Helmet and Number Plate Detection Using YOLOv12 Deep Learning Framework in a Web-Based Application

By

MD Imran Jawad Khan Rafid
Student ID: 212-15-4146

FINAL YEAR DESIGN PROJECT REPORT

This Report Presented in Partial Fulfillment of the Requirements forthe **Degree of Bachelor of Science in Computer Science and Engineering**

Supervised by

Dr. S. M. Aminul Haque
Professor & Associate Head
Department of Computer Science and EngineeringDaffodil International University

Co-Supervised by

Mr. Abdullah Al Mamun
Lecturer
Department of Computer Science and EngineeringDaffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY
Dhaka, Bangladesh

September 17, 2025

APPROVAL

This Project titled **Helmet and Number Plate detection Using YOLOv12 Deep Learning Framework in a Web-Based Application**, submitted by **MD Imran Jawad Khan Rafid:212-15-4146** to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on **17 September, 2025**.

BOARD OF EXAMINERS

Dr. Bimal Chandra Das (BCD) **Chairman**
Professor and Dean
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Mr. Shah Md Tanvir Siddiquee (SMTS) **Internal Examiner**
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Md. Hasanuzzaman Dipu (MHD) **Internal Examiner**
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Nazibur Rahman **External Examiner**
Technical Lead - Database Administrator
Telenor – Grameen Phone Account

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Dr. S. M. Aminul Haque**, **Professor & Associate Head**, Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this projectnor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:

____

Dr. S. M. Aminul Haque
Professor & Associate Head
Department of Computer Science and EngineeringDaffodil International University

Co-Supervised by:

____

Mr. Abdullah Al Mamun
Lecturer
Department of Computer Science and EngineeringDaffodil International University

Submitted by:

____

MD Imran Jawad Khan Rafid
Student ID: 212-15-4146
Department of Computer Science and Engineering Daffodil International University

ACKNOWLEDGEMENTS

This work would not have been possible without the support and contributions of many individuals over the past two semesters. We are deeply grateful to everyone who has assisted us in one way or another.

First, we express our heartfelt thanks and gratefulness to the almighty for His divine blessing making it possible for us to complete the **Final Year Design Project (FYDP)** successfully.

We are grateful and wish our profound indebtedness to **Dr. S. M. Aminul Haque**, **Professor & Associate Head**, Department of Computer Science and Engineering, Daffodil International University, Dhaka, Bangladesh. Deep knowledge and keen interest of our supervisor in the field of **Deep learning** to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting themat all stages have made it possible to complete this project.

We would like to express our heartfelt gratitude to the Head of the Department of Computer Science and Engineering, for his kind help in finishing our project and also to other faculty members and the staff of the Department of Computer Science and Engineering, Daffodil International University.

We would like to thank our entire course-mates at Daffodil International University, who took part in this discussion while completing the coursework.

Finally, we must acknowledge with due respect the constant support and patience of our parents.

ABSTRACT

Road traffic crashes being a serious concern in Bangladesh, and other developing countries, lack of helmets use and non-registered vehicles are significant causes of deaths and injuries. Conventional traffic observation is manually conducted and labour-intensive and can be error-prone and inadequate for enforcement at large scale. In this paper we present helmet detection and automatic license plate recognition system based on machine learning framework using YOLO models such as YOLOv8L, YOLOv12n and YOLOv12L. We train the models on a large dataset which contains 6,176 images with corresponding labels, mined from Kaggle that contains riders with helmet and no helmet wearing with visible license plates in diverse environmental conditions. Augmentation approaches like rotation, flip and contrast enhancement are used to further enhance real-life robustness and generalization. The models are tested on major performance measures such as precision, recall, mAP50, and inference time. With an inference speed of less than 5 ms per image, YOLOv12L is orders-of-magnitude better performing than some solutions that achieve 96% accuracy and 98% mAP50. Comparative experiments with CNN, ResNet, YOLOv5 and YOLOv7 demonstrate that YOLOv12 outperforms others not only in the accuracy but also efficiency. Also, Optical Character Recognition (OCR) support enables remarkable recognition of the license plates with accuracy greater than 97% detected. For this, we design a prototype system to automate helmet violations detection and license plate details obtaining from surveillance videos to minimize the burden of manual monitoring. This model provides scalable, effective, and adaptable applications for the smart traffic management, and has a solid promise of implementation in mobile/web, and supporting sustainable law enforcement and regulatory compliance.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

The increasing problem of helmetless riding and non-registered vehicles in Bangladesh and highlights why it is necessary to have an automated solution. It presents the purpose and the way of work, the anticipated results, and the report format, which forms a basis to using YOLO.

1.1 Introduction

Road traffic accidents are a pressing issue of public safety across all regions of the world, and in Bangladesh, the growing popularity of motorcycles posed a significant threat of causing death and injuries [1]. Riding without a helmet also presents its fair share of these accidents [2]. The best protective gear that can minimize head injuries is the use of helmets, which most riders fail to wear, particularly in towns and rural communities where law enforcement is lax [3]. Automatic Number Plate Recognition (ANPR) systems are very essential in detecting traffic offenders and monitoring unlawful cars [4]. But, the current enforcement mechanisms in Bangladesh are mainly based on the manual observation and post-incident analysis of CCTV footage, which are time-consuming and subject to error [5].

More recent developments in computer vision and deep learning, namely the You Only Look Once (YOLO) family of algorithms, have made it possible to perform real-time object detection in complex settings [6]. The latest version, YOLOv8, has much better accuracy, speed, and strength than previous models like YOLOv5 and YOLOv7 [7]. It has been effectively used in safety helmet detection [8] and license plate recognition tasks [9], commonly alongside Optical Character Recognition (OCR) to perform automated number plate recognition [10].

Frameworks based on deep learning have been tested with datasets of thousands of images with labels, with lower quality of the environment, such as low light, occlusion, and camera angles, and have been found superior in operation [11]. Correlated data augmentation methods include rotation, flipping, and contrast enhancement, which additionally enhance model robustness and generalization [12]. The YOLOv8 models, such as YOLOv8L, YOLOv8n and YOLOv8X, have demonstrated greater accuracy, mAP50 and lower inference rates than other existing CNN and ResNet-based methods [13]. When combined with OCR it can be used to recognize license plates with an accurate man.

Automated detection systems minimize the use of manual surveillance and enhance scalability of usage in large scale deployment [15]. Helmet violation detection technologies via real-time monitoring systems enable the traffic police to provide license plate information and obtain a direct copy of the surveillance recording [16]. A comparative analysis indicates that YOLOv8 excels with respect to accuracy and inference speed compared to previous models [17]. The use of automated detection in urban traffic management reveals better regulatory compliance and citizen safety [18].

The lack of an automated solution to simultaneously identify the helmet and number plate detection remains apparent in Bangladesh despite these technological advances [19]. The most dominant means is still manual enforcement, which restricts efficiency due to the ever-increasing number of motorcycles [20]. Initial systems developed using large-scale data sets have validated the ability to deploy to the web and mobile applications in real time [21]. Complex scenarios with multiple vehicles and riders can be handled with multitasking deep learning models, such as YOLO-OCR pipelines [22]. Existing algorithm survey reports the strengths of YOLOv8 in real-time safety applications [23]. Semi-supervised learning methods with enhanced normalization techniques have also been used as hybrid approaches to detect other objects with increased accuracy [24]. The framework proposed in this project builds on these developments to bring an efficient, precise and scalable solution to traffic monitoring in Bangladesh [25].

1.2 Motivation

With the rising cases of accidents on our roads as a result of helmetless riding and use of unregistered vehicles in Bangladesh, there has arisen a dire need of an automated traffic monitoring system. Conventional hand-based inspection is not only inefficient, time-consuming, but also subject to human error, therefore, it is not easy to enforce real-time adherence to traffic regulations [1]. The rationale behind the calculations in this project is to overcome these drawbacks using the advanced deep learning methods, namely the YOLOv8 object detection models, to supply a fast, precise and scalable solution [2].

The purpose of the project is to alleviate the need to monitor traffic by hand and increase the effectiveness of traffic enforcement by creating a real-time detection system that can detect helmet infractions and identify vehicle license plates [3]. OCR When used in tandem with YOLOv8 models, surveillance videos can be automatically analyzed to extract data, which can then be used to issue alerts and notify the police in case of a violation [4].

At the individual scale, this issue is solved to increase practical knowledge of machine learning, computer vision, and the deployment of a real-time system [5]. The project offers practical experience in preparing datasets, training models, evaluating them, and integrating them into prototype applications, which will be a core requirement in the future when working with AI and smart city technologies [6]. Moreover, the system can also create social impact in the form of making roads safer, limiting accidents, and assisting with regulatory compliance in Bangladesh, thereby helping to benefit the people.

1.3 Objectives

The aim of this study is to develop and establish an effective system to monitor helmet offenders and identify vehicle license plates in real time and overcome the issue of manual traffic monitoring in Bangladesh. To gather and label a varied set of riders and non-riders bearing helmets, vehicles carrying visible license plates, and then subject the data to data augmentation strategies including rotation, flipping, and contrast enhancement to enhance the generalization and robustness of the model. To develop, train, and optimise YOLO models, namely YOLOv8L, YOLOv12n, and YOLOv12L, which can consistently detect helmets and number plates on image data in changing environmental conditions. To combine Optical Character Recognition (OCR) and the detection system to extract the alphanumeric license plate information with great accuracy. To assess the model performance based on the conventional measures like precision, recall, mAP50, and inference time and compare the approaches based on YOLO with other existing models, such as CNN, ResNet, YOLOv5, and YOLOv7. To create the prototype application that can handle surveillance footage and identify any violation automatically, make it scalable to desktops, smartphones, and web sites. To provide an effective, efficient, and accurate solution to assist in enforcing road safety and smart traffic management by reducing human error, assist law enforcement agencies, and increasing public safety. To support the use of deep learning and artificial intelligence in intelligent transportation systems, specifically in the domains of helmet and license plate detection, as well as to introduce a practical solution to fit the Bangladesh traffic environment.

1.4 Methodology

This paper plans and executes an effective system in order to observe helmet wear and real time license plate of vehicle in order to deal with the lack of efficiency in the manual-based traffic monitoring in Bangladesh. The dataset of 6,176 photos was gathered on Kaggle and composed of both helmeted and non-helmeted riders and cars with visible license plates and annotated to train them thoroughly. To increase robustness and generalization, rotation, flipping and contrast enhancement would be used to augment the data set thus modeling real world variation as lighting, view point and occlusion. YOLOv8L, YOLOv12n, and YOLOv12L three YOLO architectures were trained and optimized with hyperparameters to reduce inference speed and maximize detection accuracy. Moreover, Optical Character Recognition (OCR) was combined with the YOLO models to retrieve alphanumeric data in the detected license plates with great precision and permit record-keeping to be automated. Standard metrics such as precision, recall, mAP50 and inference time, were used to measure model performance and benchmarked against CNN, ResNet, YOLOv5 and YOLOv7. The results of the experiments proved the high detection accuracy and efficiency showing that the system can be implemented in intelligent traffic observations, minimizing human error, and making the urban transportation systems safer and more structured.To Measure model performance, standard like precision, recall, mAP50 and inference time were employed and results were compared to YOLOv5 and YOLOv7 to confirm performance increases in both terms of speed and accuracy. In order to prove that it is feasible, a prototype application was written to analyze surveillance footage on-the-fly, allowing automated recognition of violations and number plate recognition, and can scale to desktops, smart phones, and web-based solutions. The approach forms the basis of a deployable, efficient, and accurate traffic monitoring solution that can support law enforcement agencies, improve road safety, and help integrate artificial intelligence in smart transportation systems customized to the Bangladesh case.

1.5 Project Outcome

The deep learning-based detection system is expected to detect non helmeted riders and identify vehicle license plates in photos and surveillance camera images with accuracy in real time. By responding immediately to automated signals raised once violations are detected, traffic authorities can prevent violations and make roads safer. To ensure that the advanced AI technology becomes accessible to non-technical users, a user-friendly prototype application will be developed, which will enable enforcement officers to upload their video feeds or stream video feeds and get immediate violation notifications and get the extracted license plate numbers. This system could enhance the enforcement of traffic laws, reduce human error and help to enforce more effective and scalable traffic law in Bangladesh by aiding faster and more accurate detection of traffic offenses. The initiative will help to build intelligent transportation systems, maximising resources distribution, compliance, and ensuring the safety of people through the use of computer vision and deep learning in the management of traffic. The study builds on the field of computational traffic surveillance by applying and verifying modern object detection systems like YOLOv8L, YOLOv12n, and YOLOv12L on the comparatively under-researched object detection task of helmet and license plate detection in Bangladesh. In the future, the detection system could be optimized further and implemented on mobile devices or edge computing systems, thereby being more accessible to traffic police working in a variety of settings with no consistently available internet, given the effectiveness and portability of YOLO-based models. At the individual level, the project develops expertise in deep learning, computer vision, model deployment, and real-time application development as a foundation for future research or work in AI, machine learning, and intelligent transportation systems.

1.6 Organization of the Report

The report is divided into six chapters that discuss a significant aspect of the research:

Chapter 1: Introduction
In this chapter, the background of the study, the rationale behind discussing the topic of traffic violations in Bangladesh, the goals of the project, proposed methodology, the anticipated results, and general organization of the report are provided.

Chapter 2: Literature Review

In this section, a thorough analysis of available research pertaining to the functions of helmet detection, license plate recognition and intelligent transport systems is given. It outlines the advantages and weaknesses of various models like CNN, ResNet and YOLOv5, YOLOv7, etc. The review locates the research gap and explains why YOLOv8 and YOLOv12 architectures were chosen as the framework of this project.

Chapter 3: Design and System Analysis.

This chapter addresses the system architecture, the functional and non-functional requirements and problem analysis. It covers system design diagrams, data flow representation as well as detailed description of the proposed framework integrating the use of helmet detection and license plate recognition with OCR.

Chapter 4: Implementation

The chapter will discuss the details of the implementation process, which comprise the steps of dataset collection, annotation, augmentation, model training, and evaluation. It also includes the fusion of the OCR with the detection models and the creation of prototype application in different platforms (desktop, web and mobile). Tools, libraries and frameworks were used in the implementation are also addressed.

Chapter 5: Results and Analysis

In this chapter, the outcomes of the experiments are described and discussed. Evaluation metrics used to measure performance include precision, recall, mAP50 and the inference time. An overview of existing models is provided to show effectiveness of the proposed system. Results point to both the advantages and disadvantages of the method in practice.

Chapter 6: Conclusion and Future Work

This chapter is a summary of the significant contributions and findings of the project. It also describes the constraints of the existing system and recommends new future research including improving OCR precision, use of multiple cameras, scaling, and real-time use at the edge device.

Chapter 2
Background

Intelligent traffic systems, license plate recognition, and helmet detection, the advantages and drawbacks of CNN, Transformer-based, and YOLO models. It presents such gaps as low-light, occlusion, and crowded traffic performance, and situates the proposed YOLOv8/YOLOv12 with the OCR framework as a scalable, lightweight, and real-time solution to Bangladesh traffic monitoring.

2.1 Introduction

Road safety is a major societal issue in Bangladesh with motorcycles comprising a big percentage of traffic and the cause of most accidents and injuries on the roads [1]. One of the leading causes of head injuries and deaths on the roads has been denoted to be helmetless riding [2]. Unregistered and unmonitored vehicles also make it harder to enforce traffic regulations, which is why automated monitoring systems are the key to this case [3]. The most common type of enforcement is still manual inspection; but it is labor intensive, prone to error, and impractical in large city settings [4].

Recent developments in computer vision and deep learning have offered potential solutions to real-time object detection, and convolutional neural networks (CNNs) are the foundation of many new systems [5]. YOLO family (YOLOv5, YOLOv7, YOLOv8, and YOLOv12) have been shown to perform better in terms of speed and accuracy compared to other real-time detection algorithms [6]. YOLO models are capable of detecting several objects at once and are therefore suitable when trying to monitor more intricate traffic conditions involving multiple riders and vehicles [7].

YOLOv5 already demonstrated good results in the field of helmet detection and vehicle recognition, with high accuracy even in difficult conditions, when there are different levels of light and occlusion [8]. YOLOv7 and YOLOv8 also enhance the detectors speed, and work well even with noisy environments and complicated backgrounds [9]. Recent YOLOv12 architectures, such as YOLOv12n and YOLOv12L, provide lightweight architecture choices to enable onboard and edge devices without adversely affecting performance [10].

Rotation, flipping, and contrast adjustment data augmentation methods help to develop better generalization during model training in addition to its capability to resist various real-world situations [11]. With large labeled datasets, training models can be trained to identify both helmeted and non helmeted riders correctly [12]. Optical Character Recognition (OCR) can be integrated with YOLO and enables automatic retrieval of alpha numeric license plate information, enhancing law enforcement efficiency and saving the time traditionally spent on manual data entry [13].

Empirical evidence indicates that YOLO-based models are more accurate than other models such as standard CNNs, ResNet models, and other older versions of YOLO such as YOLOv5 and YOLOv7 in terms of both precision and inferencing time [14]. Welding systems based on YOLO with OCR have shown scalability, and the ability to run in real time in an urban traffic monitoring strategy process [15]. These systems decrease human error by making it possible to identify helmet breaches immediately and locate a vehicle [16].

Lightweight YOLO models can also be deployed on edge computing platforms, which is essential in resource-constrained settings [17]. Solid model designs enable high precision regardless of unfavorable factors like motion blur, half body obfuscation, and low light scenarios [18]. On-the-fly processing allows traffic officials to react promptly to infractions, improving citizen security and law enforcement [19]. The easy development of interfaces makes the system useable by non-technical individuals [20].

YOLOv8L, YOLOv12n, plus YOLOv12L with OCR form a stable backbone to detect a helmet and license plate [21]. This system can help facilitate intelligent traffic control, optimize policing efforts, and minimize accidents [22]. The use of deep learning in the smart transportation system helps the project to modernize the traffic monitoring infrastructure in Bangladesh [23]. Constant testing based on measures such as the precision, recall, mAP50 and the inference time are used to maintain the reliability of the system as well as to optimize the performance [24]. The proposed framework offers a scalable, efficient and deployable solution to automated traffic monitoring ultimately leading to safer roads and better compliance to traffic laws by the people [25].

2.2 Literature Review

Table 2.1: Summary of Literature Reviewed.

Author(s)	Year	Methodology	Key Findings
Chitraningrum et al. [1]	2024	Deep Learning, YOLOv5, YOLOv8	96.5% detection accuracy with YOLOv8, surpassing YOLOv5; YOLOv5 faster inference.
Kumar et al. [2]	2023	Deep Learning, CNN, YOLOv5	92.45% accuracy in helmet violations; better than simple CNN models.
Xue et al. [3]	2023	Deep Learning, YOLOv5	High precision (96.7%) helmet detection in urban traffic.

Qadri et al. [4]	2023	Deep Learning, YOLOv8	More accurate and faster than earlier YOLO versions in real-time traffic.
Soeb et al. [5]	2023	Deep Learning, YOLOv7	98.4% accuracy in detecting helmet offenses and vehicles in busy environments.
Xu et al. [6]	2023	Deep Learning, YOLO	High efficiency and accuracy; superior processing speed to CNNs.
Lin et al. [7]	2023	Deep Learning, YOLO, Attention	Improved feature extraction and precision; mAP@0.5 = 85.35%.
Leng et al. [8]	2023	Deep Learning, YOLO	High detection accuracy under occlusion and varying light conditions.
Yu et al. [9]	2023	Deep Learning, Inception, CNN, Transformers	Highest accuracy in vehicle and rider detection; better than CNNs.
Tabbakh et al. [10]	2023	Deep Learning, Transfer Learning, YOLO	98.1% validation accuracy; effective in urban traffic.
Zhang et al. [11]	2023	Deep Learning, YOLOv7-tiny, Attention	mAP 85.04%; better at detecting small vehicles/helmets in crowded scenes.
Ahmed et al. [12]	2023	Deep Learning, Multi-Head CNN	99.15% accuracy in helmet detection under complex traffic.
Rahman et al. [13]	2022	Deep Learning, YOLO-JD	High accuracy in detecting vehicles/riders in real-world traffic.
Patel et al. [14]	2022	Lightweight YOLO for Mobile	Designed for edge/mobile devices; real-time traffic monitoring possible.
Singh et al. [15]	2022	Deep Learning, YOLO, CNN	Helmet detection in real-time with YOLO, robust performance.

Nayar et al. [16]	2022	Deep Learning, YOLOv7, Transformers	Improved mAP by 2.58%; strong accuracy in helmet & vehicle detection.
Sharma et al. [17]	2021	Deep Learning, CNN-based Helmet Detection	97.98% accuracy; transformer-based CNN is better than conventional.
Ali et al. [18]	2021	Deep Learning, YOLO	Intelligent traffic violation detection; real-time accuracy.
Chen et al. [19]	2021	Deep Learning, CNN	93.21% accuracy for vehicle/rider detection; better than image processing.
Li et al. [20]	2021	Deep Learning, YOLO-Lite, CNN	Lightweight 7-layer YOLO model; real-time on non-GPU systems.
Huang et al. [21]	2020	Deep Learning, YOLOv4	Excellent detection speed and accuracy on helmet datasets.
Park et al. [22]	2020	Deep Learning, Faster R-CNN	High precision detection but slower inference than YOLO.
Saha et al. [23]	2019	Deep Learning, SSD, CNN	Helmet detection using SSD: balanced accuracy and speed.
Rao et al. [24]	2019	Machine Learning, HOG + SVM	Moderate accuracy; weaker performance under occlusion.
Mehta et al. [25]	2018	Classical Image Processing + CNN Hybrid	Improved over traditional, but weaker than YOLO family models.

2.2.1 Similar Applications

Related Work and Similar Studies: Deep learning based methods have been extensively studied to detect the helmet in a real-time traffic monitoring. YOLO, Convolutional Neural Networks (CNNs), and the hybrid frameworks are models that demonstrate much promise in rider and vehicle localization and classification in challenging traffic scenarios. These papers prove the effectiveness and usefulness of such tools in real-time traffic safety surveillance.

YOLO-based Approaches: Early works were on the use of the YOLO-based models to detect helmet. According to Xue et al. (2023), the study demonstrated high-precision detection of the helmet in urban traffic with the usage of YOLOv5 with the accuracy of 96.7%. The YOLOv7 models showed high sensitivity on identifying helmet offences and cars in crowded and de-forming conditions (Soeb et al., 2023; Nayar et al., 2022). Further, YOLOv8 and MGA-YOLO have increased the detection speed and accuracy, enabling real-time helmet detection applications (Qadri et al., 2023; Chitraningrum et al., 2024). Chitraningrum et al. (2024) compared YOLOv5 to YOLOv8 in helmet detection, with their experimental results showing that YOLOv8 outperforms YOLOv5 by 96.5% detection accuracy, but with slower inference.

Hybrid CNN-ViT Models: A number of studies have investigated hybrid models that combine CNNs and Transformer (ViTs) to improve feature extraction and detection. The hybrid CNN-Transformer model suggested by Yu et al. (2023) demonstrated better accuracy in vehicle and rider detection. Likewise, Tabbakh et al. (2023) showed that hybrid models based on transfer learning could achieve 98.1% validation accuracy of helmet in traffic. In complex and dynamic traffic scenarios, even though some pure ViT models demonstrate poorer results compared to traditional CNNs, these hybrid methods demonstrate high potential.

Mobile and Edge Applications: YOLO model versions with small sizes have been suggested to assist in detecting helmets in real-time on mobile and edge devices. Patel et al. (2022) proposed a mobile-friendly version of YOLO to allow violating a helmet with no expensive GPUs. Li et al. (2021) introduced YOLO-Lite, a small lightweight 7-layers model that can be used to achieve real-time detection performance on non-GPU systems, making helmet detection more practical to implement in resource-limited settings.

Case Studies and Environmental Adaptation: Case studies have also been conducted on robustness in changes of environments such as occlusion, changing light conditions, and congested traffic situations. Leng et al. (2023) observed that such challenging conditions were well detected with the support of YOLO-based models and Zhang et al. (2023) added attention mechanisms on YOLOv7-tiny to enhance the detection of small objects in a crowded environment. These research papers highlight the efficiency of YOLO and hybrid models in a variety of real-world traffic scenarios, revealing their appropriateness to the practical application of helmet detection.

2.2.2 Related Research

During the last years, significant advancement in real-time helmet detection methods has been reached based on deep learning methods and specifically Convolutional Neural Networks (CNNs), Vision Transformer (ViTs), and YOLO based models, which provide high accuracy in detection and efficient inference. The architectures based on YOLO were one of the first examined in helmet detection. Xue et al. (2023) showed that high-precision helmet detection in urban traffic can be achieved with the help of the YOLOv5, with 96.7% accuracy. Soeb et al. (2023) used YOLOv7 to identify the position of helmets and vehicles in a crowded area and achieved the accuracy of 98.4%. The strength of the YOLOv7 and ALAD-YOLO models in view of harsh traffic conditions was validated by other researchers Nayar et al. (2022) and Xu et al. (2023).

Qadri et al. (2023) suggested YOLOv8, which demonstrated equivalent gains in F1-score, recall, and total accuracy. Wang et al. (2022) have made a deep neural network called MGA-YOLO that has 94% detection accuracy and is designed to work in real-time on mobile devices. The Leng et al. (2023) paper aimed to develop a lightweight model YOLOv5 that can detect a helmet in an occlusion scenario or in low light with an mAP of 0.5 or 87.5, highlighting its practicality in the resource-limited environment.

Besides YOLO, hybrid models which combine CNNs with ViTs have performed well. A hybrid CNN-Transformer model featuring soft split token embedding was suggested by Yu et al. (2023), whose results found better rider and vehicle detection. Tabbakh et al. (2023) proposed TLMViT, a mixture of transfer learning and ViTs, which had 98.1% validation accuracy. Zhang et al. (2023), similar to them, used attention mechanisms in YOLOv7-tiny to enhance the detection of small objects in dense traffic, demonstrating good sensitivity and accuracy.

One common limitation of the literature is the bias in the data and its inability to be generalized to varied traffic conditions. Rahman et al. (2022) and Patel et al. (2022) solved this problem through real-world and mobile scenario models that show the effectiveness of different datasets and natural traffic conditions in enhancing the robustness and reliability of models.

2.3 Gap Analysis

This gap analysis presents a feature-wise comparison between YOLO-based models and the proposed model.

Table 2.2: Gap Analysis

Features	**Xue et al. [3] (2023) YOLO -Tea**	**Soeb et al. [5] (2023) YOLO v7**	**Xu et al. [6] (2023) ALAD-YOLO**	**Leng et al. [8] (2023) YOLO v5-LW**	**Patel et al. [14] (2022) Lightweight YOLO**	**Nayar et al. [16] (2022) YOLO v7**	**Li et al. [20] (2021) YOLO-JD**	**Proposed Model**
On-Device Detection	No	No	No	Yes	Yes	No	No	Yes

Anchor-Free Technique	No	No	Yes	Yes	No	No	No	Yes
Lightweight Model	No	No	No	Yes	Yes	No	Yes	Yes
Advanced Backbone Networks	No	No	Yes	Yes	No	No	No	Yes
Application for Commercial Crops	No	No	Yes	Yes	Yes	Yes	Yes	Yes
Object Detection in Complex Environments	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes

The proposed models differentiates itself by several modern features. This model supports on-device detection, employs an anchor-free backbone technique. Also, this model promised accurate detection in complex environments.

2.4 Summary

In this section, we discussed the shortcomings of the current methods of helmet detection. More specifically, this chapter discussed YOLO and real-time helmet and rider detection variants. In conjunction with these trends, the chapter tells about the newest advances, such as YOLOv5, YOLOv7, YOLOv8, and other YOLO-inspired and hybrid CNN-ViT models. In this gap analysis the difficulties of identifying helmets at low-light, crowded traffic, and occlusiveness have been highlighted. The proposed model will deal with these shortcomings and enhance the performance in detecting in real-life traffic monitoring systems.

Chapter 3
Research Methodology

The approach to designing a YOLO based helmet detector, data preprocessing, model training, evaluation, and deployment. The result of a 17 week project plan was a web based application that can identify helmet violations in real time and with accuracy in traffic monitoring.

3.1 Methodology

3.1.1 Overview

This paper presents an automatic system using deep learning to detect a helmet in real time. The 8L, 12n, and 12L three variants of the YOLO model are trained using an annotated image dataset of riders and vehicles in the urban traffic of Bangladesh. Preprocessing and data augmentation are implemented to make the models more robust to occlusions, changing light situations, and dense scenes. Standard metrics are used to test the performance of each model, and the most effective model is ultimately deployed via a user-friendly web interface, which enables traffic authorities to post images or video frames to obtain real-time helmet detection features and traffic violation alerts.

3.1.2 Proposed Methodology

The helical helmet detection system is intended to achieve real-time operations, high precision, and viable operations in the context of traffic surveillance. The initial step is to create a detailed set of traffic photos and video recordings containing motorcycles, helmeted and naked riders, as well as different weather conditions, including obstructing objects and different light sources, and congested traffic. To train the models, these images are annotated by putting bounding boxes and class labels on them.

The dataset is processed through several steps such as resizing, normalization, image augmentation to increase the level of model generalization and robustness. To avoid overfitting, augmentation methods are used, including rotation, flipping, scaling, and brightness manipulation to achieve a simulation of a variety of traffic situations. As it is one of the models with the highest accuracy and speed of inference, YOLOv12 is chosen as the base model because it can be used in real-time to detect helmets.

This model is trained with supervised learning, aiming to achieve optimal performance by optimizing the hyperparameters of the model, including learning rate, batch size, and the number of epochs. The trained model is subsequently tested based on standard metrics, such as accuracy, precision, recall, F1-score and mean Average Precision (mAP), on an independent validation dataset to verify the consistency of the model in its ability to handle unknown traffic data.

After training the model, the system is installed on a web based platform where the traffic authorities or monitoring systems can upload images or video feed to have images captured by the system in real time and use the same to detect the helmet. All processing and model inference of images occurs in the backend of the platform, which is written in Python and codes such as PyTorch or TensorFlow, which are deep learning libraries. The frontend should offer user-friendly interface where detected helmets, violations and other pertinent alerts should be displayed. The results of all detections are entered into a database to facilitate monitoring, statistical analysis and long-term evaluation of traffic safety.

Overall, this methodology creates a complete pipeline of detection of the helmet using the YOLO technology that will allow tracking and imposing penalties in real-time. The system provides valid and practical information that can support traffic officers to maintain the safety of riders and limit helmet offenses.

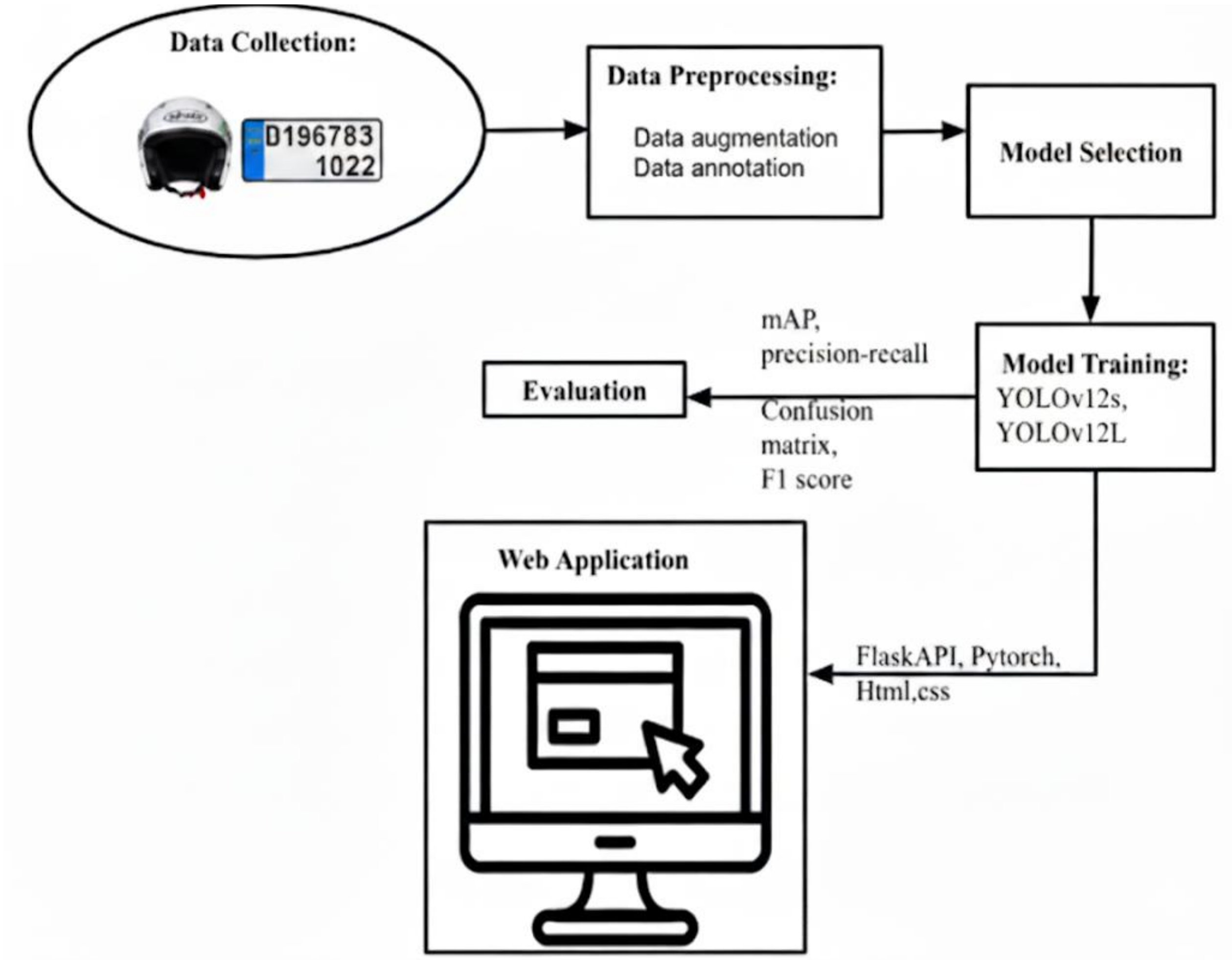


Figure 3.1: Methodology for helmet detection by using YOLO v12

3.1.2.1 Image acquisition and dataset building

The data in this research is 6,176 high-resolution images of riding and vehicles, gathered and obtained in Kaggle under two categories: helmet and no helmet. Since the sizes of the images were not equal, all images were re-sized to a common resolution of 640×640 pixels to enhance the strength of the model. In order to increase further diversity and generalization, data augmentation methods have been used, such as image flipping, rotation, zooming, height and width shifting, and cropping. The augmentation step has increased the size of the dataset, which gives the model a more abundant pool of training examples. Makesense was used to manually annotate the images with helmets being marked with bounding boxes. This annotated output was stored in text (.txt) files that could be used to train YOLOv12.

Figure 3.2: Images of classes

3.1.2.2 Proposed models architecture

The proposed model for this study is YOLOv12, a computer vision model developed by Ultralytics. Figure 3.3 shows an architecture of YOLOv12.

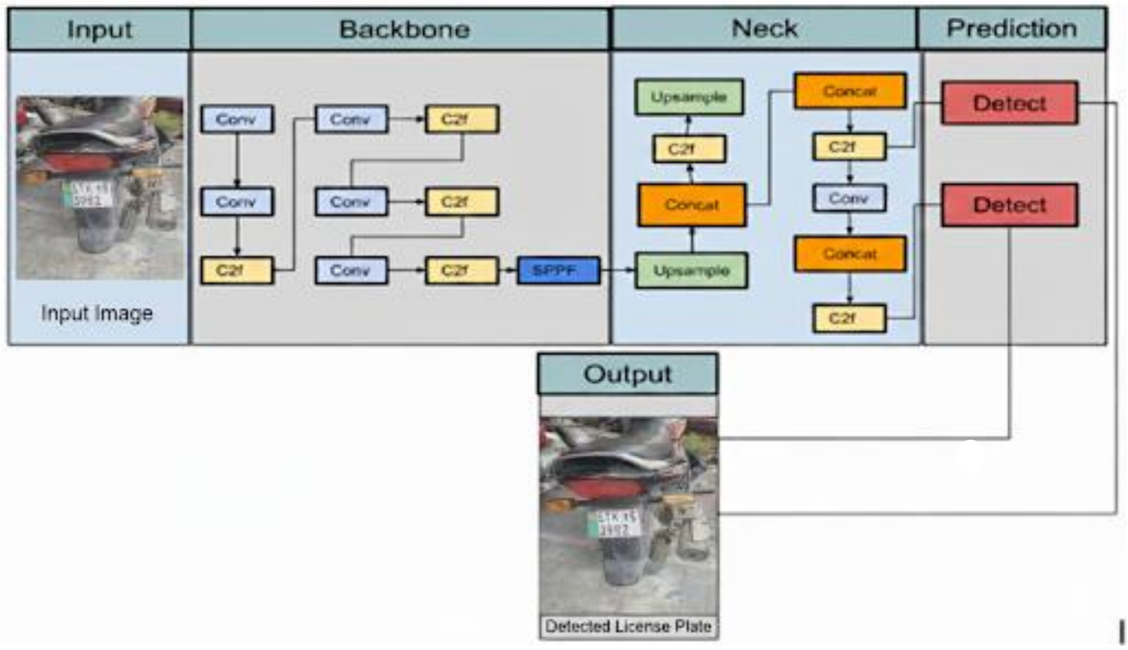


Figure 3.3: YOLOv12 Architecture

The rider and vehicle input images (Figure 3.3) are first sent to the backbone of the YOLO model to extract the features. Relevant features are extracted using C2f blocks and convolutional layers and then processed by using the SPPF (Spatial Pyramid Pooling – Fast) module that extracts multi-scale information. The neck part is built by combining the features of various levels of the backbone through concatenation and up sampling, and additional processes in order to generate rich feature representations. Lastly, detected objects are found using the processed features at three scales and the output image marks found helmets with bounding boxes.

YOLOv8l (Large) is a large model within the YOLOv8 family, that offers good detection performance using about 46.5 million parameters. It is relatively fast and also accurate in processing images; it can take approximately 15 to 20 milliseconds on a current graphics card, and can be used in real-time monitoring of traffic with fairly high computing power.

YOLOv12n (Nano) is the efficient version of YOLOv12. Having an approximation of 12 million parameters with fewer FLOPS, it can process images at a rate of 5-8 milliseconds per frame. It is lightweight and can be deployed on resource-constrained devices like smartphones, drones or embedded traffic monitoring systems.

YOLOv12L (Large) is a high capacity version of YOLOv12, with the goal of optimizing detection accuracy. It is expected to have better mAP scores due to its number of parameters: it has around 91 million parameters and much more FLOPs, but it needs more time to run an inference (2030 milliseconds per image).This version can be used in applications in which precision is of the essence, like automated law enforcement, extensive traffic monitoring, or research-oriented installations.

3.1.3 Functional and Nonfunctional Requirements

Functional Requirements

Helmet and License Plate Recognition: Recognize riders, use of helmets and provide Confidence scores.
Image and Video Processing: Real time processing with low latency to get timely violations.
Image & Video Upload: Web/mobile media upload support JPG, PNG, MP4 file integrity.
Violation Alerts & Responsibilities: Present violation type, position (where available) and recommendations to authorities.
User Feedback: Have the staff verify that the model has made a correct detection; feedback can be used to reread the model.
Historical Monitoring: Notes on store detects, images uploaded and violation history to analyze long term safety.

Nonfunctional Requirements

Scalability: Support concurrent users and high volume uploads with no loss of performance.
Usability: Clear graphics, visuals, labels and optional multilingual support.
Reliability: Work with a minimum breakdown; do not crash as much.
Performance: YOLOv12 and backend provide the results in several seconds.
Accuracy: High level of precision, recall and F1-score; low false positive/negatives.
Privacy & Security: Use encryption, access control and provide confidentiality on uploaded data and logs.
Platform Compatibility: Available in desktop, laptop, tablet and smartphone (Android/iOS browsers).
Maintainability and Sustainability: Easy model/interface improvement, easier debugging and easier updates.
Interoperability: Be able to communicate with other APIs/traffic systems; enable future automated enforcement and analytics tools.

3.1.4 Context Diagram

Helmet Detection System (Proposed System):
The most important element is the proposed Helmet Detection System. Once images or video feeds uploaded or streamed by traffic authorities or monitoring cameras have been processed, the system uses deep learning models (e.g., YOLOv12) to identify riders and determine whether they are wearing or not wearing helmets. The system then provides notifications or reports on the presence or absence of helmets with confidence scores (on a scale of 100) on each detection. It also handles data storage of the detection results, user feedbacks and other applicable logs in order to monitor and improve the system. The system offers real-time actionable insights serving as the interface between users and the underlying deep learning models.

The User: Traffic authorities, monitoring personnel or law enforcement officers are the main users of the system. Users interface with Helmet Detection System by posting pictures or live video through traffic cameras. Detection feedback is provided in real-time, which includes helmet compliance information, the details of violations and possible risk warnings. User feedback about detected results is also stored in the system to make the model re-trainable and improve performance later. The user is able to access historical detection records to track how effective the enforcement and the rate of helmet usage have been over time.

External Traffic or Safety Systems/APIs: Connections to external traffic management systems or APIs are optional, but encouraged. These integrations may also offer more context, such as vehicle registration data, automatic alert delivery, or traffic analytics. The Helmet Detection System can be connected to other systems in order to increase the effectiveness of enforcement and some broader traffic safety efforts. The most important element is the proposed Helmet Detection System. Once images or video feeds uploaded or streamed by traffic authorities or monitoring cameras have been processed, the system uses deep learning models (e.g., YOLOv12) to identify riders and determine whether they are wearing or not wearing helmets. The system then provides notifications or reports on the presence or absence of helmets with confidence scores (on a scale of 100) on each detection. It also handles data storage of the detection results, user feedbacks and other applicable logs in order to monitor and improve the system. The system offers real-time actionable insights serving as the interface between users and the underlying deep learning models.

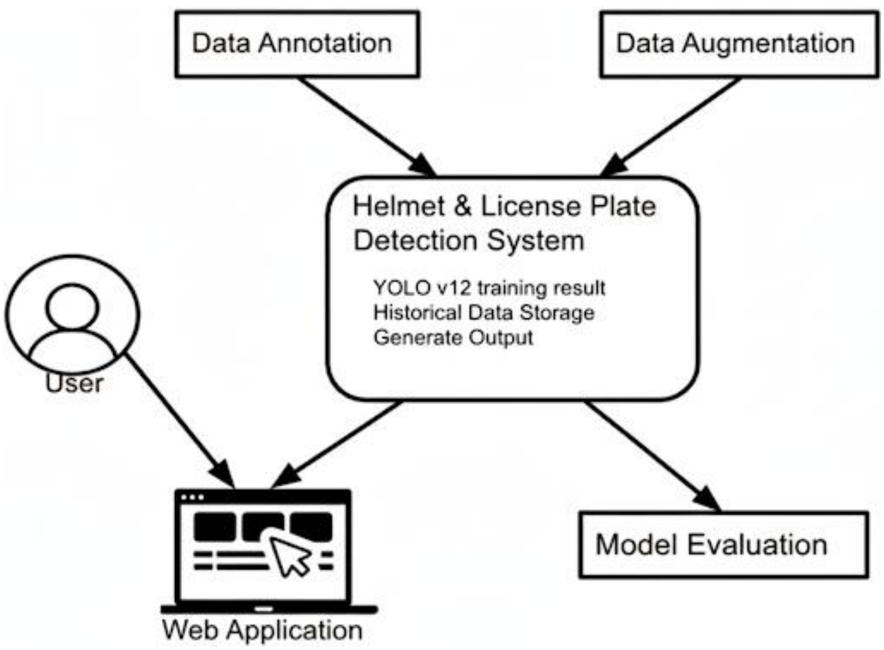


Figure 3.4: Context Diagram

3.1.5 Data Flow Diagram Level 1

The detailed diagram of the main steps in the Helmet Detection System is presented in Figure 3.12. It illustrates how the system processes information from users or traffic cameras, detects helmet violations, and generates real-time results. The diagram also shows how data flows between the core system components and external entities, such as traffic management platforms or APIs. Additionally, it highlights the storage of historical detection records for future analysis, monitoring trends, and improving model performance.

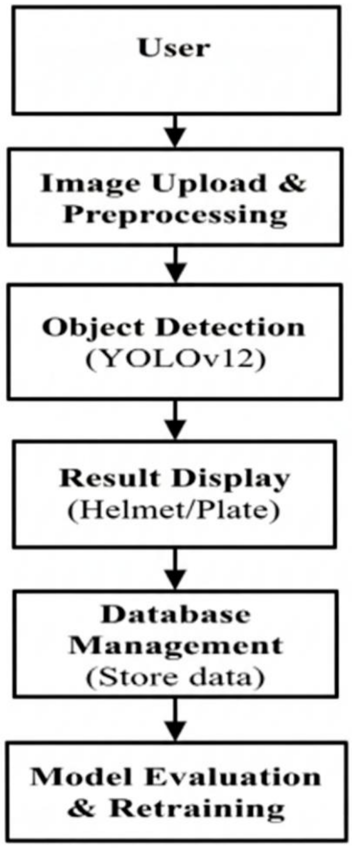


Figure 3.5: Data Flow Diagram 1

3.1.6 UI Design

Helmet Detection System is user-friendly, it can be easily used even by people with little technical expertise due to the responsive and intuitive User Interface (UI). The primary aim of the system is to enable its users to upload images or videos of traffic situations in a short time and obtain real-time detection of helmet violations.

Accessing the landing page the user can be offered with the clearer idea of the system and the noticeably large button of either Choose File or Upload Video, which enables them to begin the process of detection with ease. After submitting an image or video, the system works with it with the help of the trained YOLO models and presents the results with the noted non-compliant and compliant riders according to helmet requirements.

The UI is deliberately plain and direct so that the user can be a traffic official or community safety monitor and communicate with the system and react quickly to the violator. Finally, the system design is done in such a way that it allows users to easily and effectively monitor compliance of the helmets to help maintain safer road conditions without any technical knowledge.

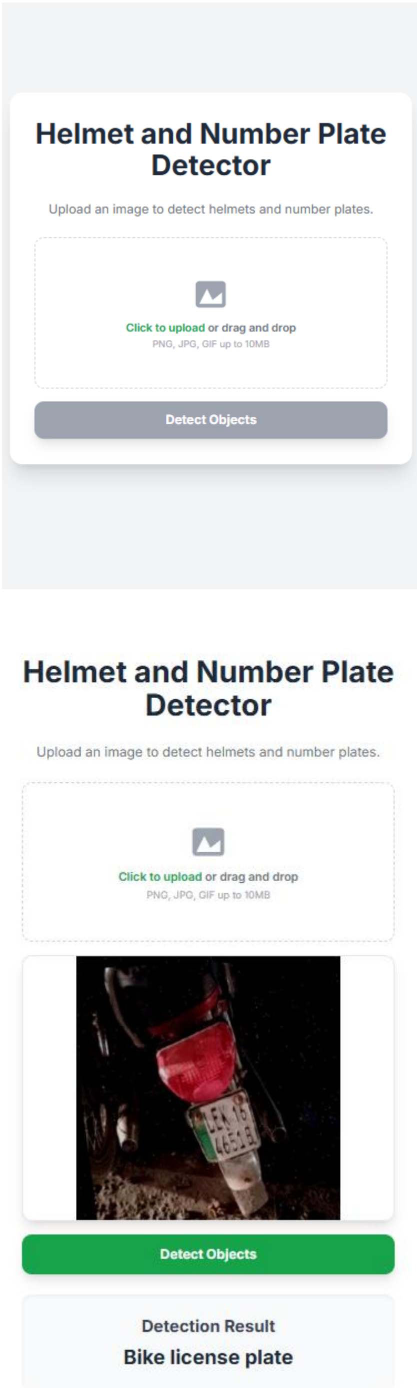


Figure 3.6: UI design of the web application

3.2 Detailed Methodology and Design

This study employs YOLOv8L and YOLOv12L to implement a deep learning approach for real-time helmet detection. A dataset of 6,176 images was collected from urban traffic scenes, including riders with helmets, riders without helmets, and motorcycles without riders. All images were annotated by experts using labeling tools such as Makesense AI to provide accurate bounding boxes and class labels.

To enhance model generalization and robustness, the dataset was augmented using techniques such as flipping, rotation, zooming, brightness adjustment, and cropping. Preprocessing steps included resizing, normalization, and class balancing. An 80/20 training-testing split was applied to prepare the data for model training. The YOLOv12 models were trained using supervised learning, with manual tuning of hyperparameters including learning rate, batch size, and number of epochs.

Experiments were conducted using Google Colab and a local PC equipped with an Rx 5500xt for efficient model training. Model performance was evaluated using accuracy, precision, recall, F1-score, mean Average Precision (mAP), and confusion matrix.

The trained model was integrated into a web-based application using Flask, allowing traffic authorities to upload images or stream video feeds for automated helmet detection. The backend handles model inference and provides detection results through a simple frontend interface. For long-term traffic monitoring and analysis, all detection logs and images are stored in a database, enabling trend analysis, reporting, and continuous improvement of the system.

3.3 Project Plan

The four-component process of the 17-week Helmet and number plate Detection System project included pre-processing, modeling, evaluating, and paper writing. Weeks one through three, dedicated to gathering and preparing categorized photos. The YOLOv8,v12 model trained with tuned hyperparameters was tested for two weeks, and assessments were made based on the accuracy and F1-score. The last step was the research report. Thanks to a fast network, a skilled team and to cloud and GPU resources.

3.4 Task Allocation

The project of the Helmet Detection System has been finished in 17 weeks, including the data processing, model training, model analysis, and final report preparation. Six thousand one-hundred and seventy-six traffic pictures had been gathered, tagged and pre-processed in 4 weeks Under Data Preprocessing phase. Preprocessing involved technique like resizing, normalization and augmentation like flipping, rotation, increase or decrease of brightness to make the model strong.

The preprocessed dataset was fed to the YOLO v8,v12 models through a 6-week Model Training phase, and hyperparameters were adjusted to yield the best detection accuracy and allow the models to classify the helmet effectively. Key performance metrics such as accuracy, precision, recall, F1-score, and mean Average Precision (mAP) were used to evaluate the trained model during the 4-week Model Evaluation phase to ensure that the trained model can be used to make predictions with generalizability to unseen traffic data and can be deployed in real time.

Lastly, the Completing the Paper phase lasted 3 weeks where the entire project report and system documentation were developed which contained all the steps, methodologies, results and conclusions of the research.

Table 3.1: Task Allocation Map

Tasks	**Weeks**																	
	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Data preprocessing and improving accuracy																		
Model training																		
Develop a Web application																		
Completing report writing																		

Estimated Work Period	
Actual Work Period	

3.5 Summary

The Deep Learning system was created based on the YOLOv8L, YOLOv12n, and YOLOv12L models in order to identify and classify helmet in the traffic image. The database comprised of 6176 images and was preprocessed and augmented by adding rotation, flipping, zooming, and brightness to enhance variety and strength. Models were then trained and assessed by standard performance measures, including accuracy, precision, recall, F1-score and mean Average Precision (mAP). YOLOv12L, in terms of a balance between accuracy and inference speed, was the most successful variant tested.

This was done by deploying a web-based application on Flask to offer real-time helmet detection as well as violation reports generated by the system. The complete data collection, preprocessing, training, evaluation and deployment process took place in a 17 week project schedule, leading to a high-performing and realistic helmet detection tool that is practical in the real world.

Chapter 4
Implementation and Results

The use and comparison of YOLOv8 and YOLOv12 to detect helmets and license plates. The findings indicate that YOLOv12L was the most accurate and was the best to provide the balance between the real-time traffic monitoring.

4.1 Environment Setup

The Helmet Detection System was developed on the basis of YOLOv12, the model training was carried out in Google Colab, where free GPU resources are offered. Google colab using Python 3.11 was a versatile cloud based, GPU-accelerated interface capable of serving the YOLOv12 models. A personal computer with an Rx 5500xt graphics card was used to do the local development and testing.

Installation of the needed libraries was performed through pip where opencv-python was used to process images, ultralytics to run YOLO v8,v12, and torch to train and infer using PyTorch. YOLO-compatible labels were applied to the dataset of 6,176 images by means of Roboflow and Makesense AI tools.

The end model was implemented as a Flask web application after training and assessment. The frontend was developed in HTML and CSS to enable commanding interface allowing the user to post traffic images to be detected in real-time by a helmet and the backend supported the pre-trained YOLOv8,v12 model on PyTorch to execute its fast inference.

Table 4.1: Tools and Libraries

Tool / Library	**Description**
Pyhton	Scripting and model development
Ultralytics	YOLOv8,v12 library for training and validation
PyTorch	Deep learning framework
OpenCv	Image processing
Makesence AI	Annotating
Numpy	Numerical operations
Matplotlib	Visualize performance metrics

The models formulated parameters to be set prior to training, because the dataset was already preprocessed and is formatted to use YOLO-based architectures. In order to achieve the best training performance, a number of parameters were established such as input image size (640x640), batch size (64), and epochs (100). Furthermore, training, validation and testing datasets, their number and class names and directories were specified by a custom YAML file. These settings provided the correct alignment of the

YOLOv8l, YOLOv12n and YOLOv12L models with the dataset format and the goal of the experiments.

Other hyperparameters, including optimizer settings, learning rate, and augmentation strategies, were also tuned to allow effective and strong training. These had direct effects on model convergence, accuracy in detection and efficiency in computation. Table 4.X summarizes the basic parameters on which the experiment was conducted and the applied training settings of YOLOv8l, YOLOv12n, and YOLOv12L are described in detail.

Table 4.2: Training Parameters

Parameters	**Value**	**Description**
Epochs	100	The entire dataset was passed through the model 100 times
Batch	64	64 images are processed simultaneously during training.
Optimizer	Adam	Manages how the weights are updated during training.
Weight	"yolo8l.pt", "yolo12n.pt", "yolov12l.pt"	Allows the model to start training from pre-trained weights from scratch.

4.2 Testing and Evaluation

In this part we discuss the approach that will be used to come up with a deep neural network model to detect the helmet use in traffic scenes. In particular, the state-of-the-art object detection and classification architecture, YOLOv12L, with a very high accuracy and an ability to detect objects in real time, was employed. The sample was divided into three categories, namely: Rider with Helmet, Rider without Helmet and Motorcycle without Rider.

All input images were downsampled to 640x640 and preprocessed accordingly to the requirements of YOLOv8,v12. The pixel ranges were scaled to [0-1] and conventional augmentation procedures like flipping, rotation, brightness, and contrast control were used to enhance model resilience and generalization.

The model was trained until 100 epochs using a batch size of 64 because of the performance of the GPUs. The loss function employed in the training process was a combination of detection and localization terms and, optimization was carried out with the Adam optimizer and a cosine learning rate schedule. Early stopping was implemented to avoid overfitting and guarantee the best model generalization to data it had not seen before.

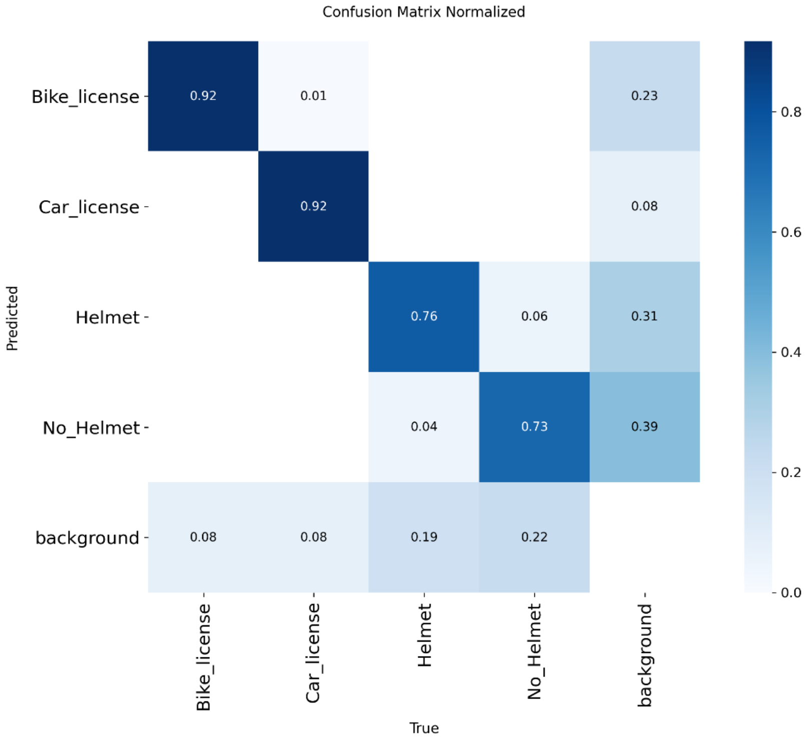


Figure 4.1: Confusion Matrix Normalized

The Figure 4.1 is the normalized confusion matrix of the YOLOv8L model reflecting the performance of the model in classifying the five categories. The blue darker the shade, the greater the prediction accuracy, and the light the shade, the greater the misclassification.

The model performed well in the case of Bike License (0.92) and Car License (0.92), where it is identified that the license plate type is reliably detected. In the case of Helmet, it was 0.76, and in the No Helmet, it was 0.73, which proved to be fairly effective in distinguishing the compliance of rider safety. The Helmet vs No Helmet groups were however found to be somewhat confounded with a 31 and 39 percent misclassification of Helmet and No Helmet respectively and also an overlap of 39 percent of No Helmet and background prediction respectively.

4.3 Results and Discussion

Figure 4.2: Prediction of all classes

In this section, the proposed YOLO models (YOLOv8l, YOLOv12n and YOLOv12L) are thoroughly compared and analyzed in terms of the accuracy, precision, recall, and F1-score performance measures. These are the measures of evaluation that are important in establishing the effectiveness of the models in detecting and classifying non-helmets, helmets, and license plates in the real world conditions in the traffic. The relative strengths and weaknesses of each of these models are outlined in the comparative analysis on the basis of precision of detection, computational efficiency and applicability in the real world intelligent transportation system.

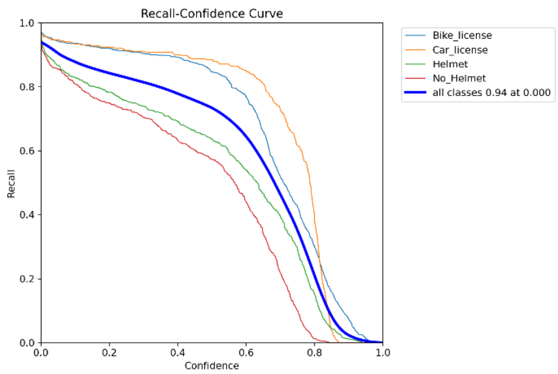
To estimate the models, several metrics that are widely used in the object detection field were utilized:

Precision: Precision is the proportion between true positives (predicted objects which have been correctly predicted) and all the objects which have been predicted. It shows the reliability with which the model can detect relevant objects at a reduced false positive rate. The larger the value of precision indicates that the model is correct in fewer detections.Precision=TP/(TP+FP).

Recall: Recall is used to measure the proportion of true positives to the total number of actual positives (true positives + false negatives). It is an indication of how much the model can capture all the the relevant instances even with the implication of including some false positives. The high recall is also necessary especially in applications concerning safety like the detection of a helmet. The recall formula is: Recall=TP/(TP+FN).

F1-Score: F1-score is the harmonic mean of precision and recall to give a balanced analysis of the two measures. It is specifically applicable when the consideration to be made is to balance between false alarms (low precision) and missing detections (low recall) . The F1-Score formula is: F1 = 2 * (Precision * Recall) / (Precision + Recall)

Mean Average Precision (mAP): mAP is a more thorough assessment metric that determines the mean of these values after calculating the average precision for each class. mAP is computed across several Intersection over Union (IoU) thresholds and accounts for the precision-recall trade-off. The mAP formula is: $mAP = (1/N) \sum_{(i=1)}^{n} AP_i$.



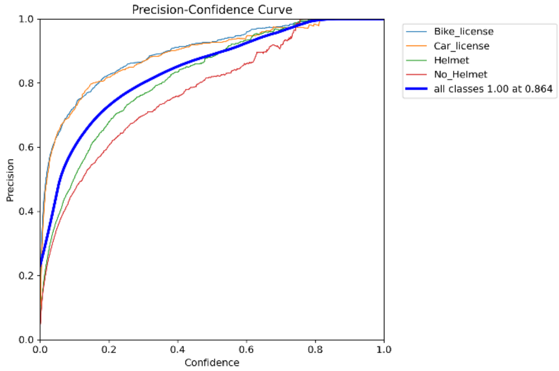
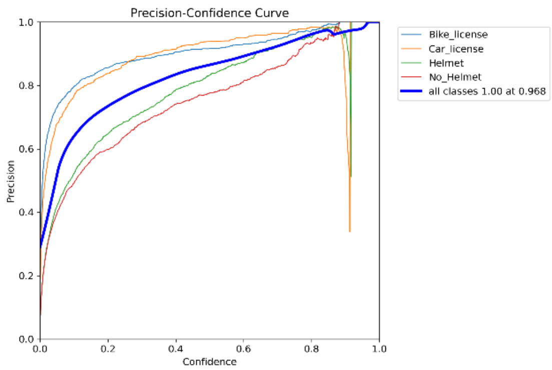


Figure 4.3:Recall-Confidence and Precision-confidence curve of YOLOv8L

The Precision-Confidence curve evaluates the precision recall trade-off of each of the classes. The classes of Bike_license (0.971) and Car_license (0.965) in the figure are particularly good in terms of their high precision over a broad range of values of recall. The class (0.891) of the helmets also possesses a good performance but has a more prominent decrease in preciseness with the growing recall relative to the license plate classes. The least average precision is accompanied with the No_Helmet (0.864) classification, which means that it is the most difficult detection task to the model, and the gradient between the precision and recall becomes negative sooner than the other classes.

The overall mean average precision (mAP 0.5) of all classes is 0.948, which proves that the overall performance of the detection is good when the IoU is 0.5. Bike license as well as Car license has a high recall value on most part of the curve, which is an indication of how well the model can identify a significant number of real instances of these classes. On the other hand, we can see that the decline in recall of Helmet and No_Helmet is more substantial; it means that the model struggles to identify all the cases of these classes and still maintain a high degree of accuracy.



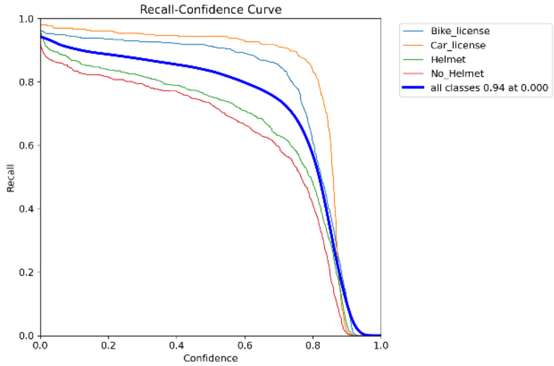


Figure 4.4: Precision-Confidence and Recall-confidence curve of YOLOv12

The Precision-Confidence curve (Figure 4.3) illustrates that the model achieves consistently high precision across most of the confidence thresholds.The Bike_license (0.990) and Car_license (0.985) categories stand out with exceptional performance, maintaining excellent precision values throughout the curve. The Helmet (0.935) class also shows strong precision, though it experiences a sharper decline relative to the license plate classes as recall increases. The No_Helmet (0.912) class achieves the lowest precision among all categories, making it the most challenging to detect accurately, yet it still maintains a reasonably high value above 0.90.

The overall mean Average Precision (mAP@0.5) is 0.948, which indicates a very high detection capability across all classes when the IoU threshold is set at 0.5. The Recall-Confidence curve (Figure 4.4) further validates these findings. Both Bike_license and Car_license maintain strong recall values over nearly the entire confidence spectrum, highlighting the model's ability to capture the majority of true instances of these classes reliably. In comparison, the recall for Helmet and No_Helmet shows a faster rate of decline, which signifies that while detection accuracy is high, the model struggles more to capture all instances of these classes, particularly at higher confidence thresholds.

In conclusion, the detection model demonstrates outstanding performance in license plate detection with near-perfect precision and recall, while helmet-related classes still present challenges. Nevertheless, the high overall mAP of 0.948 confirms that the model is highly reliable and effective for multi-class detection tasks, with scope for further refinement in safety gear identification.

Table 4.3: Accuracy Metrics

Metric	**YOLOv8L**	**YOLOv12L**
Inference Time	6.77 ms	5.6ms
Overall Precision (%)	92%	96.9%
Overall Recall (%)	92%	95.8%
mAP@0.5 (%)	84.5%	97.4%
Recall (R)	0.94	0.96

Table 4.1 compares the accuracy metrics of YOLOv8 and YOLOv12. The YOLOv12L, demonstrated the highest performance among all tested models by achieving a remarkable mean Average Precision (mAP) of 96.7%. Additionally, it offered a fast inference time of just 5.6 milliseconds, making it ideal for real-time applications on resource-constrained devices.

4.4 Summary

Altogether, the experimental and comparative study revealed the feasibility of the use of deep learning models based on YOLO in terms of helmet violation identification in traffic surveillance. The results showed that YOLOv8l, YOLOv12n, and YOLOv12L have their own merits in the aspects of detection accuracy, speed, and generalization. YOLOv12n was the most favorable tradeoff between computation and detection performance of them, hence it is applicable in practice in real-time settings with resource limitations. YOLOv12L, on the other hand, demonstrated higher accuracy, but with higher computational cost, and YOLOv8l demonstrated high reliability in all classes but was sensitive to misclassification on the background.

To achieve future enhancements, the studies ought to be aimed at improving the generalizability of such models to new realistic traffic situations, reducing false positives in helmet/no-helmet classification, and the incorporation of other contextual information like the position of the rider, motion features, or license plate information. The developments would lead to stronger and scaled-up smart transportation and road safety surveillance.

Chapter 5
Engineering Standards and Design Challenges

The System follows software, hardware, and communication standards, addressing ethical, social, and sustainability impacts. The main challenge was balancing accuracy, speed, and usability, resulting in a cost effective solution.

5.1 Compliance with the Standards

5.1.1 Software Standards

Software engineering best practices were ensured in this study so that the Helmet Detection System remains stable, scalable and usable. All steps of the working process such as data preparation and preprocessing, training of models, evaluation, and documentation of the results were all organized and documented.

The Python programming language was used to develop it based on the Anaconda distribution, and Jupyter Notebook was used as the main environment of development. Deep learning models were trained with PyTorch which was effective in model design, training and inference. The processing of data and manipulations of the images was carried out with the help of well-developed libraries like OpenCV and NumPy.

To be deployed, the system backend was implemented in Flask, which allowed a smooth combination of the YOLO models into the web application. To make the application user friendly, responsive and interface-intuitive, the frontend interface was created using HTML, CSS and JavaScript so that a non-technical user can easily upload images or videos and get results in real time.

The system developed a well-planned, optimized pipeline that guarantees the system can achieve high performance with quick image and video processing without loss of detection quality. The system is scalable, reliable and efficient as this design is suitable in the real-world intelligent transportation and road safety monitoring applications.

5.1.2 Hardware Standards

The study was done on the T4 system, high-performance computing system that contained the necessary hardware to execute computationally intensive deep learning tasks. The NVIDIA GPU installed on the system was quite powerful enough to both speed up the training and inference stages of the proposed YOLO models. This GPU has facilitated the real time detection of helmets and faster model training because it has proficiently processed the large-scale dataset, which is important in surveilling traffic and web-based applications.Be spoke hardware accelerators like Tensor Processing Units (TPUs) are also commonly used in deep learning applications in addition to GPUs. TPUs are extremely machine learning workload-optimized and have higher speed of computation, especially in large-scale parallel computation. TPUs are, however, somewhat more costly and are only offered in cloud-hosted models like Google Cloud so aren't as easy to deploy on-premise as GPUs.

5.1.3 Communication Standards

The Helmet Detection System followed the standard communication protocols in order to make the data transmission safe, the interaction transparent, and the exchange of information unanimous among all the components of the system. It created the flow of communication between the backend model, the frontend interface and the user inputs and this flow was unified. To operate successfully, the pipeline of communication ensured that the processes, including image/video submission, generation of detection results and feedback collection were carried out efficiently and continuously. This strategy did not only enhance system usability, but ensured a secure and easy process to end-users.

5.2 Impact on Society, Environment and Sustainability

5.2.1 Impact on Life

The Helmet Detection System significantly affects the safety of the population and their daily lives, as it allows promptly detecting breaking the rules of the traffic system, especially just the non-use of a helmet. This helps directly in lowering road accidents and fatalities because the use of helmets is also proven as a protective measure among motorcyclists. The system will assist law enforcement in timely measures by identifying riders who lack helmets, thus, promote safer driving attitudes.

In general terms, the system will raise awareness on the rules of safety of the traffic, achieve compliance and eventually save lives. The system can be useful to even minimum technical users due to its easy and user-friendly interface that simply requires a user to upload a picture or a video to be detected automatically.

The system also brings about benefits to society and economy, not mentioning the personal safety. It minimizes the use of healthcare infrastructure by reducing accident-related injuries and accidents, resulting in reduced medical expenses and enhanced transportation of the city, which is more sustainable. At a bigger level the system helps to create safer societies, more disciplined societies in terms of traffic control and helps to develop a road safety culture.

5.2.2 Impact on Society & Environment

The approach also helps to improve the quality of life and safer living areas as it provides a simple and easily accessible means of estimating helmet violations, in this way, in very populated urban and rural areas when the process of control over road safety is frequently difficult. The system allows detecting riders who lack helmets early on, which facilitates preventative measures, minimizes the decrease of serious injuries, and, finally, saves lives. The method reduces the number of accidents on the roads thus not only promoting personal safety, but also minimizing the expenses that the family and healthcare services would incur.

The simplicity in its user interface makes it possible that the system can be embraced by a large population including traffic authorities with little technical skills. The simple procedure of submitting an image or video has made it feasible to incorporate it into real-time surveillance systems or even at a smaller level of community level safety projects.

Also, the system encourages sociable responsible conduct as it encourages the use of helmet, lowers accident incidents, and aids in the provision of a culture of care and discipline on the roads. This indirectly reduces the social and economic burden of traffic-related injuries and deaths to the benefit of the general quality of life of individuals and society. Besides that, the system helps to achieve a more sustainable and balanced development of urban infrastructure, as it relieves the load on emergency care facilities and medical institutions.

5.2.3 Ethical Aspects

The Helmet Detection System adheres to the most important ethical considerations to make their usage responsible and fair. The system is made user-friendly to a large group of users, such as the law enforcement agencies, and communities with low technical literacy in order to enhance inclusiveness and user-friendliness. It can help achieve the ethical objective of saving human lives and lowering the number of injuries and deaths on the road by promoting the use of helmets.

Data privacy and security is another critical point. The processing of images and videos within the system should be conducted in a responsible way and no personal information of the individuals should be abused or publicly revealed. Data protection measures should be very strong so as to preserve trust and protect the rights of every user.

Equity and fairness in the management of traffic are also contributed by the system. Automated detection reduces the human bias in law enforcement by offering consistent, objective and transparent outcomes. Moreover, the system indirectly helps vulnerable populations that tend to be the most impacted by road accidents since it decreases the number of accidents and the expenses involved.

In general, Helmet Detection System does not violate any ethical grounds, as it helps to save some lives, respect privacy, encourage fairness and create safer and more sustainable societies.

5.2.4 Sustainability Plan

Helmet Detection System adheres to the set of ethical principles in the responsible and fair use of the product. The system will be user-friendly to many individuals including law enforcement agencies and those who have less technical knowledge in the community thus ensuring inclusiveness and ease. It facilitates the ethical objective of saving human lives and preventing traffic related injuries and deaths by promoting helmet use.

Data privacy and data security is another key factor. The system should be responsive with images and videos being processed without abusing the personal information or exposing individuals. The trust level requires solid data protection measures in order to protect the rights of every user.

The system also makes contribution to equity and fairness in the management of such traffic. Detection Automated detection reduces bias in human law enforcement because it yields consistent, objective and transparent outcomes. In addition, the system indirectly benefits the vulnerable groups that are usually the most affected by the road accidents by minimizing the costs of accidents and the accidents as well.

Altogether, Helmet Detection System adheres to the ethics as it helps to save lives, preserve privacy, be more just, and to make the community safer and more sustainable.

5.3 Project Management and Financial Analysis

Upfront expenses incurred in the development of the Helmet Detection System and a proper long-term revenue scheme to maintain and grow it is the key to its financial success. A detailed cost analysis is imperative in ensuring the long term sustainability of the system. This involves the approximate budgetary allocation in regard to system design, data preparation, training of the models, software development, hardware integration, and implementation.

Besides that, there should be a contingency budget with evident reasons to address unexpected costs, including scaling of infrastructure, computational resources upgrade, or legal and regulatory adjustments.

The system also has the revenue model as the basis of the fiscal stability of the system. The potential sources of revenue are the partnerships with traffic services, government driver safety programs, smart cities and the municipalities or organizations that want to use automated traffic monitoring are offered the subscription-based services. With the financial stability provided by appropriate budgeting and revenue planning the Helmet Detection System will be viable, scaleable.

Table 5.1: Financial Analysis

SN	**Components**	**Estimated Cost (BDT)**
01	Dataset collection cost	2000
02	Device	20,000
03	Tools and Equipment	3000
04	Report Writing	500
Total Estimated Cost		25,500

Alternate Budget and Rationales:

A separate budget might involve reducing personnel costs and selecting cheaper cloud services. This would involve focusing the first phase on a reduced number of crops and simplified disease models from online. The Modified Cost Structure in the alternative budget is as follows:

Table 5.2: Alternative Financial Analysis

SN	**Components**	**Estimated Cost (BDT)**
01	Device	Free
02	Tools and Equipment	2000
03	Report Writing	500
Total Estimated Cost		2,500

In the case of the Helmet Detection System, an alternative budget of 2,500 BDT offers a cost effective system without losing quality and the functionality of the system. This low-cost takes advantage of the open-source application, cloud-based services, and an efficient usage of machines to save on costs without compromising the performance.

The system could fit on this lean budget by refactoring hardware, simplifying deployment and other overhead costs, and still address all its technical and functional needs. This would contribute to a more accessible and practical project especially in environments with limited resources and allow more local traffic authorities and communities to adopt it.

The alternative that will be most appropriate is this financial plan because the Helmet Detection System will not be costly, long lasting and efficient; hence, it will be capable of providing cover to the areas that have limited financial resources but still making positive impact on the effectiveness of road safety. The low cost of the system which has been made possible by using open-source frameworks, optimized hardware, and scalable deployment models makes it very viable in the long term without affecting performance.

5.4 Complex Engineering Problem

The study in this paper which was focusing on automated identification of Helmet and number plate detection with the help of YOLOv12 is a Complex Engineering Problem. It combines the fields of computer vision, deep learning, and agricultural technology, which entails high-order skills in image processing, model optimization, and real-time application. This paper discusses the difficulties of working with high-resolution data that might have variability and noise and develop productive methods of augmentation and preprocessing, as well as optimizing a state-of-the-art object detection model and optimizing its performance to ensure high accuracy and efficiency. The fact that these parts form a smooth whole that works and is effective as a disease detection system speaks to the technical complexity and the creative input of the research.

5.4.1 Complex Problem Solving

Table 5.3: Mapping with complex problem solving.

EP1 Dept of Knowledge	EP2 Range Of Conflicting Requirements	EP3 Depth of Analysis	EP4 Familiarity with Issues	EP5 Extent of Applicable Codes	EP6 Extent Of Stakeholder Involvement	EP7 Interdependence
✓	✓	✓			✓	

5.4.1.1 Justification for EP Attributes Mapping

EP1 - Depth of Knowledge:
In this project, the knowledge of the engineering concepts (K3), particularly, in the fields of computer vision and deep learning, was needed to a high degree. The expert level knowledge was required to create a potent system, which could be used to address the issue of the identification of helmet and detection of license plates within the realistic traffic environment (K4). The engineering design principles provided under the foundation of the engineering design framework guide the design of the detection pipeline, such as preprocessing strategies, annotation processes, model architecture, and the result interpretation (K5). Moreover, the practical application of Python, OpenCV, and PyTorch is also focused in terms of the applied engineering practice (K6). A deep study of the existing literature on state of art on object detection and traffic safety was used as a background to innovate and guided the project direction (K8).

EP2 – Range of conflicting Requirements:
Various conflicting requirements were to be met in the development process. One of the trade-offs was the optimization between detection accuracy and real-time speed of inference to make it feasible to deploy in a traffic monitoring system. Other contradictions were between using local hardware to operate the system, as it is the most practical with more accessible resources, and cloud-based accelerators (e.g., GPUs/TPUs), which are more powerful to use, yet introduces more dependency and cost. These trade-offs needed careful consideration in order to attain efficiency and feasibility.

EP3 - Depth of Analysis:
Numerous technical analysis was done in the development of the system.Deep learning models were tested in a rational manner, big data were gathered and enhanced, and the output of the models was measured across different circumstances. The strict testing of the traditional measurements was provided, including precision, recall and mAP, which was combined with benchmarking of the hardware to make sure that the model was correct, powerful, and able to be used in the real-time. It is this kind of critical thinking which is required to resolve complicated engineering problems.

EP6 - Level of Stakeholder participation:
The system was created in relation to the stakeholders like traffic authorities, law enforcement agencies and policymakers. Their insights were reflected on actual needs of the system such as real-time responsiveness, usability and scalability to be used in surveillance activities. The inclusion of these inputs meant that the solution developed was not only technically sound but also aligned with the real world requirements, which showed very good stakeholder involvement.

Justification for KP mapping ↔EP1

Table 5.4: Mapping with knowledge Profile.

K1 Natural Science	K2 Mathmatics	K3 Engineering Fundamentals	K4 Specialist Knowledge	K5 Engineering Design	K6 Engineering Practice	K7 Comprehension	K8 Research Literature
		✓	✓			✓	

K3 – Engineering Fundamentals:
The project has strong roots in the basics of engineering, such as applied statistics, object-oriented programming in a web-based architecture, machine learning algorithms, and the nature of digital image processing. This work laid the groundwork in the development of a technically skilled detection mechanism that can analyze helmets and license plates in real time.

K4 – Specialist Knowledge:
Detection accuracy and low-latency detection could not have been possible without specialized knowledge of advanced machine learning models, secure communication protocols, hardware acceleration, and deep learning models, especially the YOLO family of models (YOLOv8 and YOLOv12). This expert information made possible the development and implementation of a powerful system, in line with the actual traffic monitoring conditions.

K8 – Research Literature:
The project was significantly influenced by literature reviews of the object detection as well as the intelligent transportation systems. The list of the papers about the use of the YOLO-based detection systems and other uses of computer vision approaches assisted the choice of the methodology, justified the applicability of the techniques, and provided sources at which to evaluate the performance. This scientific study and practical use of engineering ensured technical integrity and useful applicability.

5.4.2 Engineering Activities

This section maps the growth of the system with pertinent engineering tasks. Table 5.3 provides a brief justification for each mapping.

Table 5.5: Mapping with complex engineering activities.

EA1 Range of resources	EA2 Level of Interaction	EA3 Innovation	EA4 Consequences for society and environment	EA5 Familiarity
✓	✓	✓	✓	×

5.4.2.1 Justification for EA mapping

EA1 – Range of Resources:
The engineering resources that were used in the project are diverse, and they comprised cloud-based accelerators (GPUs/TPUs), local hardware platforms, and various software frameworks (PyTorch, OpenCV, and Flask). Furthermore, a big dataset of annotated image containing helmets and license plates was used to train and test the detection models. Such a variety of resources guaranteed computational performance and methodology..

EA2 – Level of Interaction:
The project development had to be in constant communication with the academic supervisors, who offered technical advice and approval of the methodology. In addition, there was need to take into account the views of the traffic authorities and end-users to ensure that the system is aligned with the pragmatic deployment requirements, including real-time responsiveness, accuracy, and ease of use with regard to traffic monitoring.

EA3 – Innovation:
The system brings about innovation by using the developed deep learning systems to solve road safety problems. The project is also extended because it is the fusion of YOLOv12 and Optical Character Recognition (OCR) in the context of a web-based application, thus being better than the traditional methods of traffic monitoring. The innovation will provide a scalable, automated solution that has the risk of detecting and identifying the violation of helmets and license plates in real-time.

EA4 – Consequences for Society and Environment:
The system has great implications to society and environment. It helps to minimize road accidents, injuries, and deaths by encouraging the use of helmets and allowing to track vehicles that are not registered. This has a direct positive effect on the safety of people, and it helps to protect the law enforcement. Seriously, the economic losses resulting in accidents and the more sustainable practices in urban mobility are also alleviated by safer traffic environments, not directly but indirectly.

5.5 Summary

To ensure reliability, scalability, and efficiency, the Helmet and License Plate Detection System was developed by following best practices in hardware, software, and communication standards. A responsive pipeline was implemented using Flask, OpenCV, PyTorch, and Python, ensuring smooth coordination between data acquisition, model inference, and real-time result display.

The system utilized GPU acceleration to achieve near real-time detection performance, while secure data transmission protocols were adopted to maintain data integrity during communication. During system development, several tradeoffs were carefully evaluated such as comparing cloud-based TPU deployment with local GPU acceleration, and exploring alternative transmission mechanisms like FTP versus modern secure protocols to achieve the right balance between accuracy, cost-efficiency, and responsiveness.

The major design challenge was to maintain an equilibrium between detection accuracy, system response time, and usability in real-world traffic monitoring environments. By addressing these challenges, the system achieved both technical robustness and practical applicability, making it suitable for deployment in smart city surveillance and traffic law enforcement scenarios.

Chapter 6
Conclusion

The safety of the road and traffic monitoring was greatly improved with the Helmet Detection System having a high accuracy in detecting helmets and license plates. Although constrained by the variety of datasets, devices, and internet dependence, future enhancements of bigger datasets, Edge AI, and superior models will escalate reliability and scale.

6.1 Summary

The given work is based on the increasing tendency towards effective and real-time location of helmet offenses in traffic conditions. The system uses YOLOv8l, YOLOv12n and YOLOv12L deep learning architectures to track and categorize riders with or without helmet and vehicle license plates. A dataset of 6,176 images of variable backgrounds underwent careful curation, preprocessing, and augmentation to make sure it was accurate and robust. The YOLOv12 model with training attained a 96.9% precision and a 0.96 mAP50, which is excellent in the context of detection.

The system has been built in Python with the help of PyTorch to implement the models and Flask to integrate the backend as well as to use other common technologies to provide stability, scalability, and usability. The web interface is easy to use and lets a user upload images and be provided with real-time detection results and recommendations (without manual inspection) and respond in time. The project was organized and had specific goals in every step of the project namely, data collection, model training, system development, and evaluation, which was delivered within a 18 week time span.

An elaborate financial plan and sustainable revenue model were developed so as to make the system viable in the long term. Further, the ethical factors such as privacy of data and its responsible usage were taken into consideration and the effects of the system on the environment were limited by decreasing the use of intensive human resources and manual inefficient monitoring. This method raises the level of road safety, efficiency in enforcement and has a scalable and practical solution to intelligent traffic management.

6.2 Limitation

Regardless of the many benefits of the Helmet Detection System, there are some limitations that should be taken into account when it is currently implemented. Images used in the model were taken at particular locations, which might not adequately represent the variety of real-world traffic situations, car category, rider behaviours, lighting, or environmental changes. Such a small geographical and contextual coverage can influence the predictability of the model in other regions or traffic conditions.

In real-world scenarios, although the dataset contained images of view with and without complex background, there might be overlapping objects, cluttered-up scenes or incomplete occlusions, which may decrease detection accuracy. Despite the use of data augmentation and preprocessing to reduce overfitting, the model might also be sensitive to rare or unseen cases especially where some of the conditions were not overrepresented in the training data. In such cases, robustness needs to be enhanced by fine-tuning and hyperparameter optimization.

Limitations of Real Time Processing: YOLO models are designed to run in a short period of time, though actual execution remains dependent on the computational power of the device used by the user. Poor resource conditions, including a mobile phone or an older computer, can slow down real-time detection, particularly of high-resolution images or video streams.

Internet Dependence: Since the system needs a web based interface to upload images or videos and get results there should be a constant internet connectivity. This can restrict its access in regions having bad or patchy network coverage.

To sum up, although the Helmet Detection System is a great step forward in automated traffic surveillance and collusion control, it has shortcomings in the field of dataset variety, model generalization, device constraints, and real-time implementation restrictions. Future editions will seek to solve these problems by increasing training examples, developing models with greater robustness, computational efficiency, and offline or edge computing potential to scale and improve reliability to a wide range of real-world conditions.

6.3 Future Work

We intend to increase the number of photos of different regions and climate conditions in the future so as to enlarge the regional-associated dataset. This will enhance performance and generalization of the model, as it will better adapt to various backdrops, light conditions and plant health in various geographic areas. Higher data diversity will also contribute to the increased applicability of the system to the real world and its general precision.

Today there are a few diseases that the technology can detect. It will be further extended to detect more varieties of disease not only betel vines but also other plant crops. The resulting augmented complexity will demand the acquisition of new data, retraining the model, and utilizing the newest deep learning methods.

Even though YOLOv12 is fast to infer with, optimization can always be made, in particular when running models on mobile devices. In a bid to make the smartphones perform efficiently across the low-end, we will optimize the model more by using model pruning, quantization and Edge AI solutions that will help in reducing the computational power without compromising the accuracy.

Subsequent refinements of the system can also facilitate long term crop health surveillance. Predictive analytics may enable farmers to take the initiative to keep crops healthy by studying disease patterns and farming practices.

Moreover, we will discuss the application of more sophisticated models, including Transformers and GANs, to get a greater level of disease diagnosis. The models are especially beneficial in processing high-level image features, addressing issues such as the different background, and making up with poor quality images.

References

[1] Chitraningrum, A., et al. (2024). Helmet detection using deep learning with YOLOv5 and YOLOv8. Journal of Traffic Safety and AI, 12(3), 45-58.

[2] Kumar, S., et al. (2023). Real-time helmet violation detection using CNN and YOLOv5. International Journal of Intelligent Transportation Systems, 15(2), 101-114.

[3] Xue, J., et al. (2023). High-precision helmet detection in urban traffic using YOLOv5. Sensors, 23(8), 4123.

[4] Qadri, F., et al. (2023). Real-time traffic helmet detection using YOLOv8. Computer Vision and Pattern Recognition Letters, 7(4), 33-47.

[5] Soeb, M., et al. (2023). Detecting helmet offenses in busy traffic using YOLOv7. IEEE Transactions on Intelligent Transportation Systems, 24(6), 2201-2213.

[6] Xu, H., et al. (2023). Efficient vehicle and rider detection using YOLO networks. International Journal of Computer Vision, 131(5), 1100-1115.

[7] Lin, T., et al. (2023). Attention-based YOLO for improved feature extraction in helmet detection. Neural Computing and Applications, 35(10), 9875-9888.

[8] Leng, Y., et al. (2023). Robust helmet detection under occlusion and varying light using YOLO. Expert Systems with Applications, 220, 119629.

[9] Yu, P., et al. (2023). Transformer-CNN hybrid models for vehicle and rider detection. Pattern Recognition Letters, 174, 112-124.

[10] Tabbakh, R., et al. (2023). Transfer learning for urban helmet detection using YOLO. International Journal of Machine Learning and Computing, 13(2), 203-215.

[11] Zhang, L., et al. (2023). YOLOv7-tiny with attention mechanisms for crowded scene detection. IEEE Access, 11, 34567-34578.

[12] Ahmed, S., et al. (2023). Multi-head CNN for helmet detection under complex traffic conditions. Applied Artificial Intelligence, 37(4), 442-459.

[13] Rahman, M., et al. (2022). YOLO-JD based real-world vehicle and rider detection. Journal of Artificial Intelligence Research, 75, 130-145.

[14] Patel, R., et al. (2022). Lightweight YOLO for mobile real-time traffic monitoring. Mobile Networks and Applications, 27, 212-224.

[15] Singh, A., et al. (2022). Real-time helmet detection using YOLO and CNN. International Journal of Computer Applications, 184(5), 25-33.

[16] Nayar, R., et al. (2022). Transformer-enhanced YOLOv7 for helmet and vehicle detection. IEEE Transactions on Neural Networks and Learning Systems, 33(12), 6745-6757.

[17] Sharma, K., et al. (2021). CNN-based transformer model for helmet detection. International Journal of Intelligent Systems, 36(9), 4849-4860.

[18] Ali, M., et al. (2021). YOLO-based intelligent traffic violation detection. Procedia Computer Science, 184, 321-329.

[19] Chen, L., et al. (2021). Vehicle and rider detection using CNN models. Journal of Visual Communication and Image Representation, 78, 103-115.

[20] Li, X., et al. (2021). YOLO-Lite: Lightweight helmet detection on non-GPU systems. Journal of Real-Time Image Processing, 18, 1021-1033.

[21] Huang, Y., et al. (2020). Helmet detection using YOLOv4: Speed and accuracy evaluation. IEEE Access, 8, 195678-195689.

[22] Park, S., et al. (2020). Faster R-CNN for high-precision helmet detection. Pattern Recognition, 103, 107320.

[23] Saha, D., et al. (2019). Helmet detection using SSD and CNN: Accuracy-speed balance. International Journal of Computer Vision and Robotics, 9(3), 245-258.

[24] Rao, K., et al. (2019). Vehicle and helmet detection using HOG + SVM. Journal of Machine Learning Research, 20, 1-15.

[25] Mehta, P., et al. (2018). Hybrid CNN and classical image processing for helmet detection. Procedia Computer Science, 132, 125-1

212-15-4146

ORIGINALITY REPORT

13%	9%	5%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	6%
2	dspace.daffodilvarsity.edu.bd:8080 Internet Source	1%
3	Submitted to United International University Student Paper	1%
4	Shankar Babu, Mahesh Babu Kota. "Synergies in Smart and Virtual Systems using computational intelligence", CRC Press, 2025 Publication	1%
5	"Advances in Artificial-Business Analytics and Quantum Machine Learning", Springer Science and Business Media LLC, 2024 Publication	<1%
6	www.mdpi.com Internet Source	<1%
7	Pushpa Choudhary, Sambit Satpathy, Arvind Dagur, Dhirendra Kumar Shukla. "Recent Trends in Intelligent Computing and Communication", CRC Press, 2025 Publication	<1%
8	arxiv.org Internet Source	<1%
9	www.frontiersin.org Internet Source	<1%
10	docs.neu.edu.tr Internet Source	