessors, this enhanced structure offers increased flexibility, making YOLOv8 a more capable and versatile tool for various object detection tasks.

A key difference between the YOLO models lies in their approach to object detection. YOLOv5 utilizes an anchor-based detection system. This method relies on a

collection of predetermined anchor boxes to forecast the dimensions and positions of bounding boxes around detected objects. Conversely, YOLOv8 introduces novel split-head architecture, moving away from the anchor-based method towards an anchor-free detection mechanism. This innovative approach in YOLOv8 allows for the direct prediction of bounding boxes around objects without the need for predefined anchors, offering a more adaptable and dynamic detection process.

This distinction underscores YOLOv8's enhanced capabilities in real-time object detection. By eschewing the constraints of anchor boxes, YOLOv8 achieves a faster and more precise performance in identifying and locating objects within an image. The shift to an anchor-free model and the implementation of split-head architecture not only simplifies the detection workflow but also significantly improves the model's efficiency and accuracy, making YOLOv8 a superior choice for real-time object detection applications when compared to YOLOv5.

The next few sections will discuss the preparation steps and the working of the proposed smart traffic detection system.

3.1 Dataset Preparation:

The methodology for gathering data involved a unique and dynamic approach, specifically designed to capture the diverse and bustling environment of Kolkata. By strategically mounting a camera on a motorcycle, researchers were able to traverse various locations throughout the city, thereby ensuring a wide-ranging and representative collection of video footage. This innovative method allowed for the recording of real-time, on-the-ground traffic conditions, offering invaluable insights into the urban landscape of Kolkata.

The videos obtained through this method were of exceptional quality, recorded in ultra-high-definition 4K resolution at a fluid 60 frames per second. This was made possible by the utilization of a cutting-edge 108MP Samsung ISOCELL HM2 sensor, renowned for its superior imaging capabilities. Such high-resolution recording not only captured the minute details of the city's traffic but also ensured that the data was of sufficient quality for in-depth analysis.

Following the acquisition of the video footage, a rigorous process was undertaken to extract individual frames from the videos. From the vast array of 5,240 video frames, a curated selection of 2,682 images was made. This selection process was guided by the relevance and clarity of the images, ensuring that only the most suitable frames were chosen for further analysis.

Once the images were extracted, they underwent a detailed annotation process using Roboflow software. This crucial step involved labeling the images according to specific traffic-related categories. The dataset was meticulously organized into five categories, reflecting key traffic signals and movements: RED (stop), YELLOW (caution), GREEN STRAIGHT (proceed forward), GREEN LEFT (turn left), and GREEN

RIGHT (turn right). This classification allowed for a structured and comprehensive dataset, providing a solid foundation for subsequent studies on traffic signal recognition and understanding vehicular movement patterns in Kolkata. Through this elaborate process, a rich and detailed dataset was constructed, capturing the complexity and diversity of Kolkata's traffic system.





Table 3.1: Annotated Data sample of the proposed dataset

3.2 Dataset Training:

The meticulously assembled dataset consists of 2,682 images, each carefully selected and annotated with the precision akin to the standards set by the COCO dataset, yet uniquely modified to ensure seamless integration with the YOLOv8 format. Throughout the annotation process, a comprehensive total of 4,531 bounding boxes were manually crafted to accurately outline the objects within these images, distributed across five distinct classes. This diligent effort resulted in an average of approximately 1.7 bounding boxes per image, showcasing a diverse array of object densities and interactions within the visual data. For a detailed understanding, Figures 3.2 and 3.3 in the accompanying documentation serve as illustrative guides, offering a visual exploration of the dataset, with Figure 3.2 focusing on the distribution of objects across the five classes, and Figure 3.3 highlighting the number of images contributing to each class.

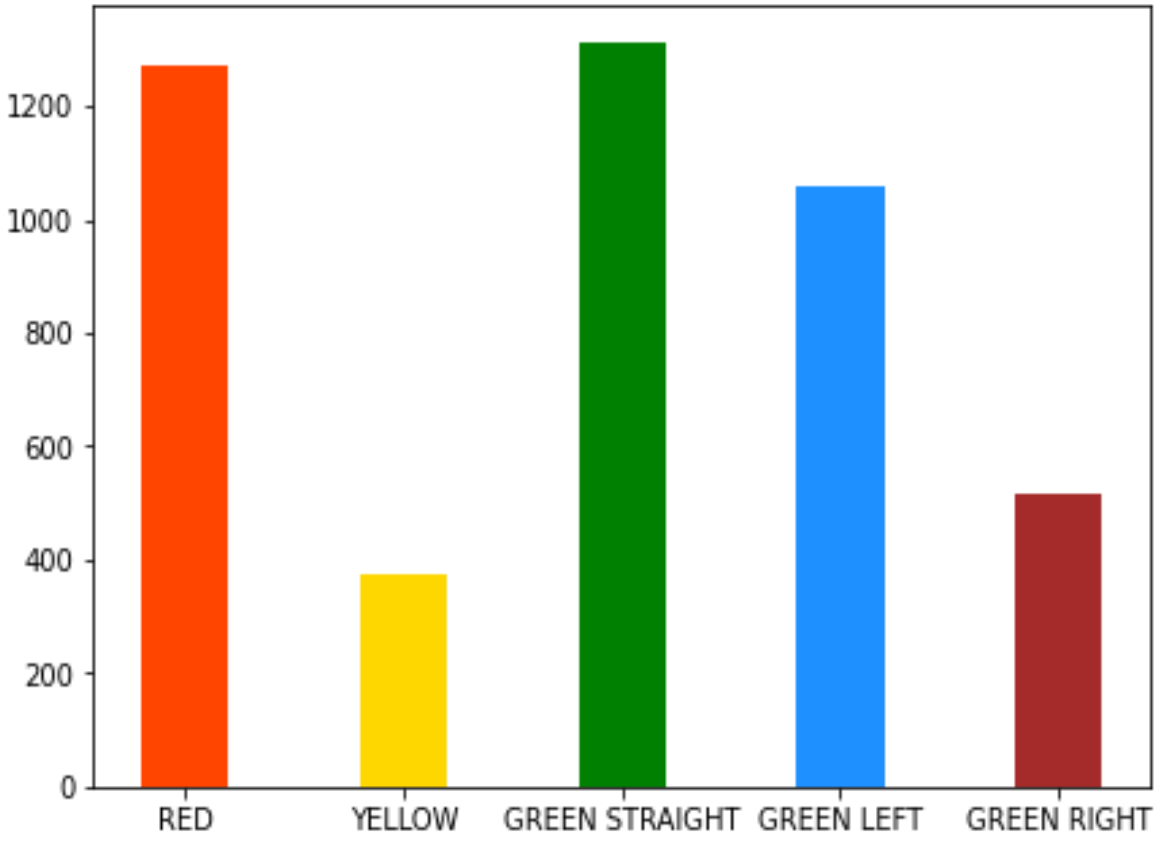


Figure 3.2: Bar chart of number of objects per class

In pursuit of enhancing the dataset's depth and diversity, a suite of advanced image augmentation techniques was strategically employed. Through careful adjustments in saturation, brightness, exposure, and the application of sophisticated blur and noise filters, the dataset was artfully expanded, growing to encompass a total of 5,680 images. This enriched collection of images was then judiciously partitioned, adhering to

an 80-20 ratio, meticulously organizing the images into training and testing sets to construct a robust framework for comprehensive model evaluation.

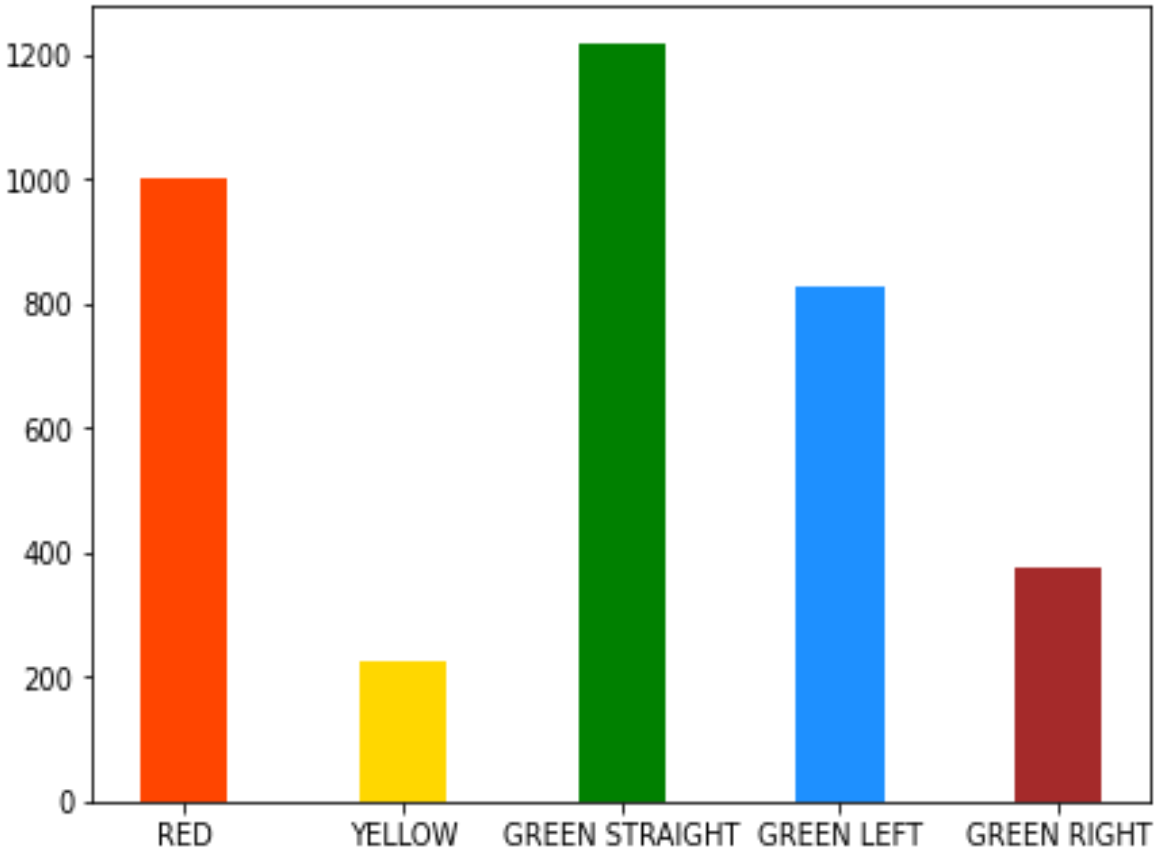


Figure 3.3: Bar chart of number of images having a class

This enriched and carefully annotated dataset embarked on a journey of intensive training under the guidance of YOLOv8s, a specialized iteration of YOLOv8 outfitted with pre-trained weights designed to turbo-charge the learning process. The images were uniformly adjusted to a resolution of 640 x 640 pixels, optimizing them for processing efficacy and consistency. Over the course of 150 epochs, the training process was finely tuned to walk the tightrope between over-fitting—where the model might over-learn from the training set to the point of compromising its ability to generalize to new data—and under-fitting, where the model does not learn sufficiently. This carefully calibrated training regimen is engineered to refine the model's ability to not just recognize but accurately classify objects within images, achieving a harmonious balance that pushes the boundaries of YOLOv8's object detection prowess.

3.3 Environment Setting:

The training of the model was meticulously executed within a virtualized setting provided by Kaggle, harnessing the robust computational capabilities of a P100 GPU. This strategic choice allowed for a highly efficient and productive training period, optimizing the use of computational resources to foster a conducive environment for the model's learning and evolution.

Following the completion of the model's training, an in-depth evaluation of its operational capabilities and overall performance was undertaken. This critical assessment was carried out on a technologically advanced hardware configuration, which included 16GB of RAM and was powered by an Intel Core i7 9th Generation processor. This evaluation phase was pivotal, offering insightful observations regarding the model's adaptability and effectiveness in real-world scenarios.

Moreover, the system underwent a specialized testing process to examine its proficiency in processing and analyzing live visual data. For this purpose, a high-definition live feed camera was employed, capable of delivering crisp and seamless video at a resolution of 1080p and a frame rate of 30 FPS. This setup was instrumental in presenting a dynamic and detailed visual input for the model, thereby testing its ability to accurately interpret and respond to live scenarios.

This elaborate and comprehensive approach, leveraging both advanced virtual training environments and rigorous real-world testing setups, guaranteed that the model was not merely trained in an isolated digital space but was thoroughly vetted against demanding performance and operational standards. This ensured its readiness and reliability for practical applications, marking a significant step forward in assessing the model's preparedness for deployment in real-world settings.

3.4 Signal Detection:

At the heart of the system's operation lies the strategic acquisition of live visual data, seamlessly captured through a sophisticated camera interface. This initiates a dynamic real-time detection process, where the system vigilantly scans its visual feed for the presence of traffic lights. Once a traffic light is discerned within the expansive view of the camera, the system promptly activates a tailored notification mechanism. This mechanism is ingeniously designed to issue alerts to the driver, serving as a pivotal source of vital information. These timely alerts play a significant role in augmenting the driver's situational awareness, thereby enriching their decision-making process in accordance with the prevailing traffic signal conditions.

Engineered with meticulous precision, the system boasts the capability to accurately recognize a wide spectrum of traffic signals that punctuate the vehicular landscape of Indian roads. This array encompasses the quintessential RED (indicating a command to stop), YELLOW (signaling caution, urging drivers to prepare to stop or proceed with heightened vigilance), GREEN STRAIGHT (authorizing the vehicle to continue forward), GREEN LEFT (directing a left turn), and GREEN RIGHT (indicating a right turn). The inclusion of such a diverse range of signal recognition ensures that the system is thoroughly equipped to decipher and act upon the myriad of directives issued by traffic lights, thereby facilitating smoother and more informed navigation through the complex web of intersections and thoroughfares.

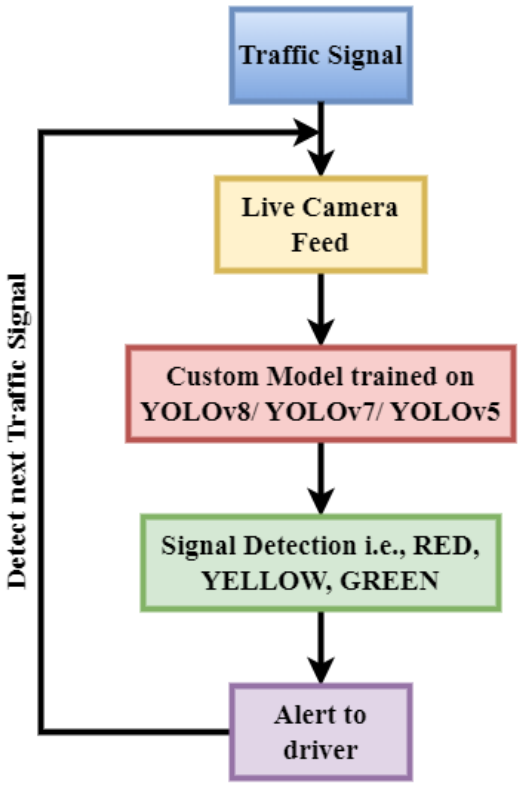


Figure 3.4: An illustration of the flowchart of proposed method

Delving into the operational essence of this system, Figure 3.4 serves as a visual conduit, offering an elaborate depiction of the signal detection process. This figure meticulously outlines the sequential steps undertaken by the system as it processes the incoming live feed, culminating in the precise identification and interpretation of various traffic signals. This graphical exposition not only highlights the system's advanced technological framework but also underscores its invaluable contribution to enhancing road safety, illustrating the pivotal role it plays in fostering safer and more informed driving behaviors.

3.5 Database Synchronization:

The proposed system introduces a cutting-edge addition through the integration of a meticulously designed database, engineered to serve as a vigilant mechanism for tracking and overseeing infractions at traffic signals. Acting as a critical observatory, this database diligently archives instances where traffic signals are not adhered to, thereby functioning as an invaluable asset for the corresponding enforcement agencies. This enables these authorities to efficiently identify and take corrective actions against individuals who flout traffic rules.

The architecture of this database is carefully constructed to encapsulate a comprehensive array of information that is crucial for the identification and prosecution

of traffic violators. It includes exhaustive details about the vehicle involved in the infraction, the identity of the vehicle's owner, essential documents such as the driving license, the vehicle's registration number, and the exact timing of the traffic signal violation. This wealth of data is amalgamated into a cohesive entity within the database, significantly simplifying the task of monitoring and managing instances of traffic rule violations.

To adeptly organize and sift through the collected data, the database is organized into distinct tables, each serving a specific purpose: Car_details, Owner_details, RTO (Regional Transport Office), Fine_details, Time_details, and Defaulters. This organizational schema ensures a systematic categorization of each facet of a traffic violation, from the identity of the violator and the vehicle involved to the particulars of the infraction and the penalties imposed. This level of detail provides a clear and accessible overview of each incident, aiding in thorough analysis and enforcement actions.

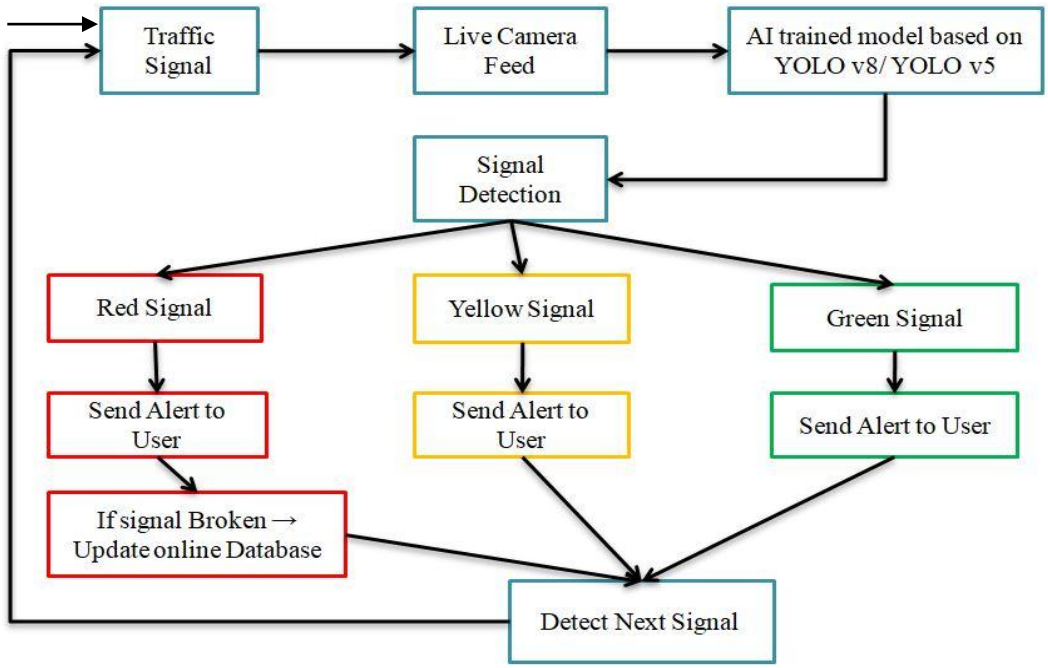


Figure 3.5: Flowchart of syncing the database to the proposed system

Upon detecting a traffic signal violation, the system is programmed to automatically execute a predetermined query within its coding framework. This action triggers an instantaneous update to the Defaulters table, populating it with all the relevant information concerning the violation. Through this seamless automation, the database evolves into an essential instrument in the enforcement of traffic laws, offering a structured and effective methodology for addressing and reducing traffic signal violations, thereby contributing to the overall enhancement of road safety measures.

CHAPTER 4: RESULTS AND OBSERVATIONS

The conducted experiments utilizing the suggested approach were carried out in Python 3.8.3 with the support of the OpenCV 4.7.0 library. This innovative method underwent rigorous testing through its application on YOLOv8, as well as on preceding versions of YOLO objects detection frameworks such as YOLOv5 and YOLOv7. Prior to delving into the outcomes of these experiments, a comprehensive explanation is provided regarding the diverse metrics employed for evaluation.

4.1 Metrics

4.1.1 Metrics Precision:

Precision emerges as a pivotal metric, illuminating the model's precision in predicting positive outcomes accurately. The equation to determine precision is expressed as:

$$Precision = \frac{True\ Positives}{True\ Positives + \ False\ Positives}$$

Within this framework, 'True Positives' denote the instances wherein the model precisely predicts a positive outcome, reflecting an accurate identification. On the opposite end, 'False Positives' are cases where the model erroneously classifies an outcome as positive. This distinction is fundamental for gauging the model's capability in accurately differentiating positive from non-positive occurrences, offering a deep dive into its efficacy and precision in prediction.

4.1.2 Metrics Recall:

Recall serves as an essential metric for evaluating the efficacy of a classification model, particularly in its capability to accurately identify true positive instances. It is determined by the formula:

$$Recall = \frac{True\ Positives}{True\ Positives + \ False\ Negatives}$$

In this context, 'False Negatives' represent the instances where the model incorrectly labels positive cases as negative. This metric is indispensable for assessing the model's sensitivity in recognizing all pertinent positive cases, ensuring that it does not mistakenly neglect them. Through measuring recall, we gain valuable insights into the model's ability to capture and correctly classify all actual positive instances, emphasizing its effectiveness in reducing the occurrence of overlooked true positives.

4.1.3 F1 Score:

The F1 score stands as a holistic indicator of a model’s performance, becoming particularly crucial in evaluating models tasked with navigating imbalanced datasets. A

model boasting a high F1 score is indicative of its proficiency in accurately identifying positive cases, alongside its adeptness at keeping false positives to a minimum. On the flip side, a low F1 score points to a model's struggles with effective classification, either due to inaccuracies in identifying true positives or an inclination towards over-predicting positives. The F1 score is derived through a formula that harmoniously blends precision and recall:

$$F1\ score = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)}$$

This metric effectively encapsulates the balance a model strikes between its precision, the measure of accuracy when predicting positives, and its recall, the ability to comprehensively identify all actual positives. Thus, the F1 score provides a singular, balanced metric reflecting the model's dual strengths in both precise prediction and extensive detection of positive instances, highlighting its overall efficacy in classification amidst challenging conditions.

4.1.4 Mean Average Precision [mAP@ (0.50:0.95)]:

The metric mAP@[0.50:0.95] represents the mean average precision calculated over a diverse array of Intersection over Union (IoU) thresholds, ranging from 0.5 to 0.95 in increments of 0.05. This metric is highly valued for its thorough approach to evaluation, providing a more nuanced and comprehensive analysis of model performance compared to the simpler mAP@0.50. By evaluating the model across a wide spectrum of IoU thresholds, it offers a rich, detailed picture of the model's capabilities.

This broad selection of IoU thresholds facilitates a detailed exploration of the model's proficiency in predicting the boundaries of objects within images. Higher IoU values require greater accuracy in the alignment between the model's predicted object locations and the actual ground truth, ensuring a detailed scrutiny of the model's precision in object delineation. As such, mAP@[0.50:0.95] emerges as an indispensable metric for gauging the model's performance with a granular level of detail, providing insights into its ability to maintain accuracy across a range of precision demands. This enables a comprehensive view of the model's efficacy in object detection tasks, underscoring its reliability and precision in varying conditions.

4.2 Training Results of YOLOv5

After training the dataset using the YOLOv5s model for 150 epochs, the following results were achieved. The results include graphical curves of Precision, Recall, Precision-Recall and F1 score, confusion matrix, and the training results.

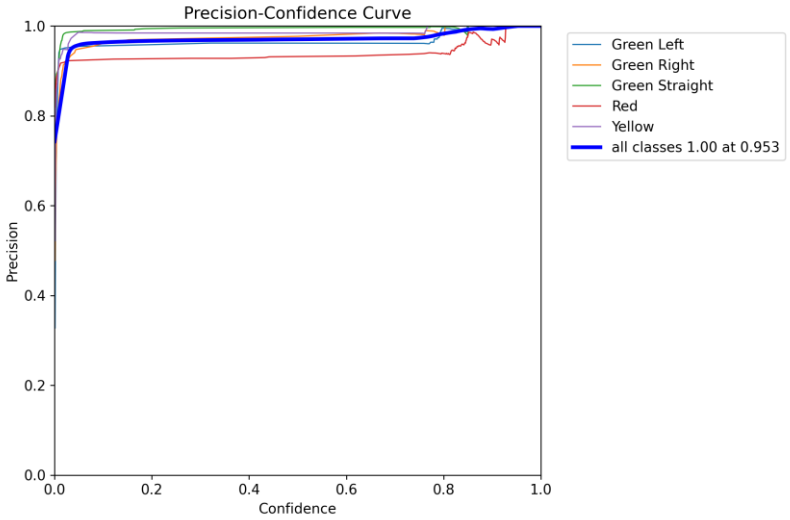


Figure 4.1: Precision curve of YOLOv5s

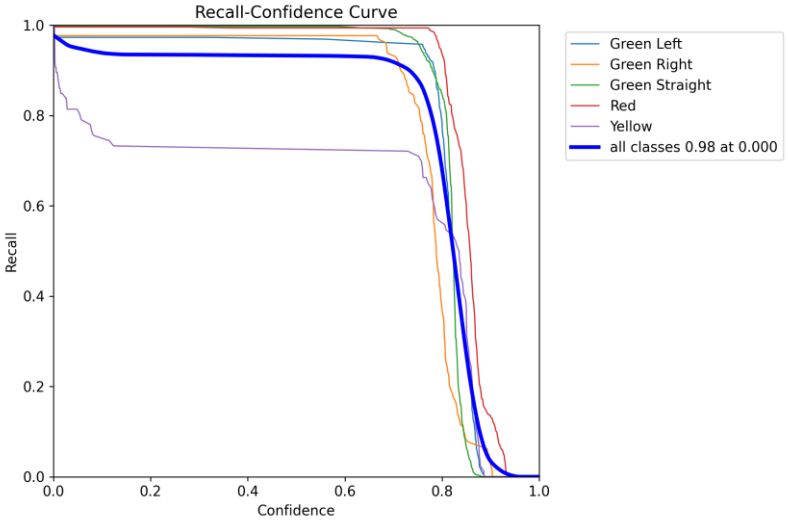


Figure 4.2: Recall curve of YOLOv5s

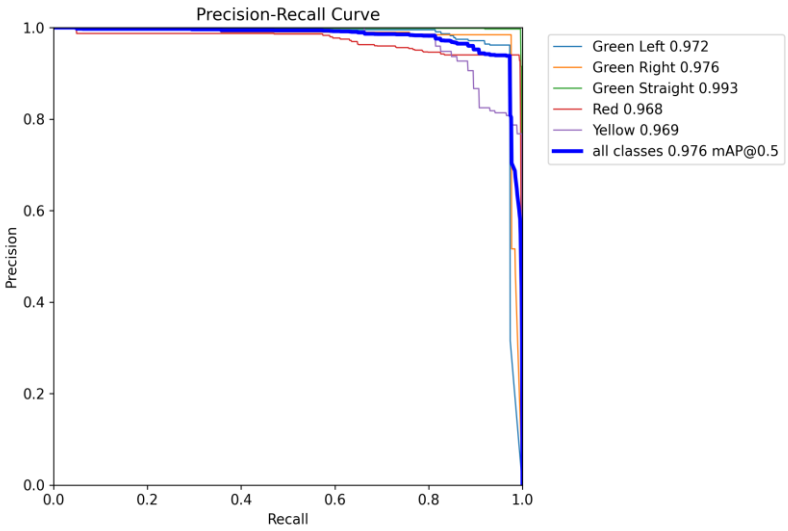


Figure 4.3: Precision-Recall curve of YOLOv5s

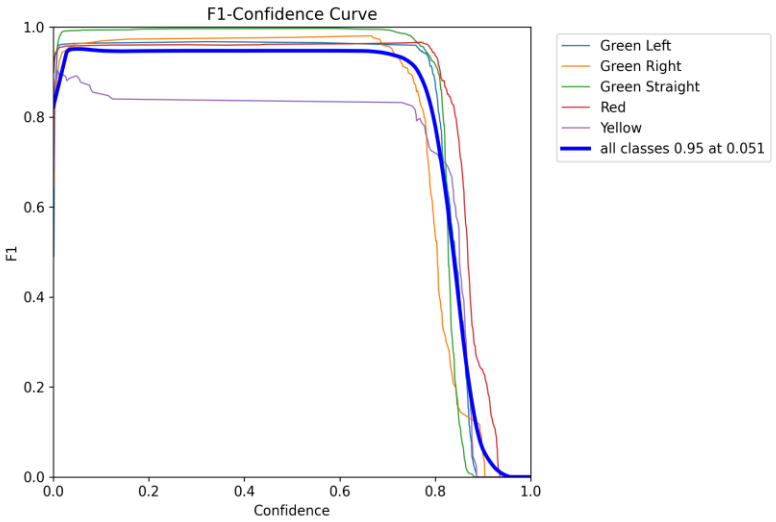


Figure 4.4: F1 curve of YOLOv5s

Figure 4.1 presents the Precision-Confidence Curve, which delineates the precision of the model's classifications across different classes. Notably, the Red class exhibits slightly lower precision in comparison to other classes, whereas the Green Straight class achieves the highest precision. This indicates a variance in the model's ability to accurately identify each class under varying levels of confidence. In Figure 4.2, the Recall-Confidence Curve is illustrated, providing insights into the model's sensitivity or recall for different classes. The Red class demonstrates superior recall capabilities, suggesting effective identification of positive class instances. In contrast, the Yellow class shows the lowest recall, indicating potential challenges in the model's sensitivity towards detecting instances of this particular class. The Precision-Recall Curve displayed in Figure 4.3 elucidates the trade-off between precision and recall at different probability thresholds. This curve is particularly useful in understanding how the model performs in scenarios where class imbalance is pronounced. It helps in assessing whether the model maintains a balance between avoiding false positives and successfully identifying true positives. Class Green Straight has the best precision-recall value, and class Yellow has the least of it. Figure 4.4 introduces the F1 Confidence Curve, which encapsulates the overall performance of the model across different classes. The Green Straight class outperforms others with the highest F1 Score, denoting an optimal balance between precision and recall. Conversely, the Yellow class registers the lowest F1 Score, highlighting areas where the model might require further tuning to enhance its predictive accuracy and reliability.

Each of these figures collectively provides a comprehensive understanding of the model's performance, highlighting strengths in certain classes and pinpointing opportunities for improvement in others. Through this analytical approach, stakeholders can better strategize future modifications to the model to achieve higher accuracy and efficiency in class predictions.

Figure 4.5 displays the confusion matrix for the YOLOv5s model, indicating a generally robust performance across most classes. However, an exception is noted in the case of the Yellow class, which exhibits a number of minor false negative errors.

This suggests a slight propensity for the model to incorrectly identify some instances as belonging to the Yellow class when they do not, underscoring a potential area for improvement in specificity. Figure 4.6 presents the training outcomes for the YOLOv5s model. Detailed examination and analysis of these results can provide insights into the model's learning efficiency and effectiveness throughout the training phase. The implications of these results are crucial for optimizing model performance and achieving higher accuracy in subsequent iterations.

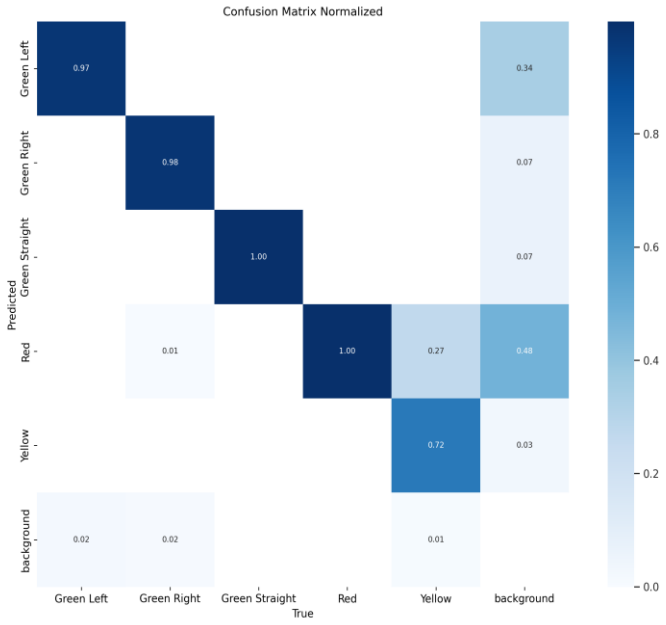


Figure 4.5: Confusion Matrix for YOLOv5s

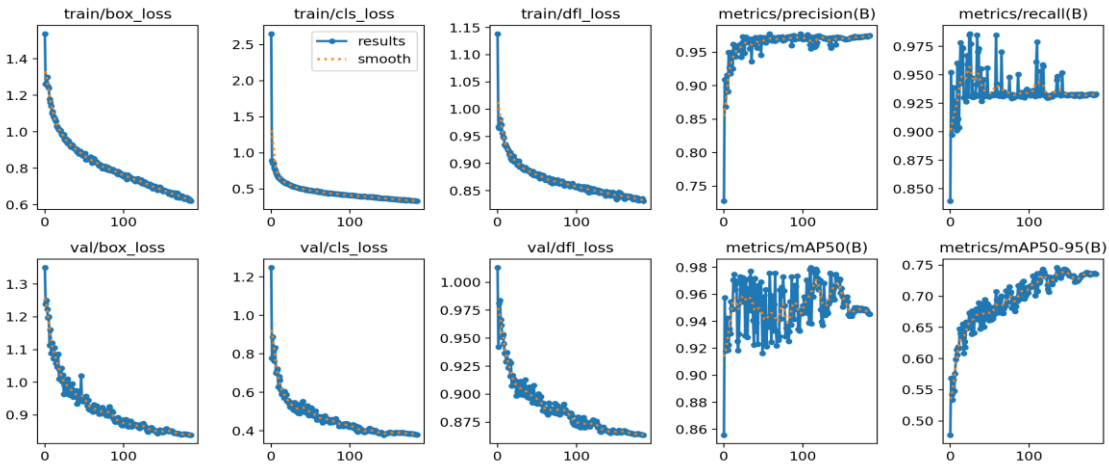


Figure 4.6: Training Results of YOLOv5s

4.3 Training Results of YOLOv7

After training the dataset using the YOLOv7 model for 150 epochs, the following results were achieved. The results include graphical curves of Precision, Recall, Precision-Recall and F1 score, confusion matrix, and the training results.

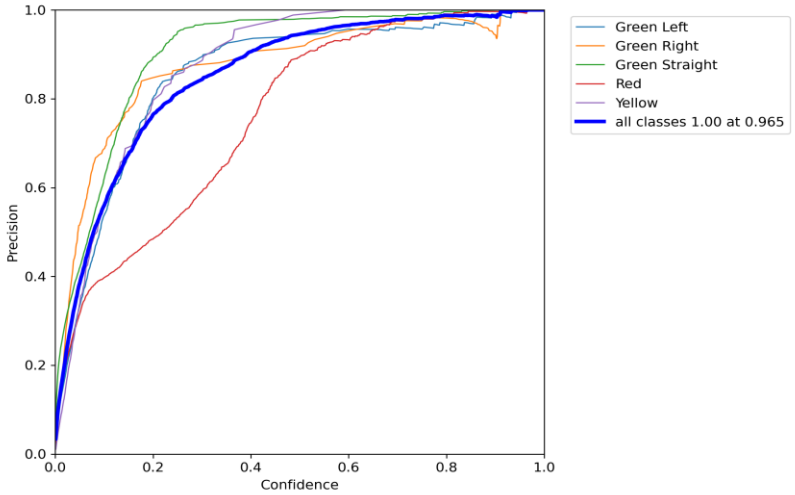


Figure 4.7: Precision curve of YOLOv7

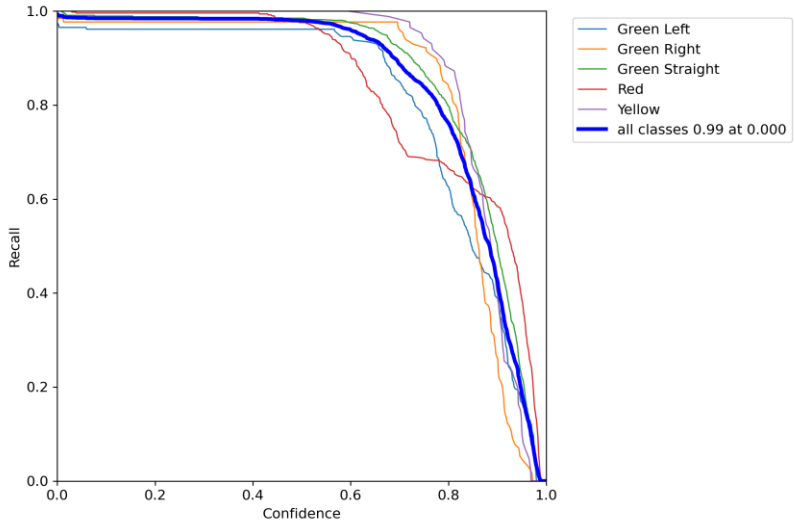


Figure 4.8: Recall curve of YOLOv7

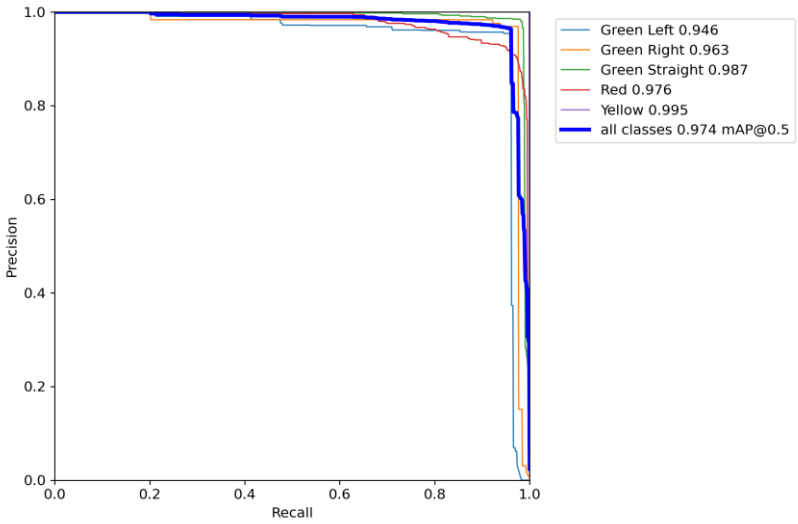


Figure 4.9: Precision-Recall curve of YOLOv7

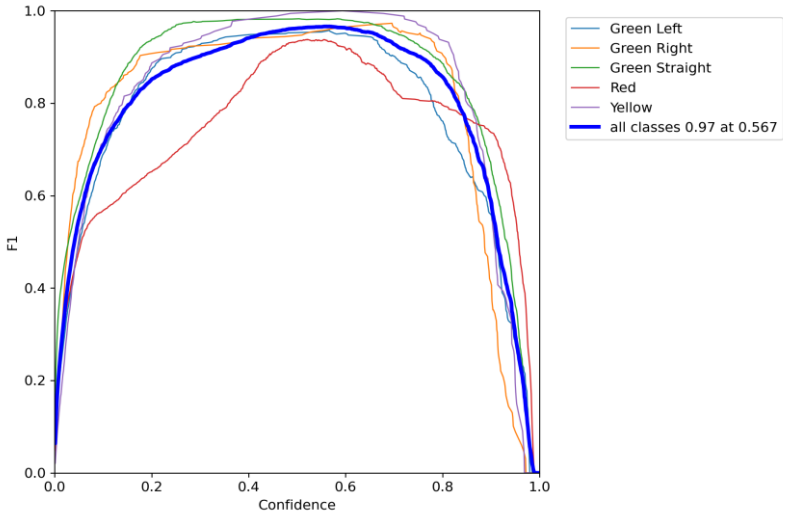


Figure 4.10: F1 curve of YOLOv7

Figure 4.7 presents the Precision-Confidence Curve, illustrating the precision of the model's classification performance across various classes. The Red class is observed to have slightly lower precision relative to other classes, while the Yellow class displays the highest precision. This discrepancy underscores variations in the model's accuracy in classifying different categories under diverse levels of confidence. Figure 4.8 depicts the Recall-Confidence Curve, offering insights into the model's recall or sensitivity for different classes. The Yellow class demonstrates superior recall capabilities, indicating effective detection of positive class instances. Conversely, the Green Right class exhibits the lowest recall, pointing to challenges the model faces in identifying instances of this particular class accurately. The Precision-Recall Curve, shown in Figure 4.9, highlights the trade-off between precision and recall at various probability thresholds. This analysis is particularly valuable for understanding model performance in scenarios characterized by class imbalance. It assists in evaluating whether the model effectively avoids false positives while successfully identifying true positives. Notably, the Yellow class shows the best precision-recall value, whereas the Green Left class records the least. Figure 4.10 introduces the F1 Confidence Curve, which encapsulates the overall performance of the model across different classes. The Green Straight class achieves the highest F1 Score, indicating an optimal balance between precision and recall. In contrast, the Red class exhibits the lowest F1 Score, highlighting areas requiring further tuning to enhance the model's predictive accuracy and reliability.

Together, these figures provide a comprehensive analysis of the model's performance, identifying strengths in certain classes and revealing opportunities for improvement in others. This detailed examination enables stakeholders to strategically plan future modifications to the model, aiming to achieve enhanced accuracy and efficiency in class predictions.

Figure 4.11 presents the confusion matrix for the YOLOv7 model, which demonstrates strong performance across most classes. However, an anomaly is observed with the Red class, where there is a noticeable occurrence of minor false positive errors. This indicates a tendency for the model to mistakenly classify some instances as belonging

to the Red class, highlighting a specific area for enhancement in terms of model specificity. Figure 4.12 details the training results for the YOLOv7 model. A thorough analysis of these results reveals insights into the model's learning efficiency and overall effectiveness during the training phase. Understanding these outcomes is vital for identifying areas where the model's performance can be optimized further, thereby increasing the accuracy in future implementations.

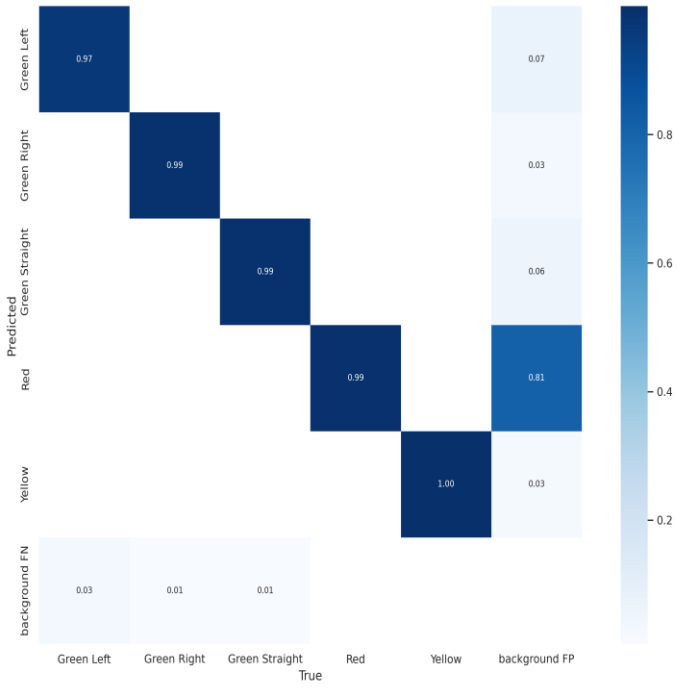


Figure 4.11: Confusion Matrix for YOLOv7

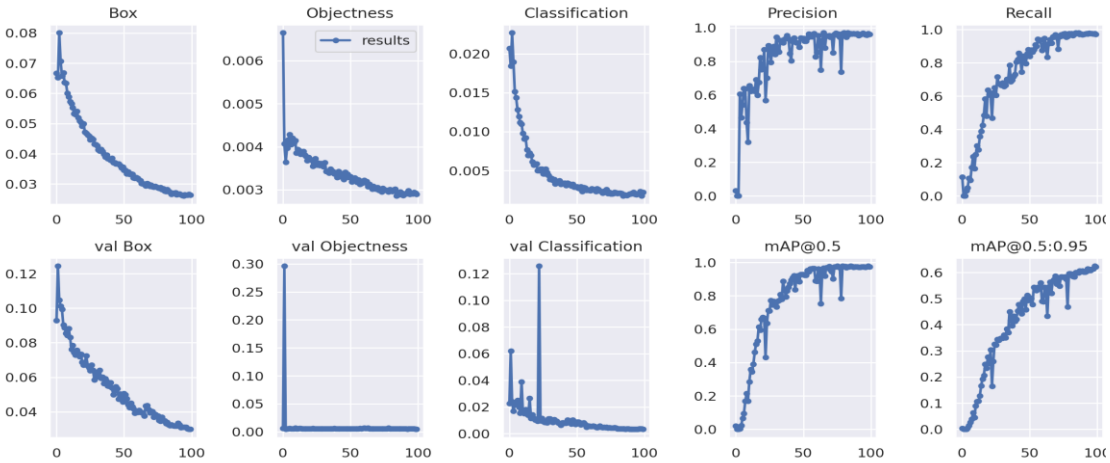


Figure 4.12: Training Results of YOLOv7

4.4 Training Results of YOLOv8

After training the dataset using the YOLOv8s model for 150 epochs, the following results were achieved. The results include graphical curves of Precision, Recall, Precision-Recall and F1 score, confusion matrix, and the training results.

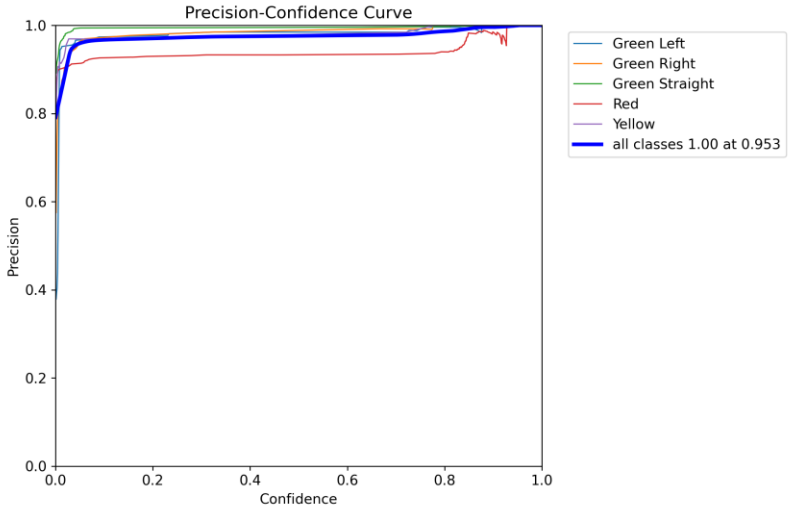


Figure 4.13: Precision curve of YOLOv8s

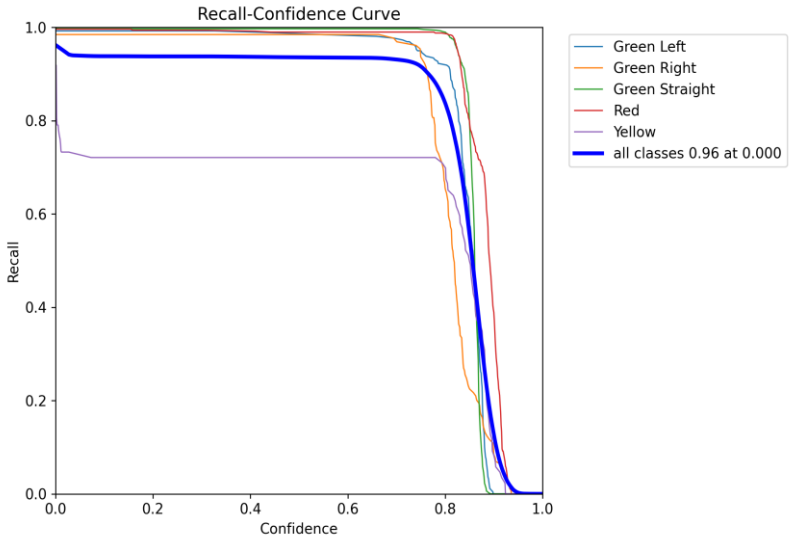


Figure 4.14: Recall curve of YOLOv8s

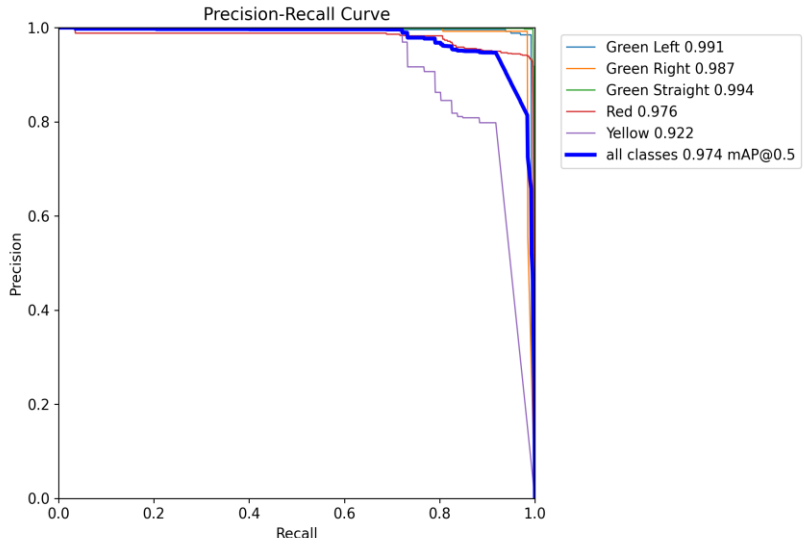


Figure 4.15: Precision-Recall curve of YOLOv8s

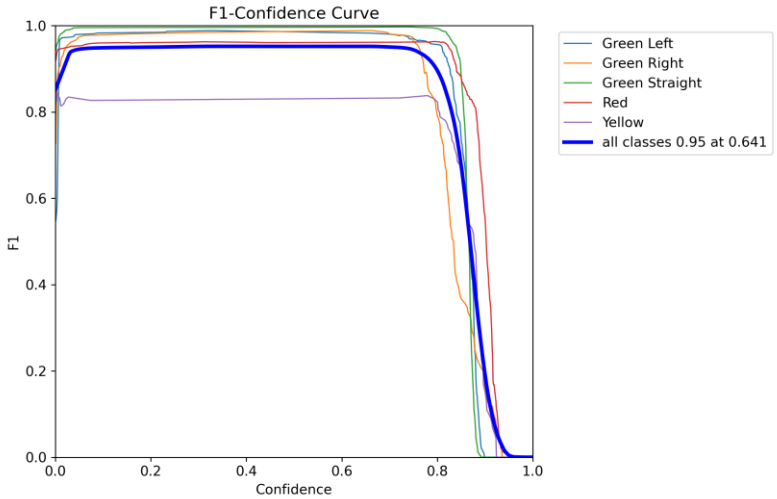


Figure 4.16: F1 curve of YOLOv8s

Figure 4.13 displays the Precision-Confidence Curve, which illustrates the precision of the model's classification performance across various classes. Notably, the Red class exhibits slightly lower precision compared to other classes, whereas the Green Straight class demonstrates the highest precision. This disparity accentuates the variations in the model's accuracy when classifying different categories at varying confidence levels. Figure 4.14 presents the Recall-Confidence Curve, providing valuable insights into the model's recall or sensitivity across different classes. The Red class shows superior recall capabilities, effectively identifying positive class instances. In contrast, the Yellow class has the lowest recall, indicating significant challenges in accurately detecting instances of this class. The Precision-Recall Curve, depicted in Figure 4.15, elucidates the trade-offs between precision and recall at various probability thresholds. This analysis is crucial for understanding the model's performance in situations with class imbalances. It helps assess the model's ability to minimize false positives while effectively identifying true positives. Remarkably, the Green Straight class registers the best precision-recall values, whereas the Yellow class shows the least favorable outcomes. Figure 4.16 introduces the F1 Confidence Curve, summarizing the overall performance of the model across different classes. The Green Straight class achieves the highest F1 Score, reflecting an optimal balance between precision and recall. Conversely, the Red class scores the lowest F1 Score, indicating areas that require further refinement to boost the model's predictive accuracy and reliability.

Collectively, these figures provide an exhaustive evaluation of the model's performance, pinpointing strengths in certain classes while identifying areas for potential improvement in others. This comprehensive analysis equips stakeholders with the necessary insights to strategically plan future enhancements to the model, with the objective of achieving superior accuracy and operational efficiency in class predictions.

Figure 4.17 showcases the confusion matrix for the YOLOv8s model, illustrating robust classification performance across a majority of the classes. However, a notable deviation is observed within the Yellow class, which is marked by a significant number of minor false positive errors. This pattern suggests a propensity for the model to

erroneously identify some instances as belonging to the Yellow class, underscoring an area that necessitates targeted improvements to enhance the model's specificity. Addressing this issue will be crucial for refining the model's accuracy and ensuring that it reliably distinguishes between classes without unwarranted misclassifications. Figure 4.18 provides a detailed exposition of the training outcomes for the YOLOv8s model. A comprehensive analysis of these results yields important insights into the model's learning efficiency and its overall effectiveness throughout the training process. The data derived from this phase is instrumental in pinpointing the strengths and weaknesses of the model's current configuration. By examining these aspects closely, we can identify critical areas where the model's performance can be optimized. Enhancements focused on these areas are expected to lead to significant improvements in the model's predictive accuracy and operational reliability in future deployments. This meticulous examination not only aids in fine-tuning the model but also facilitates a deeper understanding of how different training variables influence the model's behavior. This knowledge is vital for ongoing refinement and achieving the highest standards of performance in real-world applications.

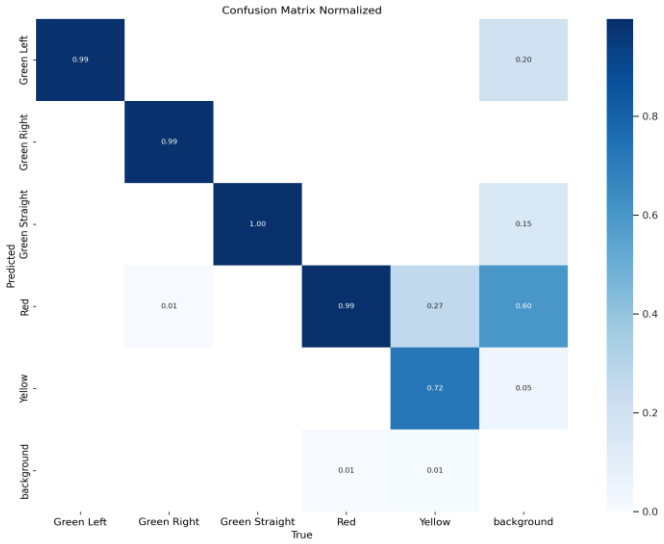


Figure 4.17: Confusion Matrix for YOLOv8s

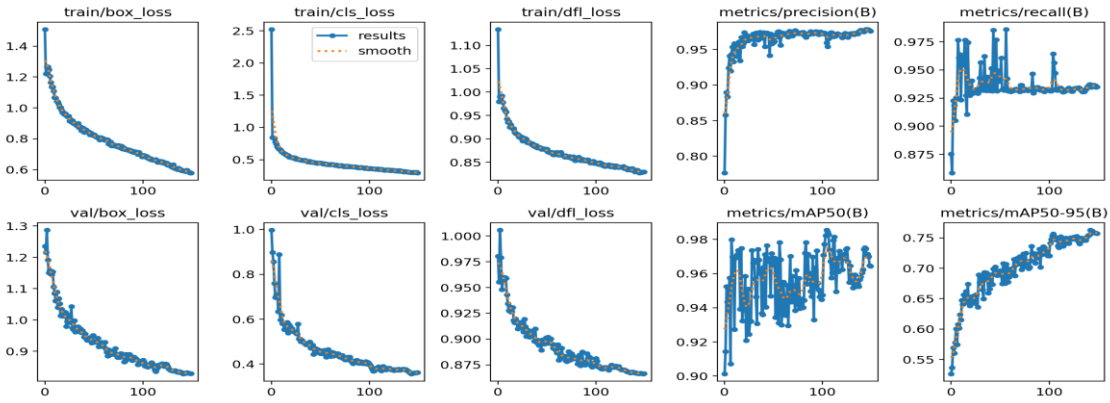


Figure 4.18: Training Results of YOLOv8s

4.5 Performance Comparison between the Models

Table 4.1: Results Obtained While Testing With YOLOv8, YOLOv5 and YOLOv7

Model	Precision	Recall	F-1 score	mAP@[0.50-0.95]
YOLOv7	0.932	0.915	0.964	0.667
YOLOv5	0.967	0.965	0.971	0.736
YOLOv8	0.978	0.986	0.982	0.762

4.5.1 Graphical Representation of the Performance of the Models

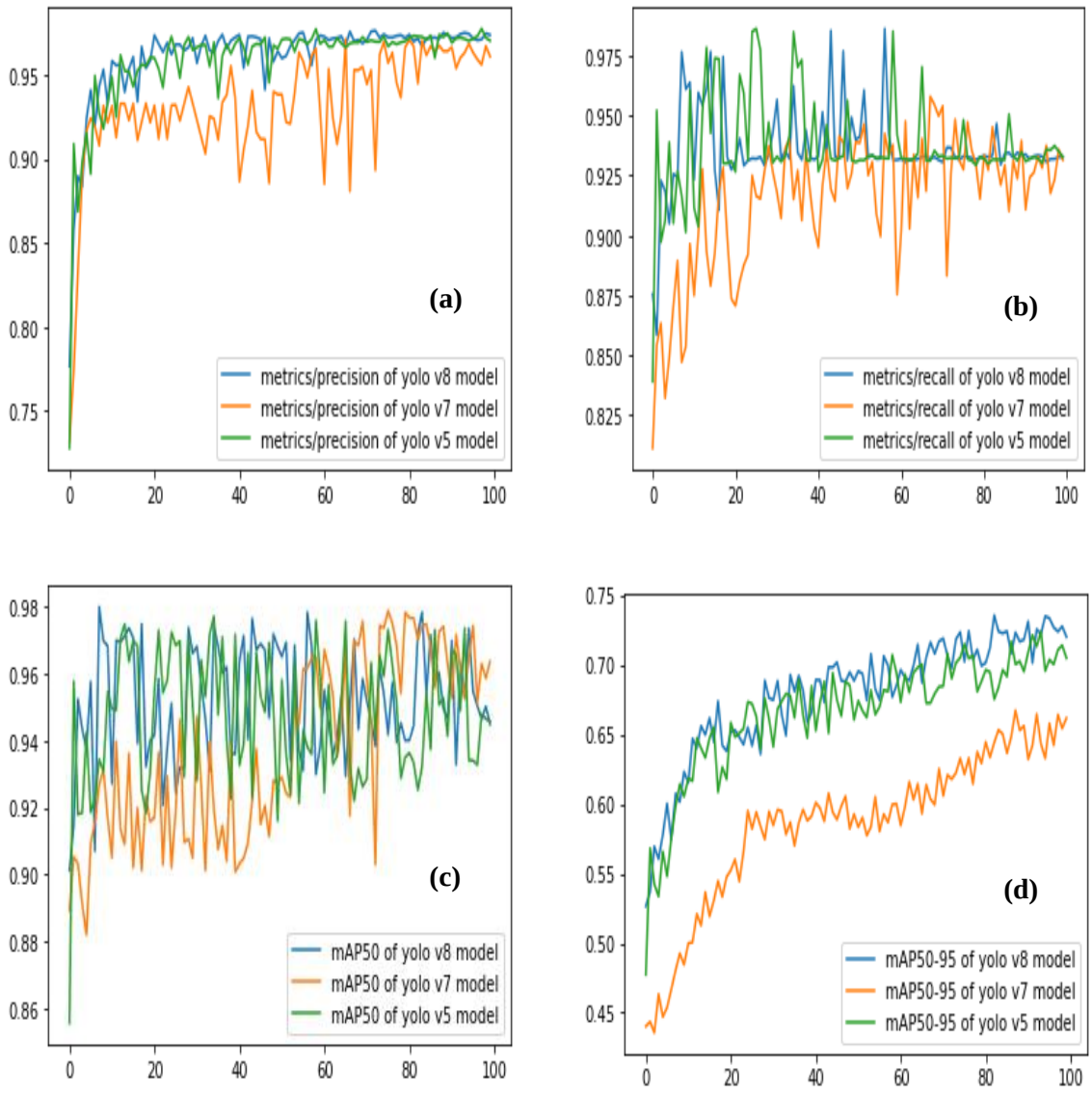


Figure 4.19: (a) Precision (b) Recall (c) mAP@0.50 (d) mAP@[0.50:0.95]

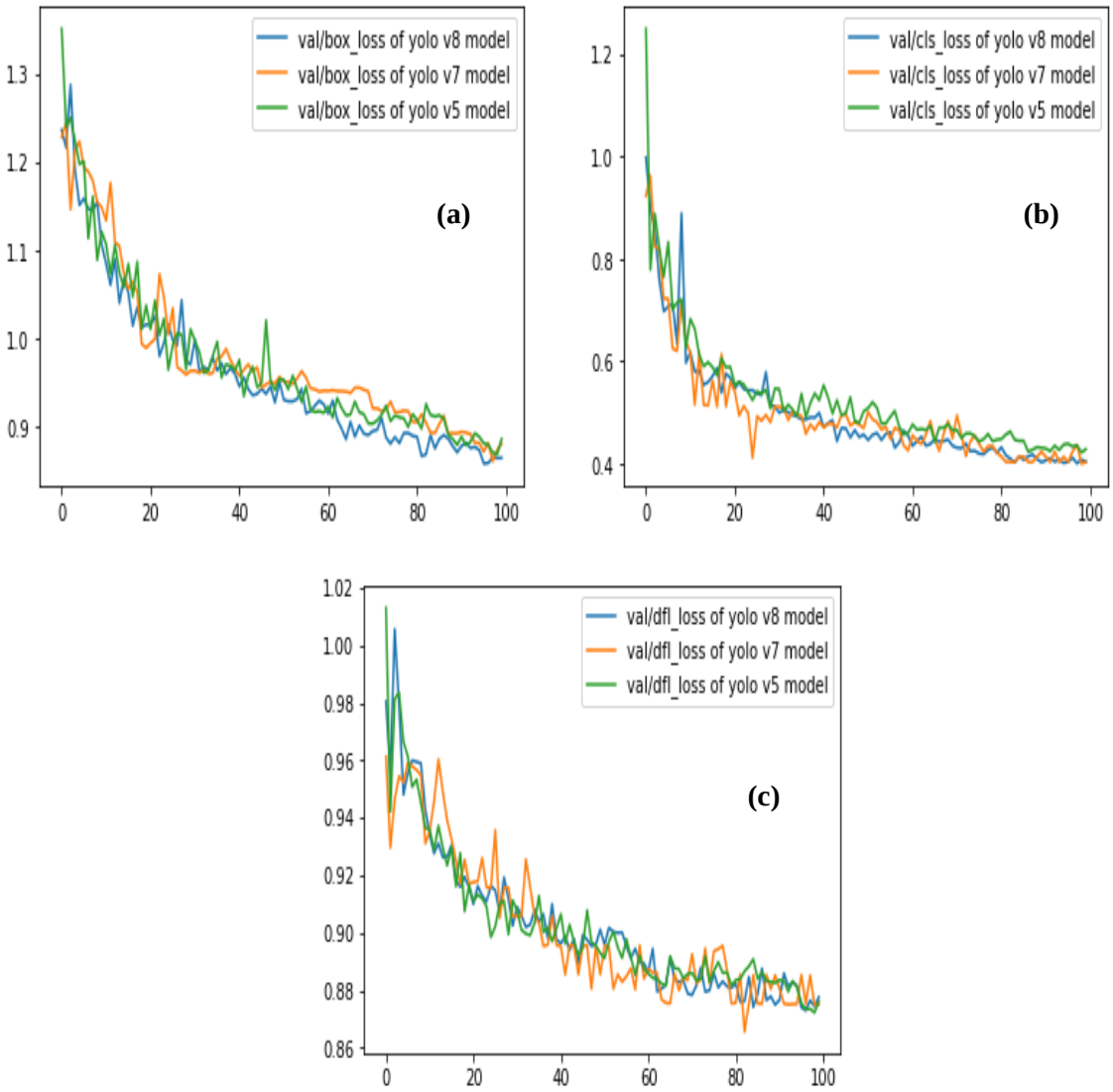


Figure 4.20: (a) Box Loss (b) Class Loss (c) DFL Loss

Table 4.1 presents a detailed comparison of the performance metrics for various YOLO models evaluated on the prepared dataset, specifically focusing on precision, recall, F1 score, and mean average precision (mAP). The analysis reveals that YOLOv8 exhibits superior performance across all metrics when compared to its predecessors, YOLOv5 and YOLOv7. YOLOv8 demonstrates an enhanced precision, outperforming YOLOv5 by 1.1% and YOLOv7 by 4.6%. This improvement in precision indicates a higher accuracy in identifying true positives, which is critical for applications requiring stringent error margins. In terms of recall, YOLOv8 shows a notable advancement, surpassing YOLOv5 by 2.1% and YOLOv7 by 7.1%. This indicates that YOLOv8 is more effective in detecting all relevant instances, thus reducing the likelihood of false negatives — a crucial factor in comprehensive detection tasks. The F1 score, which balances precision and recall, is better in YOLOv8 by 1.1% compared to YOLOv5, and by 1.8% compared to YOLOv7. This suggests a more balanced performance in terms of both precision and recall, affirming the robustness of YOLOv8 in varied operational scenarios. Moreover, YOLOv8 leads with a significant margin in mAP, showing an

improvement of 2.6% over YOLOv5 and 9.5% over YOLOv7. This metric underscores YOLOv8's superior overall prediction accuracy across multiple classes and is indicative of its enhanced capability in handling complex detection environments.

Figure 4.19 corroborates the findings presented in Table 1, displaying the precision, recall, mAP50, and mAP50-95 curves for the trained dataset. This graphical representation effectively illustrates the superior performance metrics of the evaluated models, providing a visual affirmation of the numerical data previously discussed. Figure 4.20 details the various losses incurred during the training process. These include: The Box Loss metric quantifies the model's accuracy in predicting bounding boxes around detected objects, calculated by measuring the discrepancy between the coordinates of predicted bounding boxes and their respective ground truth values. The Class Loss component evaluates the efficacy of the model in classifying detected objects. It measures the divergence between the predicted class probabilities and the actual class labels, providing insight into the model's classification capabilities. The DFL (Distribution-Focal Loss), a variant of the traditional Focal Loss, is utilized to address class imbalance challenges in object detection tasks. This loss focuses on enhancing the model's performance in classifying underrepresented categories by adjusting the focus towards more difficult or less frequent classifications. According to the insights derived from Figure 32, the YOLOv8 model exhibits lower box loss and class loss compared to YOLOv5 and YOLOv7, indicating more precise bounding box predictions and more accurate object classification. Conversely, YOLOv5 demonstrates the lowest DFL loss among the models, suggesting it may handle class imbalance more effectively than YOLOv8 and YOLOv7. Generally, lower loss values indicate more effective training and improved model performance, highlighting the nuanced strengths of each model in various aspects of the detection task.

Overall, the data clearly positions YOLOv8 as the most effective model for traffic signal detection within the dataset used, followed by YOLOv5 and YOLOv7, respectively. The performance enhancements observed with YOLOv8 not only reflect its advanced detection capabilities but also highlight its potential applicability in more demanding and critical real-world scenarios.

4.6 Database Results

Below are the results from the database, formatted to enhance system monitoring capabilities. Figure 4.21 displays the database table containing detailed information on various vehicles, including each car's specific model number. This table serves as a central repository for vehicle specifications, facilitating easy access and management of vehicle-related data. Figure 4.22 illustrates the database entries for car owners, providing comprehensive details including driving license numbers and car registration numbers. This information aids in the identification and verification of car ownership, ensuring accountability and ease of reference for administrative processes. Figure 4.23 presents the details maintained by the Regional Transport Office (RTO), including essential data pertaining to vehicle registrations and owner compliance. This figure

reflects the systematic record-keeping practiced by the RTO to regulate and monitor vehicular and driver-related information. Figure 4.24 outlines the penalties applicable to individuals who violate traffic regulations. This table specifies the fines and sanctions imposed on defaulters, serving as a deterrent against traffic rule violations and enhancing road safety. Figure 4.25 shows a list of individuals who have been caught disobeying traffic signals in real-time. This record is crucial for law enforcement and regulatory purposes, helping to identify repeat offenders and enforce traffic laws more effectively.

Together, these figures and tables constitute a robust database system that supports the efficient management of vehicle and owner information, facilitates law enforcement, and promotes adherence to traffic regulations.

C_id	Model_No	Model_Name
1	C1000	Maruti Wagon R
2	C1010	Maruti Ignis
3	C1020	Maruti Swift Dzire
4	C1030	Tata Tiago
5	C1040	Tata Tigor
6	C1050	Hyundai Santro
7	C1060	Renault KWID
8	C1070	Honda City
9	C1080	Honda Civic
10	C1090	Mahindra Scorpio

Figure 4.21: Car_Details

S_id	OwnerName	Phn_No	D_License	CarLicense
1	Sounav Basak	8240288455	WB99 20190023	WB 12 DE 1433
2	Shubhanu Banerjee	6289762370	WB87 20183477	WB 45 CR 8890
3	Supratim Biswas	9089730431	JH08 23568901	JH 17 RJ 2010
4	Sumit Kumar Mondal	7758500903	MH77 29000345	MH 34 DK 3780
5	Hrishin Roy	8248330923	PB02 88907623	PB 50 LB 8560
6	Gaurav Chauhan	8478930122	TN50 50102992	TN 21 KW 9020
7	Anay Sarkar	7547689383	AP90 20907120	AP 23 IR 1920
8	Abhishek Pratap Singh	9085802823	DL43 30019002	DL 75 VT 6602
9	Ujjwal Verma	6823932934	UP79 86009705	UP 69 DM 2248
10	Gorav Singh	7753939204	MH20 10018685	MH 05 YA 2540

Figure 4.22: Owner_Details

S_id	OwnerName	Model_No	Model_Name	Phn_No	D_License	CarLicense
1	Sounav Basak	C1000	Maruti Wagon R	8240288835	WB99 20190023	WB 12 DE 1433
2	Shubhanu Banerjee	C1010	Maruti Ignis	6289026870	WB87 20183477	WB 45 CR 8890
3	Supratim Biswas	C1020	Maruti Swift Dzire	9007528431	JH08 23568901	JH 17 RJ 2010
4	Sumit Kumar Mondal	C1030	Tata Tiago	7074950903	MH77 29000345	MH 34 DK 3780
5	Hrishin Roy	C1040	Tata Tigor	8240820913	PB02 88907623	PB 50 LB 8560
6	Gaurav Chauhan	C1050	Hyundai Santro	8456238905	TN50 50102992	TN 21 KW 9020
7	Anay Sarkar	C1060	Renault KWID	7546820083	AP90 20907120	AP 23 IR 1920
8	Abhishek Pratap Singh	C1070	Honda City	9000579843	DL43 30019002	DL 75 VT 6602
9	Ujjwal Verma	C1080	Honda Civic	6894362334	UP79 86009705	UP 69 DM 2248
10	Gorav Singh	C1090	Mahindra Scorpio	7758902241	MH20 10018685	MH 05 YA 2540

Figure 4.23: RTO_Details

S_No	Penalty	Charges
1	No License	5000
2	No Seat Belt	100
3	Signal Disobey	2000
4	Over Speeding	1000
5	Obstructive Parking	500

Figure 4.24: Fine_Details

S_id	OwnerName	Model_No	Model_Name	Phn_No	D_License	CarLicense	Penalty	Charges	Time
4	Sumit Kumar Mondal	C1030	Tata Tiago	7074950903	MH77 29000345	MH 34 DK 3780	Signal Disobey	2000	2023-06-21 17:06:58
5	Hrishin Roy	C1040	Tata Tigor	8240820913	PB02 88907623	PB 50 LB 8560	Signal Disobey	2000	2023-09-23 18:02:20
3	Supratim Biswas	C1020	Maruti Swift Dzire	9007528431	JH08 23568901	JH 17 RJ 2010	Signal Disobey	2000	2023-09-23 18:04:28
2	Shubhanu Banerjee	C1010	Maruti Ignis	6289026870	WB87 20183477	WB 45 CR 8890	Signal Disobey	2000	2023-09-23 18:06:16
2	Shubhanu Banerjee	C1010	Maruti Ignis	6289026870	WB87 20183477	WB 45 CR 8890	Signal Disobey	2000	2023-09-23 18:06:22

Figure 4.25: Defaulters_Details

4.7 Detection Result of the 3 YOLO models

YOLOv5 YOLOv7 YOLOv8

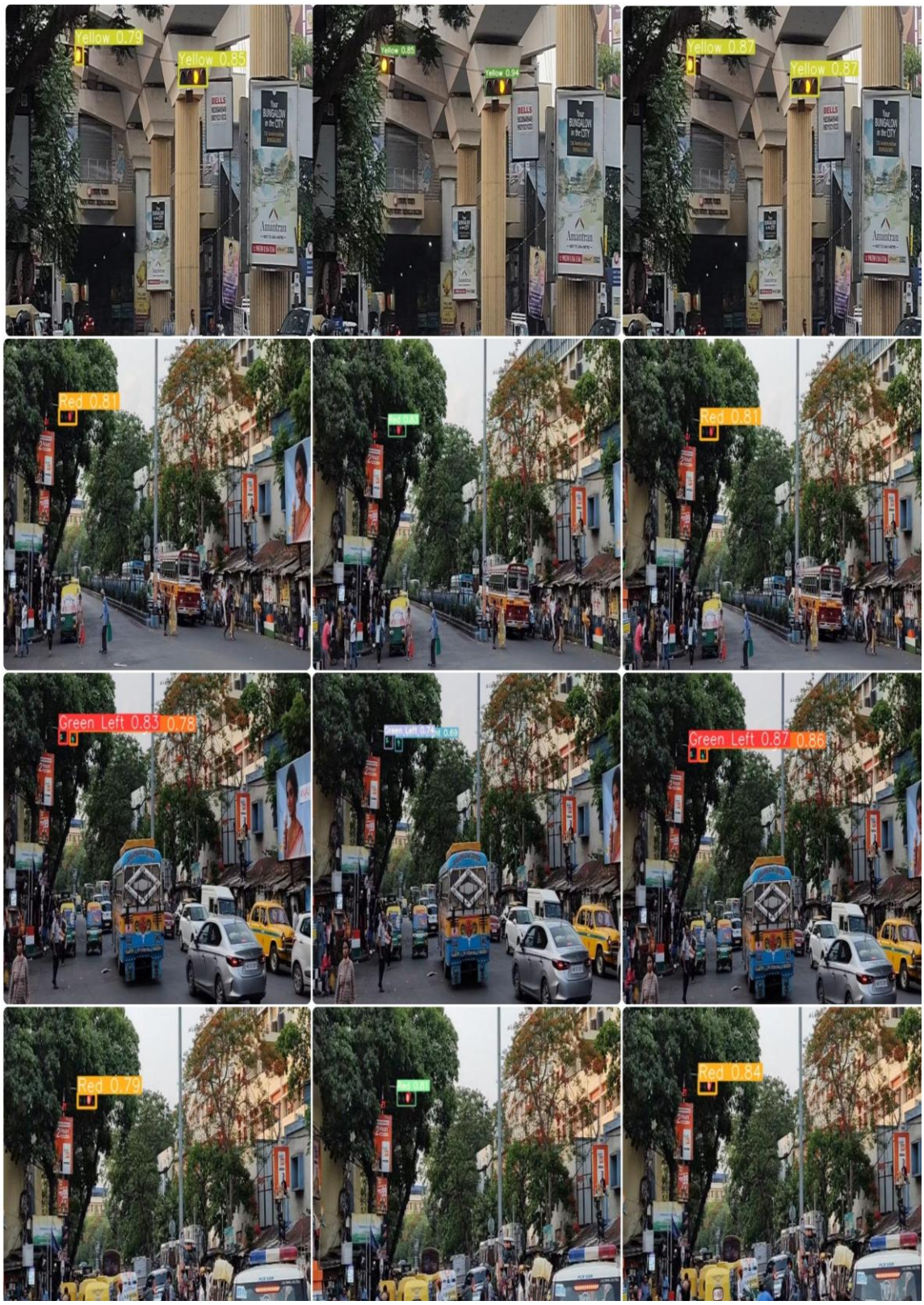


Table 4.2: Comparison of Signal Detection results using YOLOv5, YOLOv7 and YOLOv8

CHAPTER 5: ARCHITECTURAL ANALYSIS OF THE YOLO MODELS

This section concerns the comparative analysis of three YOLO models: YOLOv5, YOLOv7, and YOLOv8.

5.1 YOLOv5

The architecture of YOLOv5, built on CSPDarknet53, exemplifies a sophisticated and efficient design aimed at enhancing object detection performance through several innovative components and strategies. CSPDarknet53, which serves as the backbone, integrates the Cross Stage Partial Network (CSPNet) to optimize network efficiency and gradient flow. CSPNet partitions the feature map into two sections: one undergoes multiple convolutional layers, while the other bypasses these layers. These sections are subsequently merged; ensuring balanced gradient flow and mitigating common issues in deep networks, such as gradient vanishing or explosion. This approach also reduces computational demands, maintaining high performance even on hardware with limited resources.

YOLOv5 further incorporates a focus layer designed to enhance spatial resolution reduction and capture spatial information effectively. This layer divides the input image into patches and stacks these patches along the depth (channel) dimension. For example, a 4x4 image can be split into four 2x2 patches, treated as four channels. This spatial-to-channel conversion enables the network to deepen without losing crucial spatial details, effectively managing computational load while preserving essential spatial information for accurate object detection.

Additionally, YOLOv5 integrates the Spatial Pyramid Pooling (SPP) module to capture contextual information at multiple scales. SPP applies pooling operations at various scales, such as 1x1, 3x3, and 5x5, allowing the network to gather information from different receptive fields. This multi-scale pooling enhances the network's ability to understand the context of objects within the image, improving robustness and accuracy in detecting objects of varying sizes and aspect ratios. The rich contextual information captured by SPP ensures that the network can identify objects even if they are partially obscured or appear in unusual sizes.

The combined strengths of CSPNet, the focus layer, and SPP create a highly efficient and powerful architecture for object detection. CSPNet’s strategy of partitioning and merging feature maps ensures smooth gradient flow and stable, fast training. The focus layer’s spatial-to-channel transformation allows for deeper networks without compromising spatial information, which is crucial for detecting fine details in images. SPP’s multi-scale context pooling enriches the network’s understanding of images, making it versatile in detecting objects of various sizes and contexts.

In the architecture of YOLOv5, the neck segment incorporates both the Feature Pyramid Network (FPN) and the Path Aggregation Network (PAN) to enhance its object detection capabilities. The FPN is designed to combine features from various levels of the backbone, resulting in a comprehensive multi-scale feature map. This enables the network to leverage rich, hierarchical feature representations from different stages of the backbone, which are crucial for detecting objects at multiple scales.

Conversely, the PAN plays a vital role in refining feature propagation and improving object localization precision. PAN achieves this by aggregating features from different levels within the network, enhancing information flow and ensuring effective combination of higher-level semantic information with lower-level detailed features. This allows the network to utilize both high-level contextual information and fine-grained details, improving feature utilization.

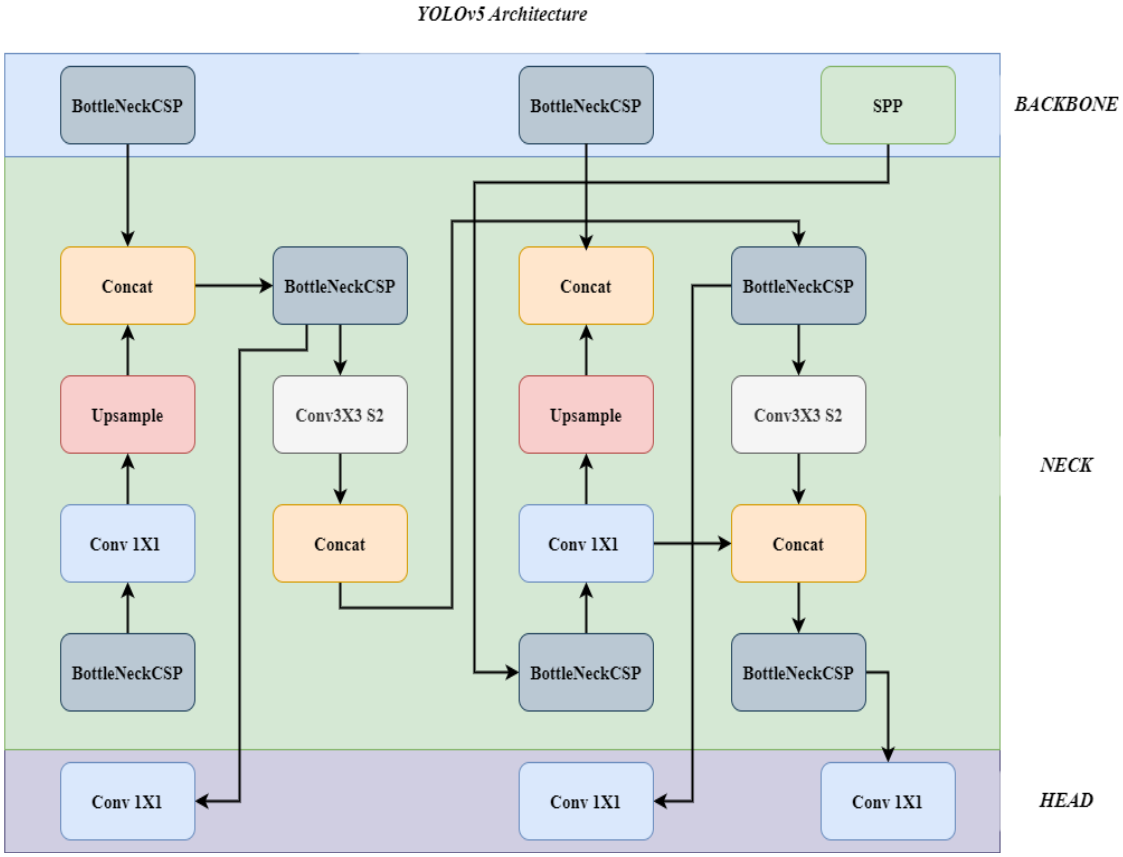


Figure 5.1: Basic Architecture of YOLOv5

The integration of FPN and PAN in the neck segment significantly enhances the model's ability to accurately identify and localize objects at various scales within input images. The FPN's multi-scale feature aggregation provides the network with a robust understanding of the input at different resolutions, which is essential for detecting objects of varying sizes. Meanwhile, the PAN's refinement and propagation of features facilitate precise localization, enhancing the network's ability to distinguish between closely situated objects and improving overall detection accuracy.

The combination of FPN and PAN within the neck of YOLOv5 represents a sophisticated strategy to address the challenges of multi-scale object detection. By merging multi-scale feature aggregation with refined feature propagation, YOLOv5 achieves high levels of accuracy and robustness in object detection tasks. This architectural choice enhances the model's performance in real-world applications, where objects frequently appear at different scales and require precise localization for accurate detection and classification.

In the head architecture of YOLOv5, k-means clustering is utilized on the dataset to determine the optimal dimensions of the anchor boxes. This technique is vital for effective object detection, as it ensures the initial anchor box dimensions are well-suited to the specific size distribution of objects within the dataset, thereby enhancing detection accuracy and efficiency. By applying k-means clustering, YOLOv5 can derive anchor boxes that better match the objects' size variations, leading to improved overall performance.

YOLOv5 incorporates three detection layers, each specifically designed to detect objects at different scales within the input data. These layers are essential for multi-scale object detection, enabling the model to manage objects of various sizes. Each detection layer predicts bounding boxes relative to the grid cells on the feature map, facilitating precise localization of objects across different scales and spatial locations.

The three detection layers operate synergistically to cover a wide range of object sizes. The first detection layer targets large objects by using a feature map with lower spatial resolution but richer semantic information. The second layer is optimized for medium-sized objects, providing a balance between spatial resolution and semantic depth. The third layer focuses on small objects, utilizing a high spatial resolution to capture fine details effectively.

By predicting bounding boxes relative to grid cells, these detection layers accurately localize objects within the image. Each grid cell is tasked with predicting a set of bounding boxes, which include coordinates, dimensions, and confidence scores for object presence. This method allows YOLOv5 to detect multiple objects within an image efficiently, even when the objects vary significantly in size.

The integration of k-means clustering for determining anchor box dimensions and the incorporation of multi-scale detection layers illustrate the sophisticated design of YOLOv5's head architecture. K-means clustering ensures the anchor boxes are optimally initialized and tailored to the dataset's specific characteristics, enhancing the model's detection performance. The multi-scale detection layers enable precise localization and identification of objects of varying sizes, contributing to the model's robustness and accuracy.

In summary, YOLOv5's architectural blueprint prioritizes computational efficiency, rendering it ideal for real-time applications and deployment on devices with limited resources. By integrating advanced components, YOLOv5 exhibits versatility, effectively adapting to various tasks and datasets while maintaining consistently high

performance levels. This holistic approach equips YOLOv5 to analyze intricate images containing multiple objects of diverse sizes, thereby enhancing its reliability and precision. The amalgamation of CSPDarknet53 as its foundation, complemented by the focus layer and Spatial Pyramid Pooling (SPP), epitomizes a meticulously engineered equilibrium between computational efficiency, adept training methodologies, and resilient object detection capabilities, positioning YOLOv5 as an indispensable model across a spectrum of real-world applications.

Moreover, the neck segment of YOLOv5, harmoniously integrating the Feature Pyramid Network (FPN) and Path Aggregation Network (PAN), underscores a strategic enhancement in object detection mechanisms. Leveraging FPN's capability to generate exhaustive multi-scale feature maps alongside PAN's finesse in feature refinement for precise localization, YOLOv5 significantly heightens its proficiency in object detection across varied scales. The integration of FPN and PAN within the neck segment signifies a refined strategy tailored to tackle the complexities of multi-scale object detection, solidifying YOLOv5's stature as a formidable asset in real-world applications.

Furthermore, the head architecture of YOLOv5 adopts k-means clustering to determine the optimal dimensions of anchor boxes and incorporates three detection layers adept at handling objects of diverse scales. This meticulous approach ensures precise object localization and effective detection across various spatial dimensions and sizes. The harmonized methodologies of k-means clustering for anchor boxes and the multi-scale detection layers further amplify YOLOv5's efficacy in accurately detecting and localizing objects.

In essence, YOLOv5's comprehensive design, encompassing elements from its CSPDarknet53 backbone to the intricate integration of FPN, PAN, and sophisticated head architecture, underscores its unparalleled performance and resilience in object detection. These meticulously engineered components collectively position YOLOv5 as a potent and efficient model, indispensable across an array of real-world object detection applications.

5.2 YOLOv7

The incorporation of the ELAN (Efficient Layer Aggregation Network) within the backbone architecture of YOLOv7 represents a significant advancement in the field of object detection models. ELAN is engineered to bolster the model's feature aggregation capabilities across diverse scales without introducing additional computational overhead. This is particularly crucial as object detection tasks often demand the detection of objects with varying sizes and aspect ratios within an image, necessitating the extraction of features at multiple scales.

The integration of ELAN blocks within the backbone architecture is instrumental in enhancing this capability. These blocks leverage multiple convolutional layers with differing kernel sizes to effectively capture a wide range of features across various scales. By utilizing varying kernel sizes, the model can extract features ranging from

intricate details to broader, high-level information, enabling it to detect objects of varying sizes and complexities within the image.

Despite the incorporation of ELAN, YOLOv7 retains the use of the CSP (Cross Stage Partial) network. The CSP network serves a dual purpose: it assists in mitigating computational complexity while enhancing the model's learning capacity. This dual functionality ensures that while ELAN improves feature aggregation capabilities, CSP helps maintain manageable computational overhead, thereby preserving the model's efficiency.

The synergistic integration of ELAN for feature aggregation and CSP for computational efficiency culminates in a highly proficient object detection model. Through adeptly extracting and processing features at multiple scales, YOLOv7 showcases superior performance in detecting objects with varying sizes and complexities within images. This contributes significantly to its efficacy in diverse object detection tasks across various domains, including surveillance and autonomous driving.

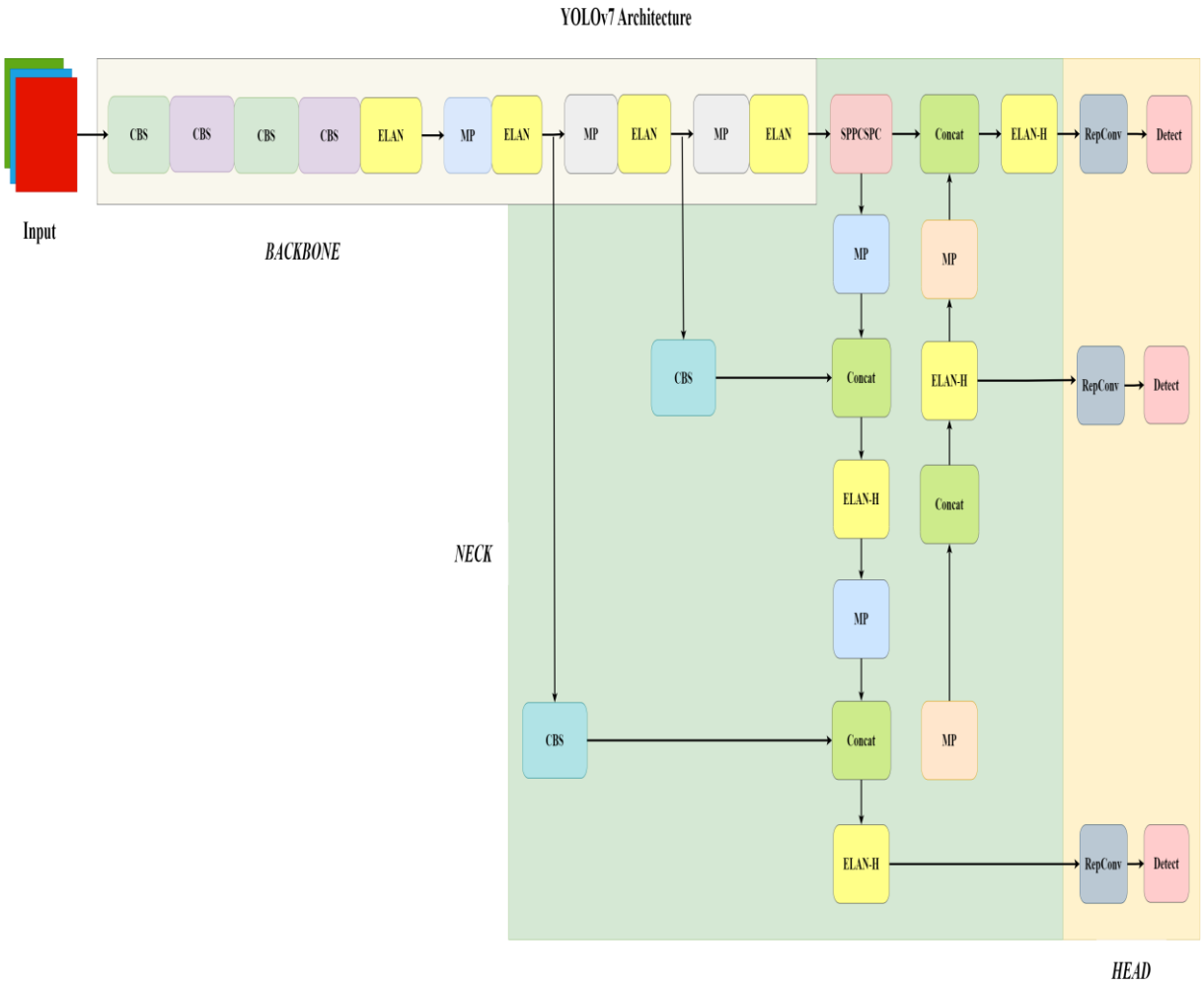


Figure 5.2: Basic Architecture of YOLOv7

The neck architecture of YOLOv7 incorporates several pivotal elements with the aim of optimizing feature processing and facilitating seamless information flow. Among these

components, the Feature Pyramid Network (FPN) assumes a prominent role. FPN is strategically utilized to merge features extracted from disparate layers of the backbone network. This integration streamlines the creation of a comprehensive multi-scale feature map, thereby augmenting the model's proficiency in detecting objects across a spectrum of sizes within the input image.

In tandem with FPN, YOLOv7 employs the Path Aggregation Network (PAN) to further refine information dissemination across various feature maps. PAN plays a critical role in ensuring precise object detection across diverse scales by enabling efficient aggregation and fusion of features from different layers of the network. By enhancing coherence and flow of information among feature maps, PAN significantly enhances the model's accuracy and overall performance in object detection tasks.

Moreover, YOLOv7 introduces the Mix Path Aggregation (MPA) mechanism, which dynamically evaluates model predictions. This adaptive approach enriches the robustness of feature maps by integrating diverse perspectives and insights during the aggregation process. Through iterative refinement of feature representations based on model predictions, MPA further strengthens the model's capability to detect objects accurately across a broad range of scales present in the input data.

The cohesive integration of FPN, PAN, and MPA within the neck architecture of YOLOv7 ensures proficient feature representation and elevates the model's adeptness in detecting objects with precision across varying scales inherent in the input data. This holistic approach underscores YOLOv7's efficacy in object detection tasks and reaffirms its applicability across diverse real-world scenarios demanding robust and accurate detection capabilities.

In YOLOv7, the head architecture closely resembles that of its predecessor, YOLOv5, with a notable enhancement—the integration of the decoupled head architecture. This addition signifies a significant advancement in object detection methodologies, as it segregates the processing of objectness score, classification, and bounding box regression into distinct components.

This architectural refinement plays a pivotal role in enhancing the overall performance of the model. By allocating separate components to handle different aspects of object detection, YOLOv7 is capable of streamlining and optimizing its processing pipeline. Consequently, this leads to a marked improvement in accuracy and reliability across various object detection tasks.

The incorporation of the decoupled head architecture brings substantial benefits to the model's efficacy in real-world applications. Its ability to efficiently manage objectness score, classification, and bounding box regression tasks separately significantly enhances the model's capacity to accurately identify and localize objects within images. Consequently, this strengthens the model's versatility and reliability across a diverse array of practical scenarios, consolidating its position as an effective and dependable solution for real-world applications.

5.3 YOLOv8

The YOLOv8 architecture incorporates several fundamental components aimed at improving the effectiveness and efficiency of object detection. Among these components is the integration of CSPNet (Cross Stage Partial Network), a sophisticated mechanism that establishes connections between feature maps at various depths within the network. This integration facilitates enhanced information exchange and reduces computational overhead by enabling earlier layers to access information from subsequent layers. Consequently, gradient flow during training is improved, leading to more efficient learning processes.

Moreover, YOLOv8 utilizes residual connections, a technique initially popularized by the ResNet architecture, to overcome the challenge of vanishing gradients in deep neural networks. By introducing skip connections that bypass certain layers, residual connections facilitate the seamless flow of gradients during training. This simplifies the optimization of deep networks, resulting in expedited convergence and enhanced overall performance.

Additionally, YOLOv8's architecture incorporates concepts inspired by MobileNetV2, such as inverted residuals and linear bottlenecks. Inverted residuals entail the utilization of depthwise separable convolutions followed by expansion and projection layers, while linear bottlenecks encompass bottleneck layers within the network that minimize the dimensionality of feature maps before their expansion. These strategies are implemented to strike a delicate balance between computational efficiency and model capacity, enabling the development of deep networks that offer reduced computational demands without compromising performance.

In the neck segment of YOLOv8, deliberate choices are made to refine the flow of information and consolidate features across different scales. Among these choices is the incorporation of PANet (Path Aggregation Network), a pivotal component that facilitates efficient communication between network layers. Through PANet, YOLOv8 adeptly amalgamates information from various abstraction levels, thereby enhancing the coherence and effectiveness of feature representation across the network.

Additionally, YOLOv8 embraces BiFPN (Bidirectional Feature Pyramid Network) in its neck segment. BiFPN introduces bidirectional cross-scale connections, enabling seamless fusion of features across multiple levels within the feature pyramid. This bidirectional connectivity mechanism fosters reciprocal information exchange between scales, ensuring comprehensive feature integration and augmenting the model's ability to discern intricate spatial relationships within input data.

In crafting the head architecture of YOLOv8, deliberate design strategies are employed to streamline the execution of various tasks integral to object detection. Drawing inspiration from the architecture of YOLOv7, YOLOv8 embraces decoupled head architecture. This methodology entails the separation of tasks such as objectness scoring, classification, and bounding box regression, effectively alleviating potential

competition among these tasks and enabling autonomous treatment of each. Through this decoupling, YOLOv8 optimizes its object detection pipeline, ultimately achieving enhanced performance in object localization and identification within images.

Furthermore, YOLOv8 integrates adaptive mechanisms within its head architecture to refine its focus on intricate regions within images. These mechanisms may encompass the incorporation of attention mechanisms, empowering the model to dynamically allocate computational resources and prioritize significant regions during the inference process. By harnessing attention mechanisms, YOLOv8 endeavours to augment its proficiency in detecting objects, particularly in challenging scenarios characterized by occlusion or cluttered backgrounds.

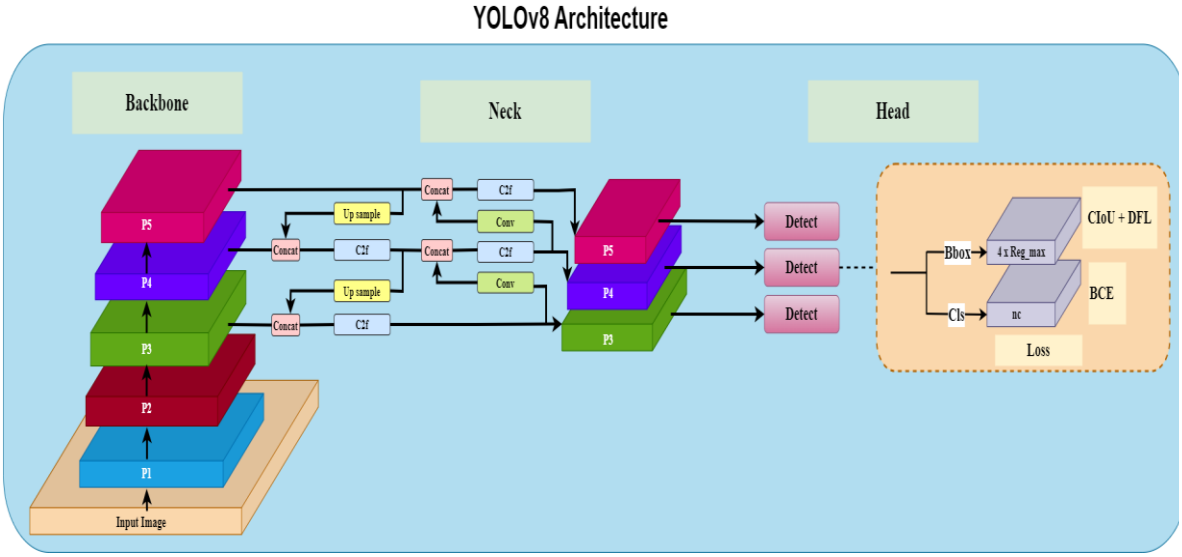


Figure 5.3: Basic Architecture of YOLOv8

The robustness and efficacy of YOLOv8, an advanced object detection model, stem from its meticulously crafted architecture, which integrates cutting-edge techniques to optimize performance across various aspects.

At the backbone level, the fusion of CSPNet, residual connections, inverted residuals, and linear bottlenecks works synergistically to enhance information flow, gradient propagation, and computational efficiency within deep neural networks. These architectural enhancements not only enable precise object detection but also render the model suitable for real-time applications and resource-constrained environments.

Moving to the neck segment, YOLOv8 incorporates PANet and BiFPN to further refine its feature extraction capabilities. These additions optimize information flow and feature fusion, allowing the model to extract pertinent representations from input data at varying scales. This nuanced feature integration significantly enhances YOLOv8's robustness and effectiveness in object detection tasks, particularly in scenarios characterized by diverse object scales and complexities.

Finally, YOLOv8's head architecture adopts a decoupled approach to task management, segregating objectness scoring, classification, and bounding box regression. This

streamlined operation, coupled with the integration of adaptive mechanisms like attention mechanisms, enables the model to focus more effectively on complex regions within images, ultimately enhancing its efficacy in object detection tasks.

In essence, YOLOv8's comprehensive architecture, encompassing enhancements at the backbone, neck, and head levels, positions it as a state-of-the-art solution capable of delivering precise and efficient object detection across a wide range of applications and environments.

5.4 Comparison based on the Results

Based on the results presented in the previous chapter, YOLOv8 demonstrated superior performance compared to YOLOv5, which in turn outperformed YOLOv7. These variations in performance can be attributed to differences in the build and architecture of each model. The primary architectural differences contributing to these performance variations are outlined below:

5.4.1 Backbone Network

YOLOv5 employs CSPDarknet53 as its backbone, striking a balance between depth and efficiency during feature extraction, which is essential for various tasks. In contrast, YOLOv7 utilizes a more complex and less optimized variant of CSPNet, which does not extract features as effectively for the prepared dataset. On the other hand, YOLOv8 incorporates an advanced backbone, significantly enhancing feature extraction efficiency.

5.4.2 Neck Design

The neck of the network combines features from various layers to enhance object detection capabilities. YOLOv5 employs a combination of Feature Pyramid Networks (FPN) and Path Aggregation Networks (PAN) for effective multi-scale feature aggregation. In contrast, YOLOv7 introduces modifications and additional layers that increase complexity but fail to enhance feature fusion for the given dataset. Conversely, YOLOv8 utilizes an advanced version of FPN and PAN, or even newer methods such as BiFPN, which provide superior feature fusion and improve the ability to detect objects at different scales.

5.4.3 Detection Head

The head of the network is responsible for predicting bounding boxes, objectness scores, and class probabilities. YOLOv5 employs an anchor-based detection head with CIoU loss, which is well-optimized for general object detection. In contrast, YOLOv7 features a decoupled head architecture that requires extensive tuning, resulting in anomalies in detecting certain classes. Conversely, YOLOv8 utilizes an optimized anchor-free detection head, improving precision and recall in object detection compared to previous versions.

5.4.4 Optimization Techniques

The optimization of training routines and hyper-parameters greatly influences performance and effectiveness. YOLOv5 benefits from well-established and optimized training routines and hyper-parameters. In contrast, YOLOv7's default parameters are not finely tuned, making the model sensitive to hyper-parameter adjustments. YOLOv8, however, features well-optimized and robust default parameters, resulting in superior performance with minimal tuning required.

CHAPTER 6: CONCLUSION

In this comprehensive study, we have meticulously developed an advanced model for the detection of traffic signals, utilizing the latest advancements encapsulated in the YOLOv8 architecture. This model, which has been specifically trained on a rich dataset representative of the diverse and often complex traffic scenarios encountered across India, stands at the forefront of efforts to significantly diminish the likelihood of vehicular accidents. It promises to bolster the safety of both pedestrians and motorists, especially in areas plagued by heavy congestion and on serpentine roads where visibility and predictability of traffic conditions are markedly reduced.

Following the model's successful testing phase, it has become evident that the performance of the YOLOv8 framework surpasses that of its predecessors — outperforming YOLOv5 by a margin of 2.6% and YOLOv7 by 9.5% in terms of mean Average Precision (mAP) in the context of traffic signal detection. This enhancement in detection capability not only speaks to the algorithm's refined accuracy but also its superior adaptability to the various traffic conditions prevalent in Indian settings. The challenge of accurately detecting traffic signals is non-trivial, compounded by an array of factors such as the unpredictability of weather conditions and the prevalence of substandard infrastructure, which collectively pose significant hurdles to reliable detection. Despite these considerable challenges, the deployment of the YOLOv8 algorithm within our model has led to extraordinary performance metrics, manifesting in a detection accuracy of 98.38% alongside a mean Average Precision (mAP) at an Intersection over Union (IoU) threshold of 0.50 reaching 0.985. These figures not only underscore the algorithm's robust capability in traffic signal detection under varied conditions but also highlight its pivotal role in the advancement of traffic safety mechanisms.

With the incessant increase in vehicular traffic, the task of maintaining effective oversight over road safety has grown increasingly daunting. In this context, the integration of state-of-the-art technologies such as the Internet of Things (IoT) emerges as a beacon of hope. By weaving a detailed network that encapsulates comprehensive details of every vehicle, it becomes feasible to implement a more stringent monitoring regime over traffic signal compliance, leveraging the constant connectivity to the internet that such a system would entail. Moreover, the application of our algorithm extends beyond mere traffic signal detection to encompass the identification of traffic signs, providing a multifaceted navigational aid to drivers. This enhanced functionality not only elevates the safety quotient of road travel but also marks a significant step forward in the quest to minimize, if not altogether eliminate, the occurrence of traffic-related accidents. Through this dual approach of leveraging advanced detection algorithms and IoT technologies, we envision a future where roads become markedly safer, significantly curtailing the frequency and severity of traffic incidents.

LIST OF PUBLICATION

[1] Supratim Biswas, SuvojitAcharjee, Asfak Ali, Sheli Sinha Chaudhuri, "**YOLOv8 based Traffic Signal Detection in Indian Road**", IEEE 2023 7th International Conference on Electronics, Materials Engineering Nano-Technology (IEMENTech) at Kolkata, India.

CHAPTER 7: BIBLIOGRAPHY

[1] Yehu Shen, Umit Ozguner, Keith Redmill, and Jilin Liu, "A Robust Video based Traffic Light Detection Algorithm for Intelligent Vehicles", 2009 IEEE.

[2] Masako Omachi, Shinichiro Omachi, "Traffic Light Detection with Color and Edge Information", 2009 IEEE.

[3] Chunhe Yu, Chuan Huang, Yao Lang, "Traffic Light Detection during Day and Night Conditions by a Camera", 2010 IEEE.

[4] Nathaniel Fairfield, Chris Urmson, "Traffic Light Mapping and Detection", 2011 IEEE

[5] George Siogkas, Evangelos Skodras, Evangelos Dermatas, "Traffic Lights Detection in Adverse Conditions Using Color, Symmetry And Spatiotemporal Information", February 2012.

[6] Ying Jie, Chen Xiaomin, Gao Pengfei and Xiong Zhonglong, "A New Traffic Light Detection and Recognition Algorithm for Electronic Travel Aid", 2013 IEEE.

[7] Zhilu Chen, Quan Shi, Xinming Huang, "Automatic Detection of Traffic Lights Using Support Vector Machine", 2015 IEEE.

[8] Florin Oniga, Sergiu Prodan, Sergiu Nedevschi, "Traffic Light Detection on Mobile Devices", 2015 IEEE.

[9] Dwi H. Widyantoro, Kevin I. Saputra, "Traffic Lights Detection and Recognition based onColor Segmentation and Circle Hough Transform", 2015 IEEE.

[10] Zhilu Chen, Xinming Huang, "Accurate and Reliable Detection of Traffic Lights Using Multiclass Learning and Multiobject Tracking" October 2016.

[11] Asaad F. Said, Mehrnaz Kh. Hazrati, Farshad Akhbari, "Real-Time Detection and Classification of Traffic Light Signals", October 2016.

[12] Karsten Behrendt, Libor Novak, Rami Botros, "A Deep Learning Approach to Traffic Lights: Detection, Tracking, and Classification", 2017 IEEE.

[13] Zhongrong Zuo, Kai Yu, Qiao Zhou, Xu Wang, Ting Li, "Traffic Sign Detection Based on Faster R-CNN", IEEE 37th International Conference on Distributed Computing Systems Workshops, 2017.

[14] Fei Wang, Yidong Li, Yunchao Wei, Hairing Dong, “Improved Faster R-CNN for Traffic Sign Detection” IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), 2020.

[15]Emin Güney, Cüneyt Bayilmis, “An Implementation of Traffic Signs and Road Objects Detection using Faster R-CNN”, Sakarya University Journal of Computer and Information Science, Vol 5, No 2, August 2021.

[16] P. Kanchanamala, A. Dharani, J Siri Chandana, D. Jyothsna, I Sudeep Kumar, G. Kushila, “Traffic Light Detection System in Self-Driving Cars”, 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC 2023).

[17]Benhe Gao, Zhongjun Jiang, Jiaman Zhang, “Traffic Sign Detection based on SSD”, Association for Computing Machinery, July 2019.

[18] Shuai You, Qiang Bi, Yimu Ji, Shangdong Liu, Yujian Feng, Fei Wu, “Traffic Sign Detection Method Based on Improved SSD”, Information 2020, 11, 475, October 2020.

[19] R. Krishna Paavani, V. Indraja, V. Neelimajyothi, S. Sai, M. Sriramullu, “Traffic Sign Board Detection Using Single Shot Detection (SSD)”, International Journal for Research in Applied Science & Engineering Technology (IJRASET), May 2022.

[20] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, “You Only Look Once: Unified, Real-Time Object Detection”, IEEE conference on Computer Vision and Pattern Recognition (CVPR), June 2016.

[21] Ruben J Franklin, Mohana, “Traffic Signal Violation Detection using Artificial Intelligence and Deep Learning”, IEEE 5th International Conference on Communication and Electronics Systems (ICCES 2020).

[22] Wael Omar, Impyeong Lee, Gyuseok Lee, Kang Min Park, “Detection and Localization of Traffic lights using YOLOv3 and Stereo Vision”, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science, ISPRS Congress, 2020.

[23] Branislav Novak, Velibor Ilić, Bogdan Pavoković, “YOLOv3 Algorithm with additional convolutional neural network trained for traffic sign recognition”, IEEE, Zooming Innovation in Consumer Technologies Conference (ZINC) 2020.

[24]Huibing Zhang, Longfei Qin, Jun Li, Yunchuan Guo, Ya Zhou, Jingwei Zhang, Zhi Xu, “Real-time Detection Method for Small Traffic Signs based on YOLOv3”, IEEE Access, Volume 8, 2020.

[25] Wan-Noorshahida Mohd-Isa, Md-Shakif Abdullah, Mahmood Sarzil, Junaidi Abdullah, Aziah Ali, Noramiza Hashim, “Detection of Malaysian Traffic Signs via Modified YOLOv3 Algorithm”, 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI)

[26] Ao Qin, Yu Jun Zhang, “YOLOV3 traffic sign recognition and detection based on FPN improvement”, Scientific Journal of Intelligent Systems Research, Volume 2 Issue 11, 2020.

[27] Feng Yu, Mingen Zhong, Shifu Tang, Zhonggang Zheng, "Improved traffic signal light recognition algorithm based on YOLO v3," Proc. SPIE 12173, International Conference on Optics and Machine Vision (ICOMV 2022).

[28]Chunpeng Gong, Aijuan Li, Yumin Song, Ning Xu, Weikai He, “Traffic Sign Recognition Based on the YOLOv3 Algorithm”, MDPI, Sensor Fusion and Advanced Controller for Connected and Automated Vehicles, 2022.

[29]S.Vinothkumar, S.Varadhaganapathy, R.Shanthakumari, S.Pradeev, S.Pragatheeswaran, K.Sankara Annamalai, “TRAFFIC SIGN DETECTION USING HYBRID NETWORK OF YOLO AND RESNET”, 2023 International Conference on Computer Communication and Informatics (ICCCI).

[30] Tai Huu -Phuong Tran, Jae Wook Jeon, “Accurate Real-time Traffic Light Detection using YOLOv4”, IEEE International Conference on Consumer Electronics – Asia (ICCE-Asia), 2020.

[31] Ziyue Li, Qinghua Zeng, Yuchao Liu, Jianye Liu, Lin Li, “An improved traffic lights recognition algorithm for autonomous driving in complex scenarios”, International Journal of Distributed Sensor Netowrks, 2021.

[32] Hui Gao, Wenhao Wang, Chengjin Yang, Weile Jiao, Ziewi Chen, Tong Zhang, “Traffic signal image detection technology based on YOLO”, Journal of Physics: Conference Series, ICCELA-VR 2021.

[33] Zhang Gan, Li Wenju, Chu Wanghui, Su Pan, “Traffic sign recognition based on improved YOLOv4”, ICIIBMS 2021, Artificial Intelligent, Robotics, and Human-Computer Interaction, Oita, Japan, 2021.

[34]Xiru Wu, Haozhe Cao, “Traffic Sign Detection Algorithm based on Improved YOLOv4”, Journal of Physics: Conference Series, ICAACE 2022.

[35] Ying Zhao, Yiyuan Feng, Yueqiang Wang, Zhihan Zhang, Zhihao Zhang, “Study on Detection and Recognition of Traffic Lights based on Improved YOLOv4”, MDPI, Sensors 2022.

[36] Daria Snegireva, Anastasiia Perkova, “Traffic Sign Recognition Application using YOLOv5 Architecture”, International Russian Automation Conference (RusAutoCon), 2021.

[37]Wentao Liu, Zhangyu Wang, Bin Zhou, Songyue Yang, Ziren Gong, “Real-time Signal Light Detection based on YOLOv5 for Railway”, IOP Conference Series: Earth and Environment Science, 2021.

[38]Yanzhao Zhu, Wei Qi Yan, “Traffic Sign Recognition based on Deep Learning”, Multimedia Tools and Applications 2022.

[39]Chuanxi Niu, Kexin Li, “Traffic Light Detection and Recognition Method based on YOLOv5s and AlexNet”, Appl. Sci. 2022, 12, 10808.

[40] Hua Khanh Huy, Nguyen Hoang Khang, Luyl-Da Quach, Hoang Tran Ngoc, “Traffic Lights Detection and Recognition Method using Deep Learning with Improved YOLOv5 for Autonomous Vehicle in ROS2”, 8th International Conference on Intelligent Information Technology (ICIIT), 2023.

[41] Shubham Gore, Manan Bhasin, Suchitra S, “Traffic Sign Detection using YOLOv5”, International Journal for Research in Applied Science & Engineering Technology (IJRASET), 2023.

[42] Avinash Padmakar Rangari, Ashwini Ravindra Chouthmol, Chaitanya Kadadas, Prashant Pal, Shashank Kumar Singh, “Deep Learning based Smart Traffic Light System using Image Processing with YOLOv7”, 4thInternational Conference on Circuits, Control, Communication and Computing, December 2022.