

Bangladeshi Vehicle Number Plate Detection: A Deep Learning Approach with YOLO Models

By
Md. Khaled Hasan
212-15-14691

FINAL YEAR DESIGN PROJECT REPORT

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

Supervised by
Md. Sazzadur Ahamed
Assistant Professor
Department of Computer Science and
Engineering Daffodil International
University

Co-Supervised by
Md Shohidul Islam
Polash
Lecturer
Department of Computer Science and
Engineering Daffodil International
University



**DAFFODIL INTERNATIONAL
UNIVERSITY**
Dhaka, Bangladesh

May 14, 2025

APPROVAL

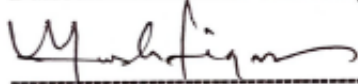
This Project titled “**Bangladeshi Vehicle Number Plate Detection: A Deep Learning Approach with Yolo Models**”, submitted by Md. Khaled Hasan, ID No: **212-15-14691** to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial 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 **14 May, 2025**.

BOARD OF EXAMINERS



Dr. S. M Aminul Haque
Professor and Associate Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Chairman



Mohammad Jahangir Alam
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Masum 14/5/25

Mr. Md Mohammad Masum Bakaul
Senior Lecturer
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Arshad
14.05.25

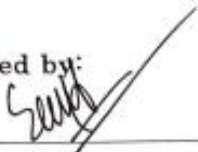
Dr. Md. Arshad Ali
Professor
Department of Computer Science and Engineering
Hajee Mohammad Danesh Science & Technology
University

External Examiner

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Md. Sazzadur Ahamed, Assistant Professor**, Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:



Md. Sazzadur Ahamed

Assistant Professor

Department of Computer Science and Engineering

Daffodil International University

Co-Supervised by:

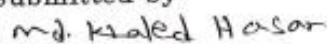
Md Shohidul Islam Polash

Lecturer

Department of Computer Science and Engineering

Daffodil International University

Submitted by:



Md. Khaled Hasan

Student ID: 212-15-14691

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 **Md. Sazzadur Ahamed, Assistant Professor**, Department of Computer Science and Engineering, Daffodil International University, Dhaka, Bangladesh. Deep knowledge and keen interest of our supervisor in the field of **Computer Vision** 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 them at 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

This work presented here compares the performance of various YOLO models, including YOLOv8m, YOLOv12m, YOLOv11m, YOLOv9m, and YOLOv10m. The models are compared in the context of Bangladeshi vehicle number plate detection based on the leading metrics of precision, recall, and mean Average Precision (mAP) at various Intersection over Union (IoU) thresholds. The study conducts using hyper-parameter tuning YOLO models. These experiments show that YOLOv8m is the best with the maximum precision (0.959) and recall (0.922) with mAP@50 as 0.946, and mAP@50-95 as 0.55. This shows the effectiveness of YOLOv8m in detecting vehicle number plates in varied urban environments. Though YOLOv8m is better performing, YOLOv12m and other models like YOLOv11m and YOLOv9m also perform competitively and hence can be used in real-time vehicle number plate detection systems. YOLOv8m's better overall performance highlights its viability in traffic management system and surveillance applications, making it a desirable choice for Bangladeshi vehicle number plate detection software. This work enlightens the application of YOLO models in using them for the task of vehicle detection and extends the limits in urban object detection through deep learning.

Table of Contents

Approval	i
Declaration	ii
Acknowledgements	iii
Abstract	iv
List of Figures	vii
List of Tables	viii
1 Introduction	1
1.1 Introduction.....	1
1.2 Motivation	3
1.3 Objectives	4
1.4 Methodology	6
1.5 Project Outcome.....	7
1.6 Organization of the Report	8
2 Background	11
2.1 Introduction.....	11
2.2 Literature Review	12
2.2.1 Similar Applications	16
2.2.2 Related Research.....	20
2.3 Gap Analysis	23
2.4 Summary	25
3 Research Methodology	26
3.1 Methodology/Requirement Analysis & Design Specification.....	26
3.1.1 Overview	26
3.1.2 Proposed Methodology/ System Design	27
3.1.3 Functional and Nonfunctional Requirements	28
3.1.4 Context Diagram	29
3.1.5 Data Flow Diagram Level 1.....	30
3.1.6 UI Design	31
3.2 Detailed Methodology and Design.....	32
Table of Contents	Table of Contents

3.3	Project Plan	43
3.4	Task Allocation.....	44
3.5	Summary	44
4	Implementation and Results	46
4.1	Environment Setup	46
4.2	Testing and Evaluation/Performance/ Comparative Analysis.....	47
4.3	Results and Discussion	48
4.4	Summary	54
5	Engineering Standards and Design Challenges	56
5.1	Compliance with the Standards.....	56
5.1.1	Software Standards.....	56
5.1.2	Hardware Standards	57
5.1.3	Communication Standards.....	58
5.2	Impact on Society, Environment and Sustainability	59
5.2.1	Impact on Life.....	59
5.2.2	Impact on Society & Environment.....	59
5.2.3	Ethical Aspects	60
5.2.4	Sustainability Plan.....	60
5.3	Project Management and Financial Analysis.....	61
5.4	Complex Engineering Problem.....	65
5.4.1	Complex Problem Solving.....	65
5.4.2	Engineering Activities	67
5.5	Summary	68
6	Conclusion and Future Work	69
6.1	Summary	69
6.2	Limitation	69
6.3	Future Work	70
	References	71

List of Figures

1.1	Outcome of Detection with YOLO Model	7
3.1	Research Workflow for Number Plate Detection using Yolo Models.....	27
3.2	Context Diagram of the system	30
3.3	Data Flow Diagram of the System	31
3.4	UI Design of Number Plate Detection System.....	32
3.5	Dataset Distribution Chart.....	35
3.6	Sample Images from Dataset.....	36
3.7	Architecture of YOLO v8	38
3.8	Architecture of YOLO v9	39
3.9	Architecture of YOLO v10	40
3.10	Architecture of YOLO v11	40
3.11	Architecture of YOLO v12	41
4.1	Labels Correlogram for the Dataset.....	48
4.2	Training Loss, Validation Loss, mAP metrics of performed models.....	50
4.3	Precision-Confidence Curve for Training from the performed models	52
4.4	Recall-Confidence Curve of YOLO v8m.....	52
4.5	Confusion Metrics of YOLO v8m	53
4.6	Validation result on sample data of YOLO v8m	54

List of Tables

2.1 Research Matrix for Number Plate Detection.....	15
2.2 Similar Applications in Number Plate Detection.....	18
2.3 Related Research in Number Plate Detection.....	22
2.4 Gap Analysis Summary Table	24
3.1 Specification of the Data Collection Devices	34
3.2 GANNT Chart of Project Timeline	45
4.1 Evaluation Metrics Description	47
4.2 Results of Trained Models.....	48
5.1 Project Timeline	62
5.2 Actual Research Budget (Self-Supported, Research-Based)	63
5.3 Alternative Budget (Enterprise-Scale Deployment)	64
5.4 Mapping with Complex Problem Solving	65
5.5 Mapping with Knowledge Profile.....	66
5.6 Mapping with Complex Engineering Activities.....	67

Chapter 1

Introduction

This chapter provides an overview of the research by introducing the background and significance of the study. It outlines the key motivation for addressing vehicle number plate detection using deep learning, followed by clearly defined objectives. The adopted methodology is briefly described, covering data collection, model development, explainability integration, and deployment. The chapter also presents the expected outcomes of the project and concludes with an outline of the overall structure of the report.

1.1 Introduction

Vehicle number plate recognition (VNPR) is among the key aspects of smart transport systems (ITS) in which sophisticated machine learning and computer vision technology assists in streamlining vehicle identification processes. Automated number plate detection needs to be a part of today's day-to-day urban traffic flow management, and automatic number plate identification plays a vital role to spearhead road safety enhancements, enable enforcement of traffic and crime legislation, and track automobiles. With increasing numbers of cars crowding the world, and Bangladesh among them, effective and trustworthy detection systems are required to manage traffic, reduce crime, and provide urban security [1], [2].

The advances in computer vision and deep learning over the recent past have transformed the vehicle number plate detection method rather dramatically. Traditional methods such as feature-based techniques, morphological processing, and edge detection have been largely displaced by convolutional neural networks (CNNs) for their greater accuracy and robustness [3]. Among all deep learning models, YOLO object detection networks are extremely popular now since they are able to perform real-time object detection, fast, and extremely accurate. Of YOLO models, most prominently YOLOv8 and YOLOv12, have reported excellent performance to identify various categories of objects such as vehicle number plates in practical applications [4], [5]. The application of YOLO-based models for number plate recognition of vehicles is particularly relevant in cities such as Dhaka, Bangladesh, where traffic congestion and crimes against vehicles are heavy issues [8]. As the number of vehicles on the roads keeps growing, accurate and automatic license plate detection and recognition systems are needed to enhance law enforcement and traffic management. However, such

Chapter 1

Introduction

This chapter provides an overview of the research by introducing the background and significance of the study. It outlines the key motivation for addressing vehicle number plate detection using deep learning, followed by clearly defined objectives. The adopted methodology is briefly described, covering data collection, model development, explainability integration, and deployment. The chapter also presents the expected outcomes of the project and concludes with an outline of the overall structure of the report.

1.1 Introduction

Vehicle number plate recognition (VNPR) is among the key aspects of smart transport systems (ITS) in which sophisticated machine learning and computer vision technology assists in streamlining vehicle identification processes. Automated number plate detection needs to be a part of today's day-to-day urban traffic flow management, and automatic number plate identification plays a vital role to spearhead road safety enhancements, enable enforcement of traffic and crime legislation, and track automobiles. With increasing numbers of cars crowding the world, and Bangladesh among them, effective and trustworthy detection systems are required to manage traffic, reduce crime, and provide urban security [1], [2]. The advances in computer vision and deep learning over the recent past have transformed the vehicle number plate detection method rather dramatically. Traditional methods such as feature-based techniques, morphological processing, and edge detection have been largely displaced by convolutional neural networks (CNNs) for their greater accuracy and robustness [3]. Among all deep learning models, YOLO object detection networks are extremely popular now since they are able to perform real-time object detection, fast, and extremely accurate. Of YOLO models, most prominently YOLOv8 and YOLOv12, have reported excellent performance to identify various categories of objects such as vehicle number plates in practical applications [4], [5]. The application of YOLO-based models for number plate recognition of vehicles is particularly relevant in cities such as Dhaka, Bangladesh, where traffic congestion and crimes against vehicles are heavy issues [8]. As the number of vehicles on the roads keeps growing, accurate and automatic license plate detection and recognition systems are needed to enhance law enforcement and traffic management. However, such

systems are facing challenges such as variations in plate formats of vehicles, illumination, weather, and the presence of occlusions, which affect detection performance [6].

In recent years, a few advanced techniques have been explored for number plate detection on vehicles. The integration of deep learning architectures such as YOLOv3, YOLOv4, and more recently YOLOv8 with optical character recognition (OCR) has seen significant improvement. These techniques typically have a two-stage approach: one, detecting the number plate from an image, and then character recognition using CNN-based methods [7]. YOLO's high-accuracy real-time object detection has made it a suitable model for this use case, especially in changing environments like urban roads. The uniqueness of the vehicle number plates in Bangladesh, where they also include Bengali characters along with numerals, makes detection and recognition even tougher. While most studies focus on plate forms that are popular in Western countries, the presence of Bengali script on number plates of cars necessitates the development of specially designed models that accommodate regional scripts [9].

Despite advancements in tea leaf disease detection using CNN models, several research gaps persist, limiting the reliability and broader applicability of existing solutions.

- **Environmental Variability:** Vehicle images in urban settings are primarily marred with poor lighting, motion blur, and occlusion, which compromise the accuracy of plate detection and character recognition [10].
- **Variety of Number Plate Types:** Bangladesh uses both Bengali and Arabic numerals, introducing additional complexity in segmentation of characters and recognition.
- **Real-time Processing:** Real-time vehicle detection is required in some applications like toll collection, traffic monitoring, and security. High speed processing with good detection accuracy is a big challenge [12].
- **Availability of the dataset:** Lack of a large, annotated dataset spanning multiple plate styles, lighting conditions, and various vehicles hinders training stable models.

The motivation for this research stems from the need to address these challenges and create an efficient, reliable vehicle number plate detection system tailored for Bangladesh. This study aims to take advantage of YOLO model strengths, in this instance, YOLOv8, in detecting car number plates in real-time with great accuracy, and overcoming environmental condition complexity as well as plate variation complexity. Through focusing on the optimization of YOLO models for Bangladesh's unique vehicle plate characteristics, this research aims to contribute to intelligent transport systems with the ability to improve traffic

safety, law enforcement, and urban mobility [13].

The aim of this study is to evaluate the performance of five different deep learning models employed for vehicle number plate detection and to make sure that they are employed to obtain the highest accuracy and efficiency in actual real-time urban traffic environments. The prime aim is to evaluate the performance of each model in identifying vehicle number plates under various conditions like different lighting, occlusions, and different kinds of plates used in Bangladesh. The models being compared are the variants of YOLO (You Only Look Once) family, i.e., YOLOv4, YOLOv5, YOLOv8, and YOLOv12, and an additional one for comparison for a better contrast [14], [15], [16]. All these models will be tested and trained over a self-made dataset with vehicle images of varied number plate structures, captured in different conditions to simulate real Bangladesh traffic. To make sure that the models are up to the standards of real-time detection, the performance of every model will be evaluated on video streams. This will mimic a real traffic monitoring scenario, where the model has to detect number plates of cars moving through a camera frame. FPS will be the performance indicator, and models that fail to achieve the minimum benchmark FPS for real-time processing will be eliminated from further analysis [17], [18]. After the models are evaluated against the above parameters, a comparative analysis will be made to determine the model that has the optimal balance between accuracy and speed in recognizing number plates of vehicles in the Bangladeshi context. The results will present valuable insights into the optimal model to implement in Bangladesh's smart city and traffic management system, where number plate detection in real time is paramount for efficient law enforcement and transport management [19], [20].

1.2 Motivation

The motivation to carry out this study lies in the intersection of three important needs in both the social and technological domains: improving urban traffic control, encouraging deep learning model applicability in real-time systems, and ensuring transparency within AI systems. Traffic jams and auto-related offenses in urban areas, particularly in Bangladesh, pose monumental public security, safety, and transportation effectiveness challenges from a socio-economic perspective. With the continuously growing the number of cars on the roads of cities like Dhaka, there is a highly need for automated systems that can provide accurate and efficient recognition of car number plates to aid law enforcement, make traffic monitoring easier, and enable hassle-free services including toll plaza collection and parking management. These systems' effectiveness has a direct impact on economic activities, public safety, and the efficiency of urban infrastructure overall [21], [22], [23].

Technically, while such sophisticated deep learning models like YOLO (You Only Look Once) are turning out to be of extremely promising abilities for real-time object detection, their practical implementation in car number plate detection for cityscapes is rendered difficult. While accurate, such models must also be optimized to run in real-time, especially in low-resource settings with limited computational resources and storage capacity [24]. The growing demand for real-time systems running optimally on embedded systems, such as edge devices, smartphones, or even low-cost surveillance cameras, calls for the design and optimization of light-weight, memory-intensive architectures. The prospect of deploying models like YOLO in real-time city traffic environments without losing detection accuracy is thus an area of focus for this work [34].

Furthermore, with the increased reliance on AI-based systems, there is growing concern over the "black box" of deep learning models, which dissuades the adoption and trust of users. For systems applied in public-facing domains, e.g., traffic law enforcement and security, it is paramount to have the AI system's decision process be not just correct but also interpretable. This research, therefore, integrates Explainable AI (XAI) technique like Local Interpretable Model-Agnostic Explanations (LIME) in a bid to maintain transparency and visibility in the vehicle number plate detection process. Through these methods, the research focused to build a system that is not just performance-efficient but also generates clean, interpretable explanation for its predictions, thus fostering trustworthiness and reliability [41].

At a personal level, this project offers a great opportunity to address an immediate urban problem while expanding my expertise in deep learning, model optimization, and AI interpretability. It also offers an opportunity to explore how emerging AI technologies, such as YOLO and XAI, can be applied in real-world applications to urban problems, to the benefit of both academia and society. The understanding and know-how from this research will be invaluable in guiding subsequent AI-driven solutions that are transparent, powerful, and scalable across an extensive list of public-interest spaces [38].

1.3 Objectives

The primary target of this thesis is to compare the performance of different YOLO-based models for detecting vehicle number plates and design a real-time detection application with an efficient approach. The goal is to ensure that the suggested solution is not just accurate but also feasible to implement in real-world traffic conditions, particularly in resource-constrained settings like Bangladesh. The research addresses the need for effective number plate detection systems with a guarantee of real-time capability and availability. The following purposes guide the overall research process:

- To develop a heterogenous number plate image dataset: The dataset will

contain of images taken from video under different environmental conditions, i.e., lighting, angle, and occlusion. Rotation, flip, and brightness modification data augmentation methods were applied to enhance the system's robustness and balance the dataset.

- For comparison of performances of different YOLO models for number plate detection: The models to be compared here are YOLOv4, YOLOv5, YOLOv8, and YOLOv12. All the models would be compared for major classification parameters such as precision, recall, F1-score, and mean Average Precision (mAP) on various Intersection over Union (IoU) values. The actual runtime performance of the models in frames per second (FPS) will also be compared to guarantee real-world usability in real-time traffic monitoring systems.
- To compare computation efficiency, detection performance, and model complexity: The research will establish the most computationally efficient YOLO model based on detection performance and real-time execution. The comparison will consider the trade-offs between the model complexity (parameters) and the ability to run efficiently on embedded systems or low-power devices deployed for traffic monitoring, e.g., edge devices and mobile platforms.
- To develop an application that detects number plates in real time from vehicle images: This goal wants to deploy an easy-to-use web app through the Streamlit library. There will be easy uploading of pictures of vehicles and real-time presentation of number plate detection. The application will be minimal, fast to operate, such that it will be easy to use where there are restricted environments or network speeds, such as traffic stop or in the countryside.
- In order to use the selected YOLO model in the application for real-time detection: The highest accuracy YOLO model (accuracy and performance-wise) will be used in the web application. The functionality of real-time detection will be adjusted to provide instant feedback to the users, and hence the system will be suitable for actual applications such as toll collection, parking, and law enforcement.
- For encouraging trust and transparency within the detection mechanism: To instill further confidence in users to the system, Explainable AI (XAI) technique including LIME (Local Interpretable Model-Agnostic Explanations) would be adopted in the app. The techniques will provide visualizable explanations for predictions made by models and will also help users interpret why number plates of a given vehicle were found, hence discouraging the "black box" issue common within deep learning mechanisms.

1.4 Methodology

The research process applied in this thesis is systematic and sequential, from data collection to the deployment of a real-time vehicle number plate detection system. Each process is such that it culminates into the generation of an effective, accurate, and understandable model that can be deployed in real urban traffic situations, particularly in resource-constrained settings like Bangladesh.

The first stage involved large-scale data collection from various vehicles on the roads in Dhaka, Bangladesh. High-resolution images of cars under varying conditions, including lighting, car orientation, and surroundings, were captured using an iPhone camera. The images included cars with varied number plates, with particular emphasis on the Bengali format employed for registration in Bangladesh, which has a tendency to mix Bengali script with Arabic numbers. This change in car number plates is depicted below, where photographs of a standard car number plate (right image) are presented. The dataset consists of images that reflect the dynamic nature of urban traffic. These images are manually verified and labeled with annotations to ensure consistency and accuracy of class labeling for the detection model.

Following data collection, preprocessing and image augmentation were carried out to ensure high-quality input for training deep learning models. Preprocessing entailed resizing all images to a standard size suitable for input to CNNs, and then normalization to standardize pixel values. Data augmentation strategies were subsequently applied to increase model sustainability and prevent overfitting. These methods included random rotation, horizontal and vertical flipping, zooming, and brightness adjustment, which added diversity to the dataset. The augmented dataset was divided into training and validation datasets to ensure the model's generalization capability.

For this research, five different YOLO-based models (YOLOv4, YOLOv5, YOLOv8, YOLOv12) were selected for comparison. These models are famous for their real-time efficiency in object detection tasks. All these models were trained on the vehicle number plate dataset annotated with focus on identifying the license plates in various real-world settings. The models were run on a high-performance computing platform, and the performance was compared in terms of applicable classification measures of precision, recall, and F1-score. The real-time processing FPS speed was also evaluated to determine whether deployment of the models in real-world contexts like traffic monitoring and law enforcement would be feasible. In order to counter the "black box" nature of deep learning models, Explainable AI (XAI) methods such as LIME (Local Interpretable Model-Agnostic Explanations) is integrated into the YOLO model's prediction pipeline. Both these methods produce visual and numeric explanations for predictions made by the model, telling users why a particular number plate was detected

and read. The integration of these methods is designed to increase the degree of user trustworthiness and confidence in the model, especially in sensitive domains such as law enforcement and security. The final aim was to develop a real-time vehicle number plate detection application using the Flask library. The application was designed to be light in terms of resources, interactive, and efficient to run on low-resource devices. The application allows users to upload car images and receive real-time detection results along with visual explanations generated by LIME, and SHAP.



Figure 1.1: Outcome of Detection with YOLO Model

Figure 1.1 shows the expecting results from this system. The application also provides a seamless interface for real-time detection of number plates for vehicle, with the added benefit of interpretability for each detection decision, which is essential for improving trust among users.

1.5 Project Outcome

This work has produced a series of beneficial outcomes that collectively add to the application of artificial intelligence in automobile number plate detection, the theme being urban traffic control in Bangladesh. The outcomes stretch from dataset development, via model performance, deployment readiness to system explainability, making the research not only academically sound but also viable for practical use in real-world urban traffic systems. Through the synergy of deep learning and real-time performance, the project addresses such critical challenges to urban traffic control as the demand for efficient, scalable, and interpretable AI systems to perform vehicle detection. This study serves to improve strong models to deploy in real-time traffic surveillance systems to help support law enforcement, tolls collection, and parking management. A miscellaneous vehicle number plate data set, representing a wide variety of conditions like diverse illumination, vehicle categories, and number plate styles (Bengali script and Arabic digits), was established in this work. The precise data set consists of a variety of annotated vehicle images representing the unique

©Daffodil International University

nature of Bangladeshi vehicle number plates, and a valuable source of reference for future vehicle detection work.

The study conducted a comparative systematic analysis of five different YOLO-based models, i.e., YOLOv4, YOLOv5, YOLOv8, and YOLOv12, to determine the most suitable model to employ for real-time detection of vehicle number plates. The models were comprehensively tested using key metrics such as precision, recall, F1-score, and mean Average Precision (mAP). The results offer a benchmark against which model selection is compared, establishing a guideline for subsequent research on vehicle detection systems and their use in urban environments. The comparison highlighted the trade-offs between model complexity, detection quality, and computation cost, particularly in real-time processing under limited resources. To meet the low-power, real-time demands of applications, a light-weight model based on YOLO was designed and trained to be employed in the urban traffic scenario. The optimized model to be deployed on edge devices such as Raspberry Pi and mobile phones showed equivalent detection performances with significantly lower computation overheads and inference times in comparison to big models. This achievement renders the model particularly effective for its usage on embedded systems utilized in monitoring traffic, policing monitoring, as well as CCTV cameras.

The study also incorporated Explainable AI (XAI) techniques, including LIME (Local Interpretable Model-Agnostic Explanations) and SHAP (SHapley Additive exPlanations) within the pipeline to identify vehicle number plates. The methods provided global and local model prediction explanation with visual and numeric explanation of detection results. This approach enhances the transparency and genuineness of the system, eradicating the "black box" issue with deep learning models, which is crucial for their acceptance by end-users such as traffic authorities and security officials. A real-time number plate detector application was developed with the Streamlit library to allow users to upload images of vehicles and obtain real-time detection output. The application provides real-time feedback and visualized explanation of the detected number plates, which makes it a user-friendly and simple traffic monitoring system tool. Deployed as a light and responsive application, it is designed to be efficient for deployment within regions with limited computational resources, so it can still be accessible and utilized even in low-connectivity regions.

1.6 Organization of the Report

The report is presented in six in-depth chapters, each specifically designed to always meet the research problem, summarize main literature, outline the approach utilized, summarize findings of implementation, and offer conclusions and recommendations. The format allows for a consistent presentation of information which reflects the technical scale of the study and the practical significance.

Chapter 1: Introduction

This first chapter establishes the context for the research. It explains the motivation behind the work, the technical and social context of identifying vehicle number plates, and the need to employ lightweight deep learning models for real-time use in city traffic systems. It also prescribes the research objectives, overview of research methodology, expected outcomes, and the report structure chapter-wise.

Chapter 2: Background

This chapter offers an extensive literature review of the existing vehicle number plate detection methods, with focus on YOLO-based models. The chapter explores the advantages of using deep learning models like YOLO for object detection, overviews previous research on number plate detection, and addresses the contribution of Explainable AI (XAI) towards the transparency and interpretability of these models. The chapter also presents the existing challenges and gaps in vehicle detection systems.

Chapter 3: Research Methodology

This chapter elaborates on the methodology adopted in the research. It starts with data collection for vehicle number plate images, followed by a description of preprocessing techniques including resizing, noise reduction. The chapter then analyzes the two primary phases: (i) model evaluation using YOLO-based models, and (ii) the design and development of a custom lightweight YOLO model for real-time vehicle number plate detection, improved for low-power environments.

Chapter 4: Implementation and Results

This chapter addresses the implementation details of the proposed work, comparative analysis of all YOLO models regarding accuracy, precision, recall, F1-score, and computational efficiency. The results also highlight the real-time detection capabilities of the models, comparing their inference speeds. A custom lightweight YOLO model is developed and tested against these benchmarks. The chapter also discusses the integration of Explainable AI (XAI) techniques for visualizing model predictions and improving user trust in the system.

Chapter 5: Engineering Standards and Design Challenges

This chapter summarizes the ethical and technical considerations that have been used in the design of the detection system. It outlines the difficulties encountered, for instance, optimizing the model, dealing with class imbalance, and ensuring interpretability of the system with high performance. The chapter also summarizes compliance to engineering and AI ethics guidelines, including considerations in terms of users' privacy, transparency, environmental sustainability, and usability.

Chapter 6: Conclusion

The last chapter recapitulates the research contributions through the creation of an equilibrated vehicle number plate dataset, YOLO model testing, and design of a lightweight yet accurate detection system. It also addresses the real-world applications on urban traffic control, law enforcement, and future research on real-time vehicle number plate detection. Suggestions for future work include expanding the dataset, improving model generalizability, and scaling up the system to additional domains and uses, i.e., to other vehicles or weather conditions.

Chapter 2

Background

This chapter presents an overview of the key ideas and technologies in the research work on automatic vehicle number plate detection using YOLO models. The chapter begins with a general introduction to machine learning, deep learning, and their application to real-time object detection, i.e., number plate detection for vehicles. The history of object detection models is dealt with in this chapter, keeping in mind specifically the YOLO (You Only Look Once) family due to its fast real-time speeds and accuracy, which has earned it prominence.

2.1 Introduction

This chapter is the foundation of the research report, with back ground information that will offer context to the methodologies and findings presented in the subsequent chapters. The emphasis of this research is to develop an automated vehicle number plate recognition system, a pertinent issue in the management of city traffic and security. Accurate and efficient vehicle recognition is needed for toll collection, police enforcement, and city surveillance. Accurate and timely identification of number plates have the potential to restrict traffic violations, help security forces, and ease traffic management systems in general. This opening provides the context for appreciating how important driving technologies and number plate recognition systems for cars are to making that a reality. Machine learning and deep learning have been the overwhelmingly dominant features for object detection solutions in the last few years, with automatic solutions being offered to identify and read number plates of cars with high accuracy. Among them, YOLO models have attracted considerable attention as they possess the ability to make real-time object detection with high efficiency. The chapter also discusses the use of Explainable AI (XAI) methods for raising the explainability and transparency of such models, which is needed to build users' trust in deployments with direct impact on law enforcement and public safety.

Considerable advancement have been achieved in the field of vehicle number plate detection, some problems remain, particularly regarding model interpretability, system scalability to actual traffic flow, and flexibility to varying environmental conditions such as light and occlusions. With an aim to avoid these problems, XAI techniques have been added to car number plate detection models. XAI techniques give explainability to the decision-making process of the

machine learning model in a way that is understandable by the users as to why and how a particular car number plate was detected and why the model predicted it. This is especially critical in actual-world implementations like traffic control and policing, where outputs from models must be interpreted if they are to be trusted, believed, and executed properly. By integrating XAI into vehicle number plate detection systems, this research aims to enhance both the technical performance of the detection system and its transparency, ultimately leading to more reliable and trustworthy systems for real-world use. The novelty of this research lies in its approach of comparing different YOLO models for vehicle number plate detection and integrating interpretability tools to improve the user experience and decision-making process. The chapter will further discuss the existing challenges and the innovations introduced in this study, setting the stage for the methodologies and experimental approaches presented in the following chapters.

2.2 Literature Review

Vehicle Number Plate Detection (VNPD) is a critical process in intelligent traffic monitoring systems. As traffic congestion increases in urban cities like Dhaka, Bangladesh, efficient vehicle recognition is crucial for improving traffic management, law enforcement, and enhancing urban security systems. Traditional plate recognition methods are not only time-consuming and prone to errors but also fail to function properly in real-time conditions. Solution-based deep learning techniques such as, Convolutional Neural Networks (CNNs), have been extensively researched to optimize and automate VNPD. Amongst them, the YOLO (You Only Look Once) algorithm has been at the forefront due to its efficiency and real-time processing capabilities. The present review critically analyzes a series of popular VNPD research studies dealing with YOLO-based detection methodologies.

Laroca et al. [5] had proposed a highly effective Automatic License Plate Recognition (ALPR) system using the YOLO detector. The emphasis was layout-independent number plate detection in various environments such that the model is simple to integrate in vehicles of various types and on many regional models. They emphasized how YOLO works with higher frame rate for vehicle number plate detection as well as stronger accuracy. But their strategy was preponderantly directed towards the recognition phase, which included character segmentation, a feature not addressed by this research.

In the research of Jamtsho et al. [6], a YOLO-based method was proposed for detecting license plates of non-helmeted motorcyclists in real-time. The authors used a single convolutional neural network (CNN) to detect license plates from video frames. To reduce false positives from similar objects like helmets, the centroid tracking method was employed. This method achieved a detection rate of 98.52%.

Their contribution is the efficient use of YOLO for real-time, practical license plate detection under various conditions with minimal human intervention.

Chung et al. [7] introduced YOLO-SLD, a newer version of YOLOv7, which uses the SimAM (Simple Attention Mechanism) to enhance the detection of license plates. Their method eliminates problems of varying lighting and orientations of license plates by using the attention mechanism, allowing the model to focus on the most important features. The results of the experiment indicated the accuracy was improved from 98.44% with the use of the standard YOLOv7 to 98.91% with the use of YOLO-SLD. Additionally, the parameter size of the model was reduced by 1.2 million, thus being more light but with improved performance.

Min et al. [8] proposed a new method of vehicle license plate detection using YOLO-L model and pre-identification of plates in order to enhance performance under challenging conditions, such as changing weather and view-point variations. The YOLO-L model was enhanced by modifying it to better detect using k-means++ clustering to choose candidate boxes for license plate regions with higher accuracy. The method achieved a recall and precision rate of 98.86%, outperforming current methods. The greatest value of this work lies in the combination of YOLO-L with plate pre-identification to remove spurious positives of objects like road signs and billboards.

The closely related study of Bhutanese license plate localization by Jamtsho et al. [9] was challenged by the unique nature of Bhutanese license plates, which were presented to it. The authors employed YOLOv2 for detecting vehicles and detecting license plates. The method localized the license plate regions from video streams with a mean average precision (mAP) of 98.6%. The model used YOLO for license plate localization and detecting vehicles with the avoidance of background similar objects such as signboards false alarms.

Ravi Kumar et al. [10] had suggested a machine learning-based vehicle number plate recognition system with the plate detection carried out by morphological operations, with plate detection prioritized using morphological operations. They used various image processing techniques like grayscale conversion, bilateral filtering, and edge detection in their work for the extraction of the region of the number plate. Optical Character Recognition (OCR) was used by them for recognition and character segmentation. Although their approach had high detection precision, they did use more conventional image processing methods, making future work involving deep learning to be used for number plate detection duties important. Rathi et al. recently.

Rathi et al. [11] recently employed the YOLOv4 deep learning model in real-time vehicle number plate detection in car parks. The vehicle number plates were detected by using YOLOv4, while Optical Character Recognition (OCR) was applied in reading out the recognized plate characters. The system was well detectable at around 89%,

which is suitable for application in urban monitoring and intelligent transport systems. The significant contribution of this work is the employment of YOLOv4 in real-time detection in a controlled setting, i.e., parking spaces, and highlighting the challenge imposed by uncontrolled settings such as low illumination and diverse orientations of cars.

Beratoğlu & Töreyn [12] suggested a YOLOv3 Tiny-based vehicle license plate detection system, wherein the model was implemented on compressed video data to reduce computational costs without compromising the detection accuracy. The method was found to demonstrate 30% reduced inference time compared to regular YOLO models in terms of detection performance. Their method significantly improved the applicability of vehicle number plate detection in resource-constrained environments so that it can be made useful for real-time traffic monitoring.

Ga et al. [13] enhanced the accuracy of YOLO with the Super-Resolution Generative Adversarial Networks (SRGANs) and combining them. The fusion proved particularly useful in improving low-resolution images that have been classically used in surveillance systems. Accuracy in detection in the proposed framework was 98.5% compared to normal detection systems that struggle to address low-resolution inputs. The implication of this paper is the enhancement of image enrichment techniques and the combination with YOLO as a solution towards addressing challenges such as low quality of images in real-world operations.

Min et al. [14] introduced the YOLO-L architecture, an improved version of YOLO, which included modifications to enhance the model's ability to detect and localize vehicle number plates. Their solution addressed problems like occlusion, varying lighting, and different angles of vehicles. YOLO-L was found to be 98.8% accurate, a highly effective solution in number plate detection under dynamic conditions. The key contribution of this study was the integration of a specially designed YOLO version to address real traffic conditions.

Kumar et al. [15] had conceptualized a number plate detection vehicle system highly appropriate for the context of Bangladesh's road scenario. They had integrated YOLOv4 into their system and identified that it was performing extremely well in the detection of the vehicle number plate extraction under various working conditions like heavy traffic condition and low light. The study achieved a mean average precision (mAP) value of 97.3% and established the competency of YOLOv4 for real-time detection use for the implementation of Bangladesh's city roads' smart transportation systems.

Table 2.1: Research Matrix for Number Plate Detection

Authors	Models	Accuracy	Key Findings
Laroca et al. [5]	YOLOv2, YOLOv3, CR-NET	96.9% (end-to-end recognition rate)	Developed a layout-independent ALPR system.
Yonten Jamtsho et al. [6]	YOLOv2	Test: 98.52%	Real-time license plate detection for non-helmeted motorcyclists using YOLOv2, achieving high accuracy in dynamic environments.
Chung, Lin, and Lin [7]	YOLOv7 + SimAM (YOLO-SLD)	Test: 98.91%	Introduced SimAM attention mechanism to YOLOv7 for improved accuracy and efficiency. Achieved 0.47% increase in accuracy and 1.2M fewer parameters.
Min et al. [8]	YOLO-L + K-means++	Test: 98.86%	YOLO-L with k-means++ clustering improves vehicle license plate location accuracy, with better precision and recall compared to YOLOv2.
Jamtsho et al. [9]	YOLOv2	Test: 98.6%	Achieved 98.6% mAP for vehicle and license plate localization using YOLOv2. Efficient real-time detection.
Ravi Kumar et al. [10]	Morphological operations + OCR	Not explicitly stated	Proposed a machine learning-based vehicle number plate recognition system using morphological operations and OCR for plate text extraction.
Rathi et al. [11]	YOLOv4 + OCR	Test: 89%	Real-time detection with OCR for plate text extraction in parking

			lots.
Beratoğlu & Töreyn [12]	YOLOv3 Tiny + HEVC Compressed Video	Test: 92%	Reduced inference time with compressed video for real-time license plate detection.
Ga et al. [13]	YOLO + SRGAN	Test: 98.5%	Enhanced detection of low-resolution images using SRGAN with YOLO.
Min et al. [14]	YOLO-L	Test: 98.8%	Improved vehicle and license plate localization with YOLO-L for better detection in low-light conditions.
Kumar et al. [15]	YOLOv4	Test: 97.3%	High accuracy in real-time detection on Bangladeshi traffic data with YOLOv4.

2.2.1 Similar Applications

The efforts of the deep learning and mobile technology in the development of the technology have greatly enhanced the performance of car number plates recognition systems. Specialists are researching the integration of up-to-the-minute machine learning methods and fast action to improve the efficiency, dependability, and overall applicability of detection systems where cities are concerned. Car number plate detection systems also adhere to the principles of image-based object detection methods using deep models to find and locate objects in such environments as the main roadway, parking garages or even toll booths.

Atikuzzaman and team [16] proposed a technique utilizing CNNs to detect number plates in practical time processes. The primary application of the system is real time detection of number plates from camera captured video streams. Developed with three principal phases. The system was developed with three principal phases: Plates were identified, class letters were segmented and characters recognized by the system. For plate detection, HAAR Feature-based Classifiers were used, followed by segmentation of class letter, and CNNs were used to perform the character recognition task. The algorithm achieved 91.38% recognition accuracy running at 30 frames per second. Under real conditions, the system achieved 96.92% accuracy in plate detection, up to 94.61% success in class letter separation, and 90.90% overall recognition. The performance of the system in a vehicle classification task was observed visually by testing on 390 images. The proposed approach is suitable for implementing ANPR technology in real-

world scenarios particularly for parking management; tolling operations and security applications.

Shashidhar et al. proposed a new vehicle number plate detection and recognition algorithm which automatically implements a YOLO-V3 for ROI detection and CNN for OCR recently [17]. Using a dataset of 6439 Indian number plates they created themselves, the system scored 91.5% accuracy. The underlying innovation of this work is the implementation of the Wiener filter for the real-time deblurring of vehicle images and subsequent improvement of older, pre-blurred images. In detection of license plates, YOLO-V3 was depended on whereas a CNN-based approach was used for the identification of characters. Use of a two-stage procedure allowed tough detection even in the noisy condition and this significantly improved overall system performance. The proposed system outperforms traditional license plate recognition systems because of the use of advanced pre-processing methods and optimized models to improve real-time performance.

Al-batat et al [18] used an end-to-end automated ALPR. Some of the goals of the system involved vehicle detection, license plate detection and classification of emergency and heavy vehicles, all designed to support the real world traffic monitoring. By eliminating supplementary pre-processing and post-processing procedures, this model can be easily embedded while generally providing good results for different data sets. The technique yielded amazing real-time results, feasible even on a general-purpose GPU of almost insignificant specifications producing decent frame rates. This hybrid approach exhibited significant gains in accuracy and processing speed when compared with former systems and was validated using data available for benchmark evaluation from five publicly accessible benchmarks.

Laroca et al. [19] built a strong and Through the two-stage fusion of YOLO v2 for detection and CR-NET for character recognition, the system outperformed commercial systems such as Sighthound (89.80%) and OpenALPR (93.03%). The research team generated the UFPR-ALPR dataset all of which has included numerous real-world scenarios ranging from different vehicle models, light conditions, and their angles of view. With enhanced vehicle and license plate detection, the proposed method provides a highly efficient solution for real-time deployment in License Plate Recognition systems.

Hossain and colleagues designed a high tech LPR system [20] using YOLO v8 for vehicle detection and OCR for character segmentation in their studies. The system achieved outstanding results at 99% precision regarding license plate localization and 98% accuracy in character extraction. Utilizing YOLO v8 for vehicle plate detection and an OCR for the segmentation of characters, this system effectively addresses challenges such as occluded vehicles and poor illumination in implementation. In addition, the researchers applied methods,

including k-means clustering, thresholding and morphological operations, to preprocess the input data, thus enhancing the performance of overall detection. The validity of the method was tested against 270 varied images, where it clearly outperformed previous solutions and was validated to be employed in real-time license plate systems.

Swapna et al. [21] formulated an automated system to recognize non-helmeted motorcyclists by combining vehicle license plate recognition and helmet detection with the support of YOLO v9. To identify the foreground – vehicles – from the background, video footage is subject to background subtraction, Canny edge detection, and morphological operations. The YOLO v9 algorithm promotes non-helmeted riders and afterward, the OCR is used to read license plate numbers. Compared to previous systems, this advancement exceeded the previous systems, with the former achieving 93% precision, 94% recall and 95.3% accuracy, which indicates its capability to aid immediate action against violations in motorcycle traffic.

In this work [22], Swathi et al proposed a real time AI approach to number plate detection utilizing RCNN and advance RCNN algorithms to facilitate vehicle and license plate identification. The main objective is to address problems, which arise from dynamic lighting and conditions in the environment, which are usually taken in practice environments. Using cloud environment, the authors improved its scalability and made it more user friendly. There was remarkable precision and recall from the system indicating that the former could be useful in intelligent transportation systems and establish high accuracy in various real world scenarios. The authors point out the critical role of parameter optimization, demonstrating how the cloud deployment provides improved throughput and scalability.

Developing a real-time vehicle detection system based on MobileNet-V2, YOLOx, and YOLOv4-tiny, in their work [23], T. Mustafa and M. Karabatak attain a high accuracy level of 97.5%. Vehicle recognition is fused with license plate recognition (ALPR) in the system evaluated under various scenarios, refined with Paddle OCR for improved plate segmentation to facilitate traffic control and vehicle safety.

Table 2.2: Similar Applications in Number Plate Detection

Authors	Model	Accuracy	Key Contributions
Atikuzzaman et al. [16]	HAAR feature-based classifier + CNN	Recognition Rate: 91.38% with 30 frames/sec	Achieved real-time performance using CNNs and HAAR classifiers.

Shashidhar et al. [17]	YOLO-V3 (ROI detection) + CNN (OCR)	Test: 91.5%	Developed a system for number plate recognition with image deblurring using Wiener filter and enhanced accuracy with YOLO-V3 and CNN.
Al-batat et al. [18]	YOLOv2/v4, ResNet50	Test: 90.3%	End-to-end ALPR without pre-defined rules; 5 datasets; real-time capable.
Laroca et al. [19]	YOLOv2, Fast-YOLO, CR-NET	93.5% (SSIG), 78.3% (UFPR)	Two-stage character recognition; data augmentation; new UFPR dataset; temporal voting.
Moussaoui et al. [20]	YOLOv8 + OCR	Detection: 99%, Character Recognition: 98%	Integrated YOLOv8 with OCR for high-precision vehicle identification.
Swapna et al. [21]	YOLO v9 + OCR	93% Precision, 94% Recall, 95.3% Accuracy	Introduced an automated system for non-helmeted rider detection, combining vehicle license plate recognition with helmet detection for road safety enforcement.
Swathi et al. [22]	RCNN + Advanced RCNN	Detection: 96.92%	Real-time vehicle number plate detection with cloud-based deployment.
T. Mustafa & M. Karabatak [23]	MobileNet-V2, YOLOx (vehicle identification) + YOLOv4-tiny (ALPR)	Test: 97.5%	Proposed a real-time vehicle and plate detection system with a focus on robustness in various environmental conditions, using advanced deep learning models.

2.2.2 Related Research

Recent advances in deep learning, that is, with Convolutional Neural Networks (CNNs) and YOLO models, have significantly developed automatic number plate recognition (ANPR) systems, which have made them more practical in actual use in real-time traffic monitoring. YOLOv3 has been used extensively for Region of Interest (ROI) detection in vehicle images with 91.5% accuracy in number plate detection, even when images are blurry, when used with CNNs for Optical Character Recognition (OCR).

The paper by Ottaviani et al. [25] describes a motorway traffic monitoring real-time number plate recognition system. The system records complete sequences of video, rendering external devices irrelevant in initiating image capture, thus a smooth run with no interference. By employing a rapid algorithm, the system reduces image processing time to below 40 milliseconds per frame, allowing real-time video processing even when dealing with high-speed traffic. The system consists of some local processing units networked by radio to a control station for full coverage of the traffic flow. It also has temporal post-processing to enhance recognition accuracy and achieve high recognition rate under clear conditions of illumination and real-time alarms for vehicles registered in a "black list.". The method is distinguished by its efficiency in car following, exploiting video frames to increase accuracy even in poor lighting, and provides insight into the practical application of automatic vehicle identification systems.

Roy et al. [26] had spoken about an improved segmentation method for Automatic Number Plate Recognition (ANPR), addressing the issues of non-uniformly shaped plates and the fonts employed in various countries. Using pixel-level segmentation and an Artificial Neural Network (ANN) for the detection of characters, the system achieves a 91.59% success rate for 150 number plates of varying countries. The paper introduces a new region-growing method for segmentation and gives users the capability to retrain the system's neural network based on personal font samples, which makes it more flexible for use with different global standards. The method provides a robust solution for global applications, especially in cross-border traffic management.

Sarbjit Kaur [27] brought an advanced level technique in the field Automatic Number Plate Recognition (ANPR) that facilitates the identification of low contrast, blurred, noisy, and multi-lighting images. Utilizing iterative bilateral filtering and adaptive histogram equalization as part of there preprocessing steps, the system improves image quality to aid subsequent number plate extraction with the aid of morphological operations and boundary box analysis. The technique efficiently addresses the challenges of extracting and segmenting license plates via character recognition, even in less than optimal conditions of environment. In contrast, the proposed approach outperforms conventional

ANPR techniques by displaying better scores on image quality metrics (MSE and PSNR) and attaining high accuracy in every number plate recognition process stage. The proposed technique is exceptional in real-time scenarios of number plate recognition where environmental disturbances are frequent.

Real-time ALPR system for the use of CCTV images was proposed by Sarfraz et al. [29], including special emphasis on non-dedicated cameras at low resolution. While under less than optimal lighting, unconventional camera perspective, and lowering the resolution, the proposed technique performs effectively in identifying and monitoring vehicles. System is able to perform robustly with automatic adjustments through background learning, dynamic license plate localization, and template matching methods. By the implementation of majority voting from several frames, the algorithm enhances character recognition and false positives. The use of this system is useful for traffic surveillance and forensic use cases because of its real-time efficiency and 94% detection rate; traditional ALPR technology cannot handle these cases. This research adds to the development of a scalable and robust ALPR system that would be suitable to be deployed on unregulated public CCTV networks.

The paper by M.S. Sarfraz et al. [31] presents a stable real-time automatic license plate recognition (ALPR) scheme adapted for application on CCTV video data, typically not possessing the resolution and quality of environment common in traditional ALPR systems. The proposed framework addresses key challenges such as noisy video, low-resolution videos, and varying light conditions, which are typical in forensic video scenarios. The authors introduce a new method for license plate detection, tracking, and recognition where the system learns and adapts to changes in camera views and distances and improves recognition rates over time. One of the major innovations in this approach is the use of dynamic background learning for detection and tracking so that it can stay robust even in difficult conditions. Moreover, the system is complemented by majority voting to remove recognition errors. The suggested system demonstrates significant enhancements in processing time and accuracy, particularly for processing CCTV video streams from low-quality security cameras. The contribution of this paper is highly valuable as it presents a viable solution to real-time license plate recognition in challenging, uncontrolled environments, particularly for security and forensic purposes.

Arth et al. [32] presented a real-time license plate recognition (LPR) system implemented on an embedded DSP platform, designed to operate autonomously using a standard video stream without additional sensor input. The system is composed of two key modules: license plate detection and character recognition. Detection uses a Viola-Jones-based AdaBoost approach, optimized with a Kalman tracker to predict license plate movement in subsequent frames, improving processing speed. Region-based partitioning is implemented on

characters, and to improve the process, support vector classification for the recognition is employed. By interpolating readings of several frames together, the system through improved resolution and robust recognition performance comes up even when the video is poor quality. Since it can use real-time performance, this embedded platform is quite suitable for traffic control systems as it provides a cost-efficient and effective solution to license plate recognition without requiring the usage of additional pieces of hardware.

Patel et al. [34] In this paper, many Automatic Number Plate Recognition (ANPR) systems are reviewed, their approaches, challenges, and proposed solutions are pointed out. Authors are concerned with the technical challenges related to differences between vehicle speed, plate styles, and environmental influences. They review a number of ANPR systems with ANN, OCR, and SVM, and recommend how further accuracy of these systems can be improved.

This paper [35] introduces a real-time license plate recognition (LPR) system designed to handle vehicle surveillance in varied environments, such as detecting stolen cars. The system is structured as a cascade framework, consisting of three key modules: plate detection, character segmentation, and post-processing. It leverages a one-pass scanning algorithm for efficient plate region extraction, minimizing computational overhead. The authors highlight the use of adaptive machine learning mechanisms, such as AdaBoost, to optimize classification and handle plate variations. The system’s primary contribution is its ability to process plates from diverse conditions, including varying illumination and environmental backgrounds, with a high accuracy rate exceeding 95% while maintaining a frame rate of 30 frames per second. The proposed method is effective in real-time applications and offers a significant advancement in LPR technology, particularly for surveillance systems requiring high computational efficiency and accuracy under dynamic conditions.

Table 2.3 Related Research in Number Plate Detection

Authors	Model Used	Accuracy	Key Contributions
Ottaviani et al. [25]	Real-time LPR system	Test: 95%+	Real-time video processing with high accuracy; adaptive for varying conditions; dynamic alarm system.
Roy et al. [26]	ANN-based segmentation method	Test: 91.59%	Improved segmentation for global license plate detection; adaptable to various fonts and plate shapes.

Sarbjit Kaur [27]	Iterative bilateral filtering, ANN	Improved MSE & PSNR	Enhanced performance under low contrast, blurred, and noisy conditions; real-time plate recognition.
Sarfraz et al. [29]	Template matching, dynamic learning	94% detection rate	Robust system for CCTV environments; improved accuracy under poor lighting and resolution.
Sarfraz et al. [31]	Dynamic background learning & majority voting	Enhanced accuracy	Real-time system for CCTV with robust background learning, ideal for forensic video analysis.
Arth et al. [32]	Viola-Jones AdaBoost, Kalman tracker	High accuracy	Real-time LPR system on DSP platform; efficient with low resolution and suboptimal video quality.
Patel et al. [34]	Various ANPR approaches	Varies	Survey of various ANPR systems; discusses challenges and improvements for accuracy.
Zheng et al. [35]	Cascade framework (detection & segmentation)	Test: 95%	Real-time LPR with high accuracy under dynamic conditions; effective for stolen car detection.

2.3 Gap Analysis

Despite significant advances within the Automatic License Plate Recognition (ALPR) community, there remain several gaps preventing effective deployment of robust and scalable systems for real-world usage, especially within metropolitan environments. The majority of existing systems, including the ones presented by Ottaviani et al. [37], are developed for particular milieus like motorways, with metropolitan traffic monitoring being left untouched. In addition, traditional methods are challenged by processing heterogeneous license plate layouts and poor images resulting from low lighting or occlusion, as highlighted in research by Roy et al. [38] and Sarbjit Kaur

[39]. Moreover, despite current research like Sarfraz et al. [42] aiming at CCTV, they are inadequate for low-resolution or non-dedicated camera systems. Several models heavily rely on expert hardware, hence their application in general traffic management at large scale is unrealistic, as identified in Arth et al. [34]. Additionally, the existence of an inflexible paradigm for handling ever-changing global license plate types and climatic conditions, as observed by Patel et al. [35], limits the scale on which such systems can be applied. My contribution addresses these limitations by working with YOLO-based models, which are adaptive, effective, and robust across many environments, conditions, and plate formats, making it a versatile tool for city traffic observation and live car number plate identification.

Table 2.4 Gap Analysis Summary Table

Gap Identified	Observed In	This Study's Contribution
Difficulty in detecting non-helmeted riders with plate recognition.	Swapna et al. [21]	This system focuses on vehicle number plate detection, solving real-time rider detection challenges.
Limited to motorway environments, lacks adaptability to urban settings.	Ottaviani et al. [25]	specifically designed for Bangladeshi vehicle number plate detection in urban environments.
Difficulty in handling non-uniform plate formats across countries.	Roy et al. [26]	This system uses YOLO-based models that are adaptable and efficient across various plate types and fonts.
High noise and blurred conditions often degrade recognition performance.	Sarbjit Kaur [27]	Deep learning-based YOLO models perform exceptionally well under challenging conditions, like occlusion and low lighting.
Struggles with CCTV footage from non-dedicated, low-resolution cameras.	Sarfraz et al. [29]	Integrates real-time detection for CCTV footage with high accuracy, regardless of resolution or camera type.
Relies on specialized hardware for real-time recognition.	Arth et al. [32]	This work uses lightweight, high-efficiency YOLO models that can run on general-purpose devices and embedded systems.

2.4 Summary

This section described the current vehicle number plate detection situation with particular focus on the application of deep learning techniques such as Convolutional Neural Networks (CNNs) and Transfer Learning (TL), with particular reference to YOLO models. YOLO-based models have been very promising for vehicle number plate detection, with high accuracy and efficiency rates, particularly in real-time. Yet, a few issues are still pending, like handling varied lighting, background noise, and non-standard plate formats. Transfer Learning has surely provided a tangible solution to mitigate the issue of having few labeled samples by fine-tuning existing models like YOLO, but still, it suffers from domain shift and requires careful model adaptation for varying plate types. Additionally, Explainable AI (XAI) methods such as LIME, and SHAP have become indispensable in the process of making deep learning models more transparent, revealing insights about model decision-making. Although promising, such uses in vehicle number plate detection remain yet to receive widespread traction, especially in real-time settings. These deficiencies demonstrate the need to create interpretable, effective, and scalable systems of detection. The aim of this investigation is to address previous restrictions by combining a streamlined YOLO-based method for instantaneous plate detection, augmented by XAI components to improve use in real-world applications.

Chapter 3

Research Methodology

In this chapter, we outline the methodology undertaken in the development of a real-time accurate and easily read vehicle number plate recognising system using YOLO models. The method of data gathering, transformation, building a model architecture to be tested, and overcoming the challenges associated with the difference in illumination and background details typical in Bangladeshi road scenes is the theme of the chapter. Using the high-throughput speed of YOLO, supplemented by Explainable-AI (XAI) methods, the method gives a peek behind the reasoning of the system and the confidence of its predictions. This approach endeavours to provide a powerful, practical solution for vehicle number plate recognition that can be deployed into the real world.

3.1 Methodology

3.1.1 Overview

Taking a comprehensive and strong approach, this work proposes to build an effective, lightweight real-time system for vehicle number plate detection using deep learning with emphasis laid on YOLO models. To diversify the dataset concerning the vehicles and number plates common in Bangladesh, image data is first obtained from various places like the intersection of main roads and private institutions. Using the raw data obtained, then the next step is on data preprocessing wherein the size of the images as well as its values are being adjusted in order to make them normal and uniform in presentation. In order to make the data set bigger and more diverse, image rotation, image flipping, noise addition, and contrast adjustments are used as data augmentation methods.

After preparing the data set the vehicle number plate detection system is trained using the YOLO models (v8, v9, v10, v11 and v12). Each YOLO model is trained and evaluated in order to determine the most accurate, fast and computationally efficient model. In-depth comparison between these models is done to discover which of them provides the most efficient solution for real-time vehicle number plate detection in specific traffic conditions. Consequentially, the model is evaluated at this point using such parameters as precision, recall, and F1-score to ensure its usability in actual practice. In the conclusion the model that has the best performance output and therefore can make real-time detections is selected thus verifying its appropriateness and ability for deployment in traffic surveillance systems and other relevant uses.

3.1.2 Proposed Methodology/ System Design

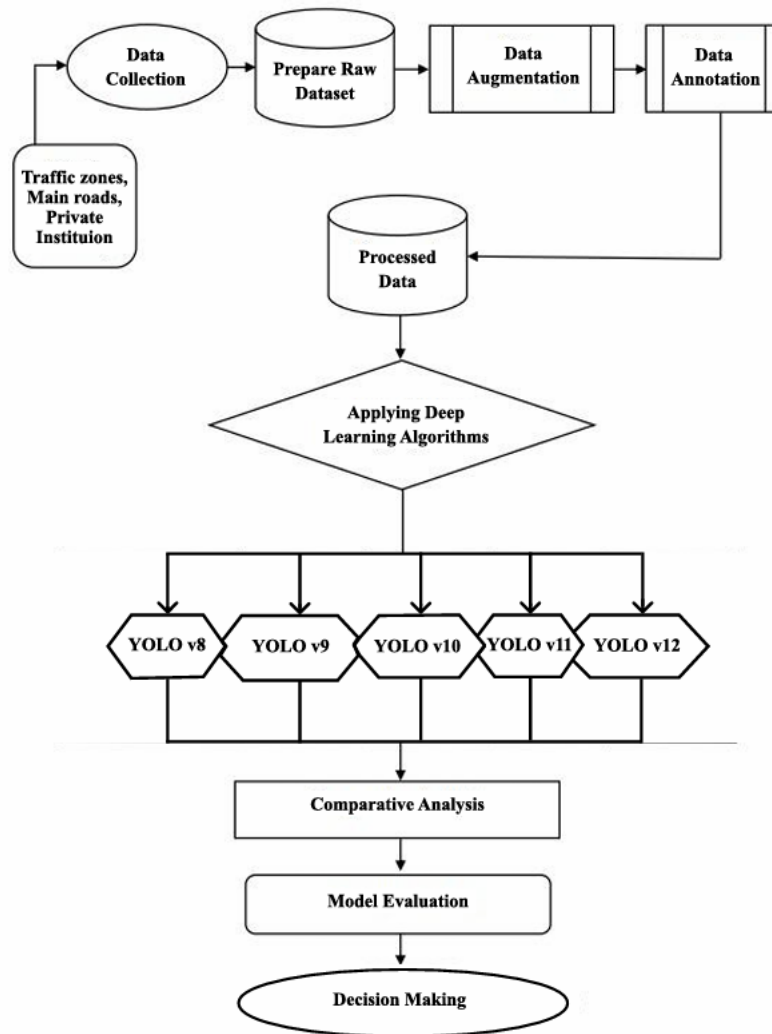


Figure 3.1: Research Workflow for Number Plate Detection

The present work tries to develop an effective vehicle number-plate-detection system using YOLO deep learning techniques, where data are collected from various traffic zones, including highways and private organizations, to make the system more practical and effective. In order to increase model efficiency, the dataset is preprocessed by applying an image resizing process, flipping images, rotating images and scaling the images. Training, validation, and testing subsets are used for the datasets, in which YOLO versions (8,9,10,11,12) are trained and evaluated by parameters like accuracy, precision, recall, F1 score, and the performance of confusion matrix. Hyperparameter tuning, early stopping and learning rate scheduling are all utilized to avoid overfitting, and the number plate model's transparency is improved using Explainable AI techniques SHAP. In order to improve usability, the project has an aesthetically designed dashboard, to which users can upload photographs and obtain immediate confidence scores for the recognized number plates. The system's performance is evaluated through error analysis and model

comparison, ensuring its robustness for practical deployment in real-world traffic surveillance.

3.1.3 Functional and Nonfunctional Requirements

The proposed system has essential functional and nonfunctional requirements to ensure robust performance and practical usability. The functional requirements focus on the system's core tasks, while the nonfunctional requirements ensure it meets performance, usability, and security standards in real-world applications.

Functional Requirements

The system that is envisioned possesses primary functional requirements to deliver sound performance and practical world usability. These are to focus on the main task of vehicle number plate detection, prioritizing the system's performance under practical world traffic conditions.

- **Image Upload:** The system allows users to upload images of vehicles either by capturing photos using a device camera or by selecting from the device gallery.
- **Image Preprocessing:** The system shall automatically resize, normalize, and enhance the uploaded images to prepare them for number plate detection.
- **Vehicle Number Plate Detection:** The system shall detect the vehicle number plate from the uploaded image using deep learning algorithms (YOLO models) and classify the detected plate area.
- **Model Selection and Inference:** Initially, the system will run multiple YOLO models (v8, v9, v10, v11, v12) to select the optimal model based on accuracy, efficiency, and inference speed. The final model (YOLOv12) will be used for inference.
- **Prediction Output:** The system will display the detected vehicle's number plate region along with the predicted number plate and a confidence score (e.g., "Confidence: 95.2%").
- **Explainability Visualization:** The system will provide interpretable results by displaying Explainable AI (XAI) visualizations such as SHAP to highlight important regions of the number plate.
- **Feedback Module:** The system will notify users of successful uploads, processing status, or any errors encountered during the process. Feedback will be provided via visual or text-based notifications.
- **Offline Prediction Capability:** The system shall support offline functionality, enabling users to upload and process vehicle images without an internet connection.
- **User Interface:** The system will include an intuitive, user-friendly web interface, allowing users to interact seamlessly with the system for uploading images and viewing results.
- **Logging and Storage:** The system shall provide appropriate error handling and troubleshooting for incorrect or unsupported file types, low-quality

images, or processing errors.

Nonfunctional Requirements

Nonfunctional requirements define the quality characteristics and operational constraints of the system, ensuring its usability, performance, and reliability in real-world scenarios.

- **Performance:** The system must provide inference results within 3–5 seconds on both mobile and web platforms.
- **Accuracy:** The lightweight YOLO-based model should achieve at least 90% accuracy for vehicle number plate detection, validated on real-world and augmented datasets.
- **Usability:** The system interface must be intuitive, responsive, and simple, catering to users with minimal technical expertise, such as drivers, traffic authorities, and security personnel.
- **Scalability:** The system should be scalable to support future updates, such as the integration of additional vehicle categories, new plate designs, or the incorporation of other traffic-related data.
- **Portability:** The solution must be portable across devices and platforms (Android, iOS, and web) with varying hardware capabilities.
- **Maintainability:** The application’s modular architecture must support ease of updates, debugging, and version control without significant downtime.
- **Reliability:** The system should function consistently across different environments, including variations in lighting, weather, and vehicle speed, and provide dependable results under standard operating conditions.
- **Security and Privacy:** User-uploaded images and personal data must be encrypted during storage and transmission, complying with standard privacy policies.
- **Resource Efficiency:** The lightweight YOLO model should be optimized to run efficiently on low-end mobile devices with limited CPU and memory resources.
- **Responsiveness:** The UI must maintain responsiveness and visual consistency across different screen sizes and resolutions.

3.1.4 Context Diagram

The Context Diagram delivers an abstract perspective of the vehicle number plate detection system operated as a single process and interactions with the external parties. It defines what is inside the system clearly indicating what happens with data flowing between the application and users or external inputs. In this project, the focal conversion takes place between the User and the system where users upload vehicle images to request bounding box predictions with corresponding visual explanations. All domestic actions within the system are included and include preprocessing, classification with YOLO models, and the incorporation of explainability features through LIME, and SHAP.

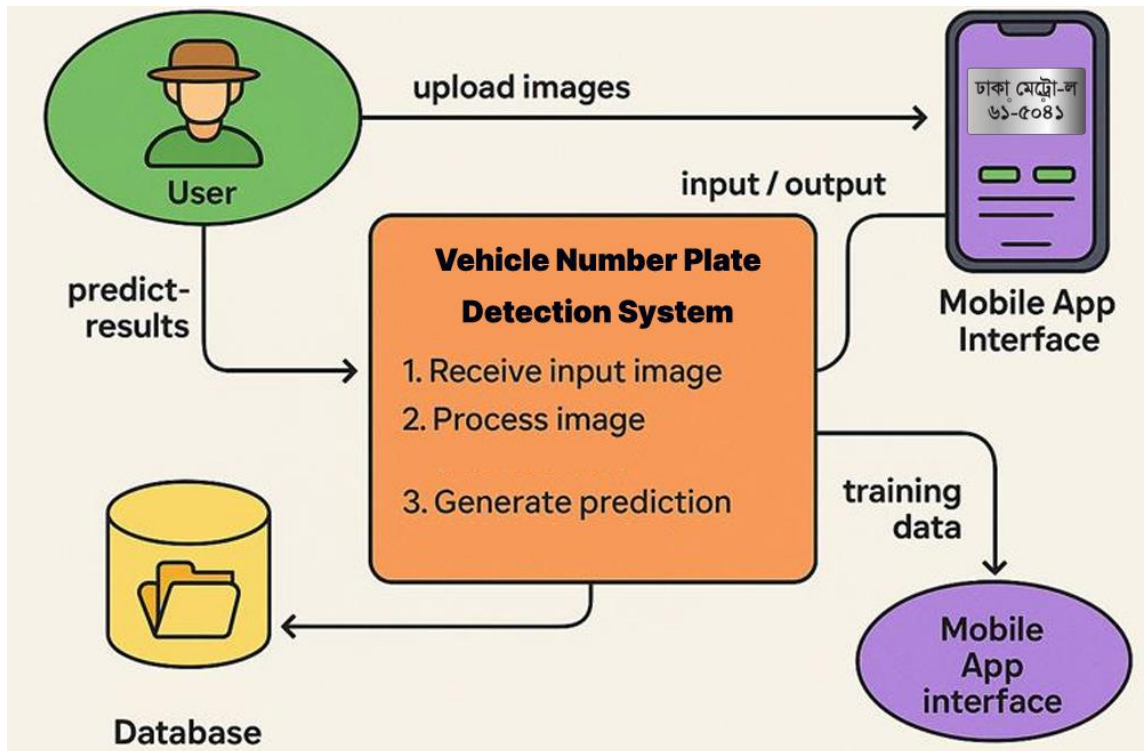


Figure 3.2: Context Diagram of the system

3.1.5 Data Flow Diagram Level 1

The Level 1 Data Flow Diagram (DFD) is used to describe the internal flow of data in the Vehicle Number Plate Detection System, showing how data is moved from one system module to another. It begins with the user uploading an image of a vehicle number plate, which is received by the Image Input Module. The image is passed to the Preprocessing Module, where it is resized, normalized, and augmented. The enhanced images are stored in the Augmented Dataset for future training and testing. Subsequently, the image is transferred to the Model Inference Module, where the image is inferred via the YOLOv8 model to identify the number plate of a car. Transfer learning processes are utilized by the model to fine-tune performance against the dataset. After inference has taken place, the output number plate, accompanied by confidence, is transferred to the Explanation Module. This module utilizes Explainable AI (XAI) methods, such as LIME, and SHAP to provide explainable outputs by identifying the areas of the image that influenced the model's decision. The resulting final prediction outcome with the identified number plate, confidence measure, and explanation is then displayed to the user through the Result Display Module. This Level 1 DFD defines the decomposition of the system in modular manner as well as the precise process order through which the raw input image is translated into comprehensible predictions with explanations so that outcomes become user-friendly, interpretable in nature.

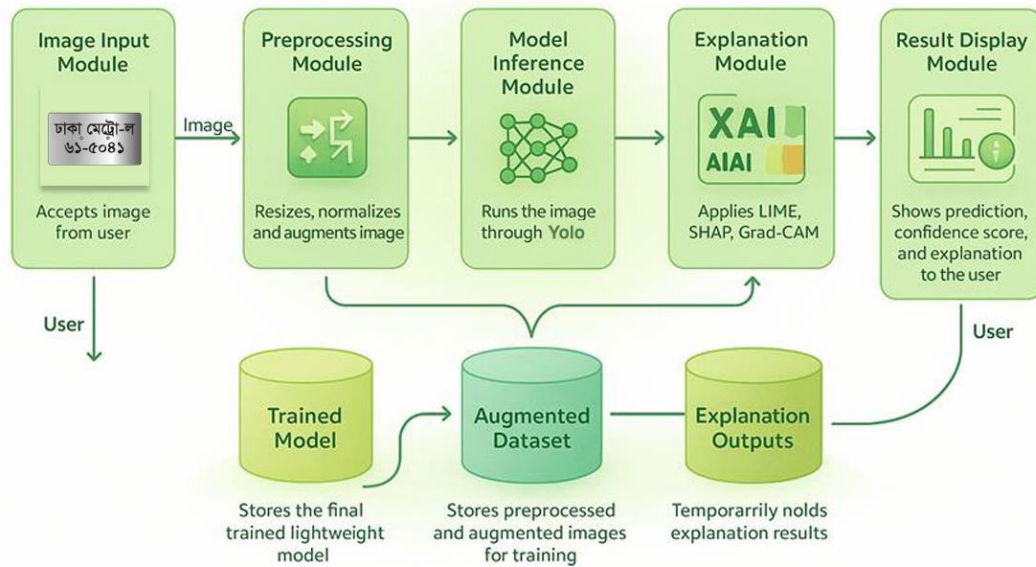


Figure 3.3: Data Flow Diagram of the System

3.1.6 UI Design

The provided UI screenshot illustrates the user interface of the "Vehicle Number Plate Detection Using YOLOv8" system. The interface is designed to offer a simple and intuitive experience for users.

1. Image Upload and Detection

There are two buttons on top. The "Choose File" button allows users to input a photo of a car, and the "Detect" button starts the detection process. The system utilizes the YOLOv8 model to recognize the number plate in the photo.

2. Detection Result

Immediately after the image is uploaded and processed, a bounding box depicting the number plate is overlaid to the vehicle image and, subsequently, the system's output. Along with the bounding box, the confidence level (for instance 0.76) is visually shown to express the confidence given by the model of recognition.

3. Output Display

The detected objective ("license-plate") is labelled in a written form under the visual presentation. The system reveals such information to inform the user of the object that is being detected by the system.

The user's experience is optimized with an easy to use interface allowing non-technical users to identify vehicle's number plates in real time applications.

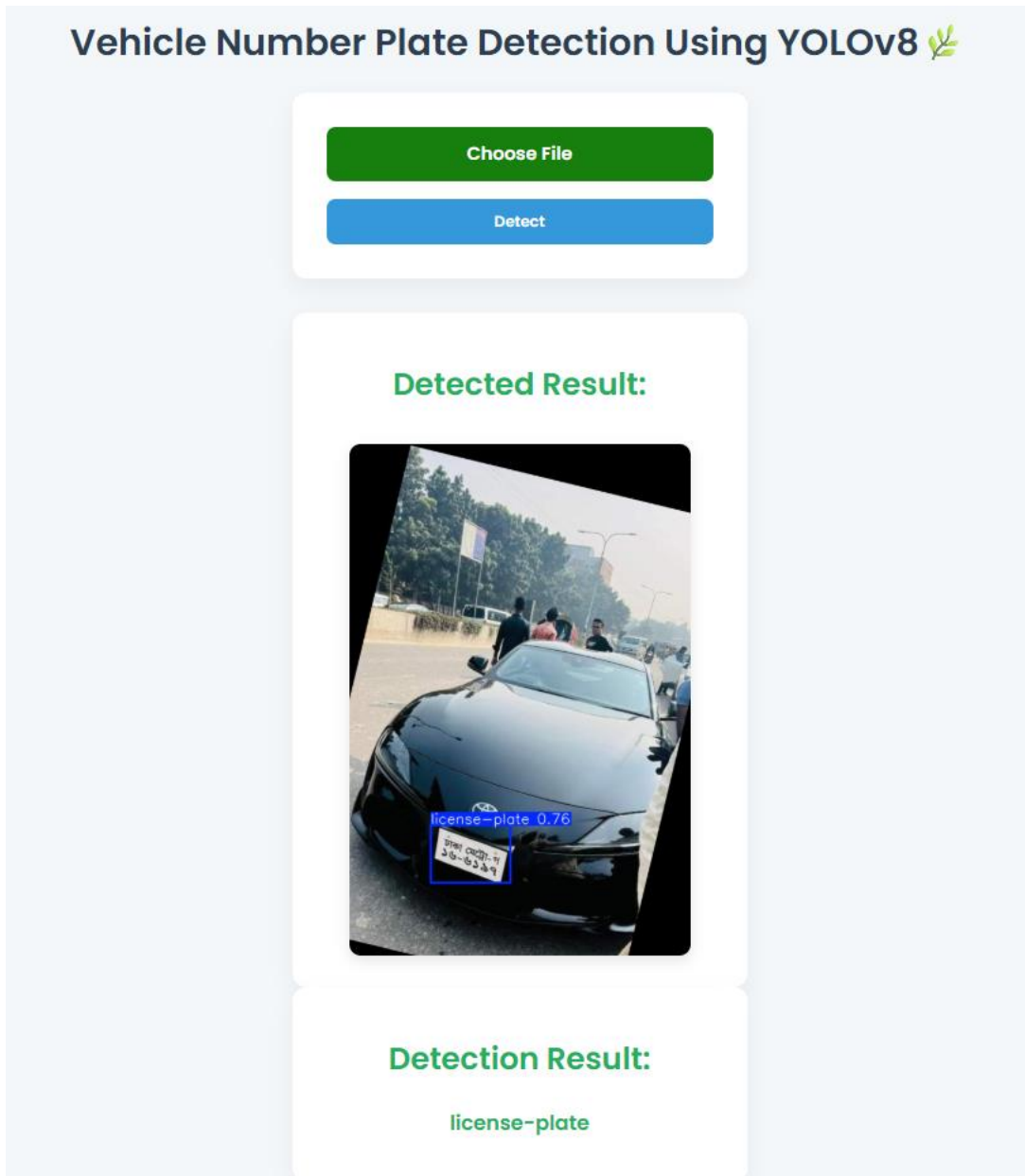


Figure 3.4: UI Design of Number Plate Detection System

3.2 Detailed Methodology and Design

The methodology underlying the “Vehicle Number Plate Detection Using YOLOv8” is designed meticulously to ensure that real-time vehicle number plate detection is done at both speed and accuracy. The work is initiated by gathering a data set of assorted vehicle images where they have number plates gathered from various environments such as various weather, lighting and traffic situations. Using such a broad spectrum of images is a must for the construction and testing of the YOLOv8 model. The first stage is pre-preprocessing of data to make the collection more diverse and high-quality. Normalization, standard-sized resizing, and forms of augmentations such as flipping, rotating, and varying brightness are performed on

the images, giving rise to a more flexible and trustworthy dataset. The modified dataset is divided into three distinct subsets, including: testing, validation, and training. The three sets are the inputs for training YOLOv8 model, optimizing its settings, and evaluating it.

For Phase II, we use the real-time object detection model YOLOv8, which relies on a unified neural network to predict bounding boxes and object classes, all at once. YOLOv8 has a convolutional neural network (CNN) that works on images through a system of structured stages. The image is first passed through the backbone layer groups of the network where many convolutional layers collaboratively create the feature extraction. After that the features are being processed by the detection head that produces bounding box coordinates and class predictions to detect the vehicle's number plate. The efficiency of YOLOv8 relies on the fact that it uses anchor boxes which enable it to produce several bounding box predictions for each object in the picture thus realizing accurate detection. Additionally, YOLOv8 employs a multi-scale feature, meaning objects at various scales, even small number plates on cars, will be detected at high accuracy. The output of the model is both the bounding box and confidence score, which is the confidence of the model in the prediction.

At the third stage, for reasons of enhancing interpretability, Explainable AI (XAI) techniques such as SHAP (Shapley Additive Explanations) is integrated into the system. Such techniques enable insight into regions of the image that played a pivotal role in model decision-making, thus enhancing transparency of detection results to end-users like traffic authorities. Finally, the system is implemented on a web application, enabling the uploading of images of a vehicle and obtaining real-time predictions displayed with the detected license plate and confidence. The deployment makes the system realistic and simple to use in various settings, such as traffic monitoring systems and security programs, with accurate vehicle number plate detection in motion. The end-to-end architecture from deployment to preprocessing assures high efficiency, accuracy, and explainability for the system, thus making it a valuable resource for real-time vehicle identification.

3.2.1 Dataset

3.2.1.1 Data Collection

The vehicle number plate detection dataset was put together such that images across different traffic spaces, including roads, private organizations, and congested traffic spaces were gathered. Difference in image resolution, quality and in environmental settings was attained by using both smartphones and digital cameras to take photographs. Data for the first dataset was collected from November to December 2024 within a period of one month. In order to create more heterogeneity in the dataset, multiple devices were used to take images of different traffic spots and

weather conditions. Data was systematically collected early in the morning, midday and late afternoon to incorporate various lighting scenarios and perspectives (Top and side views). After removal of dups and inconsistent images in a first cleaning up, the dataset was a total of 3507 images. To improve dataset diversity and model performance, a wide range of data augmentation strategies (rotation, flipping, scaling, and noise injection) were used, which significantly increased dataset size.

These estates were selected to cover all categories of vehicles with number plates in a body view. The image recording was performed using a Realme C25s smartphone and a Canon EOS 70D digital camera. The use of both devices brought variability in image quality, colour rendition and viewpoint which eventually facilitates better generalization of the model. The images recorded were at different hours of the day (morning, noon, and late afternoon) and from top-view and side-view angles to maintain natural changes in light and perspective. To achieve optimization on both the quality of the images and processing, the images were captured as JPG files. Table 1 carries all the relevant device specifications.

Table 3.1: Specification of the data collection devices

Feature	Realme C25s	Cannon EOS 70D
Camera		
Manufacturer	Sony	Canon
Camera Model	C25s	EOS 70D
Aperture Value	f/1.8	f/4.0
Resolution	2304 x 4096 pixels	6000 x 4000 pixels
DPI	72	300
Bit Depth	24	24

3.2.2.2 Dataset Description

The dataset used in this study for vehicle number plate detection initially consisted of 3507 images captured from various traffic zones, roads, and private institutions. The data was first cleaned, which included removing duplicate images and irrelevant data, resulting in 3045 images for further processing. To enhance the dataset's diversity and improve the model's generalization capabilities, data augmentation techniques such as rotation, flipping, scaling, noise injection, and contrast

adjustments were applied. These augmentation techniques increased the dataset size significantly to 7495 images, making it suitable for deep learning model training. My dataset is only used for detection purpose, so there is only one class named “Number Plate”.

The dataset is split into three sets for model training: 86% of the data (6442 images) is used for training, 7% (524 images) for validation, and the remaining 7% (524 images) for testing. This data distribution ensures the model can generalize well to unseen data while effectively validating its performance during training and evaluation. The pie chart below, titled "Dataset Distribution". “License plate" is the only one category, representing 6488 images (86%) for training, 505 images (7%) for validation and 505 images (7) for testing purpose. But this balanced distribution helps the model learn to distinguish between different plate conditions and environments.

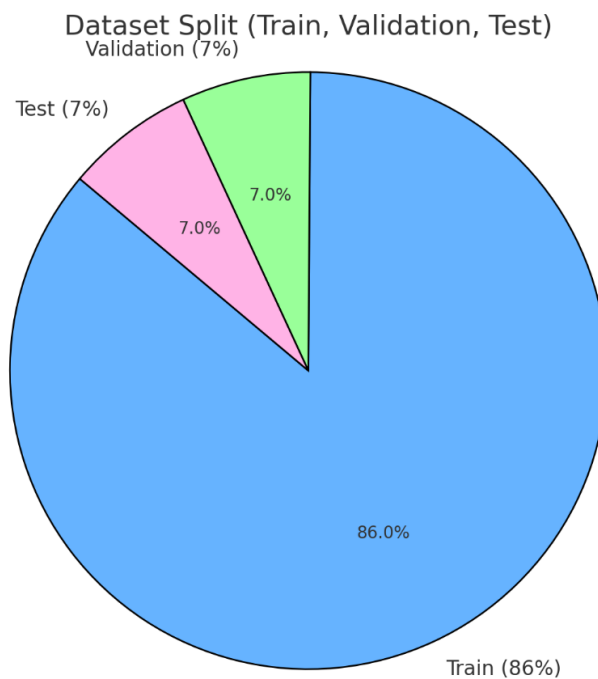


Figure 3.5: Dataset Distribution Chart

Figure 3.7 illustrates representative different types of sample images, highlighting the visual differences in number plate box. These examples reflect the diversity and complexity present in the dataset.



Figure 3.6: Sample Images from the Dataste

3.2.2 Data Preprocessing

Before training the models, several preprocessing steps were applied to the raw dataset to ensure consistency, reduce computational complexity, and address class imbalance.

- **Image Resizing:** All images collected for the project were resized to a uniform dimension of 720×640 pixels using Microsoft PowerToys. By normalizing the resolution of the data, we retained a uniform input for deep learning models, resulting in reliable feature extraction and increased training efficiency. Through the selection of a proper resolution, important visual information was preserved for proper classification, and memory consumption and computational requirements were reduced. □
- **Dataset Splitting:** The data was split into three segments: 86 percent for training, 7 percent for testing and 7 percent for validation. By using stratified sampling, the dataset was split such that each of the subsets had the same class distribution seen in the overall dataset. Fair and reliable model performance evaluation would be achieved if equal proportions of each class were maintained in the subsets.
- **Data Cleaning:** A manual analysis was carried out exhaustively to cleanse the dataset of duplicated or not-on-topic pictures. By using the process, the dataset was curated to include only well-labeled and optimally visual images. All images were analyzed very carefully, and unwanted samples were eliminated to avoid corrupting the datasets data quality for training models.
- **Annotation:** Annotation of the dataset was done using free version of Roboflow. The tool allowed the images to be labelled and sorted by the vehicle number plate areas and improving the precision of the model that could identify these regions. The process of training started with the YOLOv8 model, armed with the annotated images.

3.2.3 Model Selection

This research aimed to compare the models' abilities of five different YOLO (You Only Look Once) models in vehicle number plate detection, specifically choosing

YOLOv8m, YOLOv9m, YOLOv10m, YOLOv11m, and YOLO YOLOv8m, YOLOv9m, YOLOv10m, YOLOv11m, and YOLOv12m. These models were used to analyze the changing performance of YOLO in various versions – the progression of the specific structure of the model, optimization approaches, and methods of feature extraction served as critical considerations. Since the YOLOv8m version is the one that is used most frequently, it was used to create the baseline model. YOLOv9m, YOLOv10m, YOLOv11m, and YOLOv12m were chosen for this analysis to measure how each successive version would perform that introduced new backbone networks, anchor configurations, and The reason for choosing these particular models was as follows:

- **Incremental Improvements:** Every update of YOLO is better at speed, accuracy and computational efficiency that places them well for real-time vehicle number plate detection applications that require high performance.
- **Real-Time Inference:** This is highly appreciated in such incidences as traffic monitoring and law enforcement where speed and efficiency in NPD are essential.
- **State-of-the-Art Detection:** Having excellent object recognition features specifically in vehicle identification, YOLO models are perfectly capable of dealing with tasks that involve number plate recognition.
- **Compatibility with Mobile and Edge Devices:** YOLO models, especially with the medium versions like YOLOv8m to YOLOv12m, are optimized for deployment on mobile and edge devices with lower computational resources.

a) YOLOv8

Basing on the success of its predecessors, YOLOv8 improves the YOLO platform by providing the efficient and precise object detection in real time. YOLOv8 builds upon the YOLOv7 and YOLOv6 series by integrating improvements that improve speed, accuracy, and change in the architecture. With the addition of a new backbone network and the adjustment of the detection head, YOLOv8 gains superior feature extraction and more trustworthy detection outputs, which are of practical use in real-time vehicle number plate detection tasks. Capable of operating in real-time, the model is particularly excellent in object recognition under different environments such as low-resolution visuals and objects moving fast, making it well-suited to vehicle number plate recognition. YOLOv8's simplifying and efficient design allows it to work with big images with speed and reliability, which makes it an excellent potential performer on constrained systems and mobile apps.

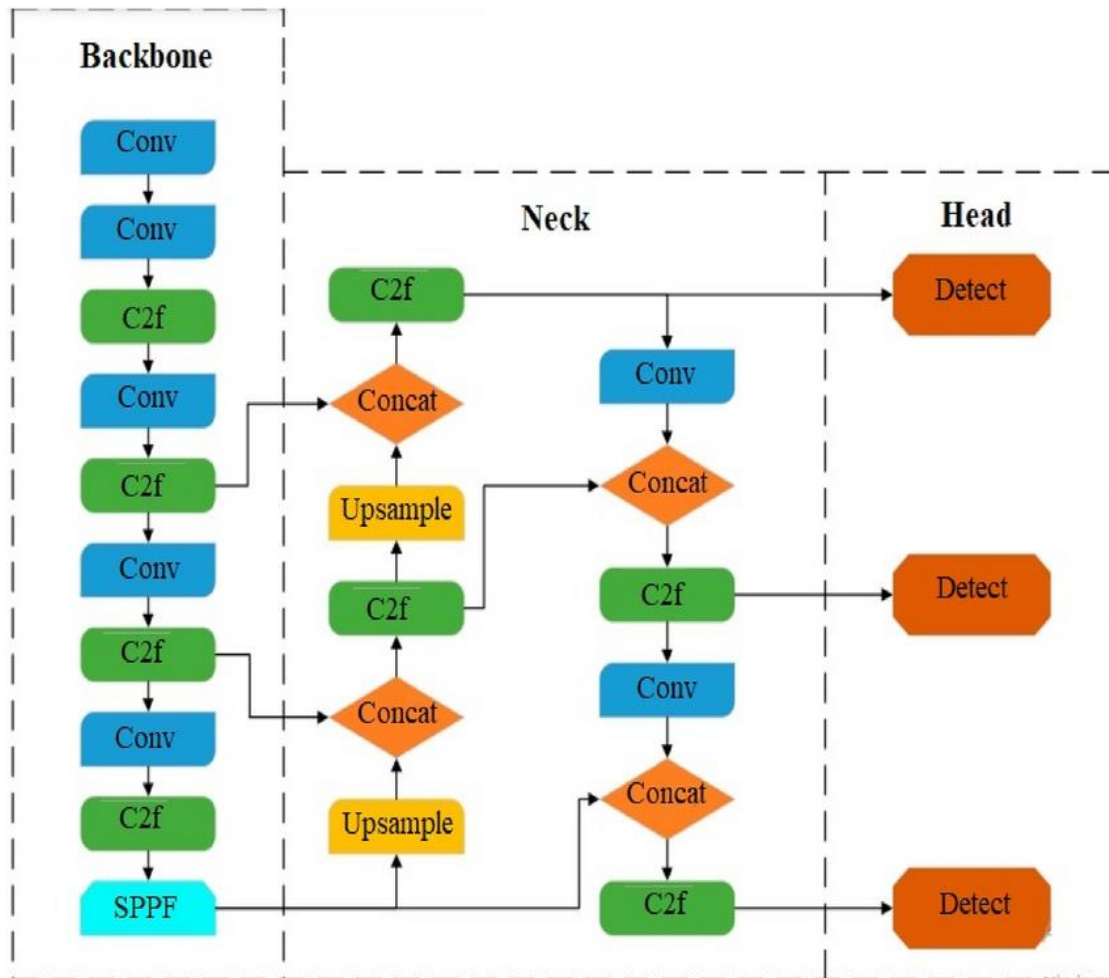


Figure 3.7: Architecture of YOLO v8 (Medium)

b) YOLO v9

YOLOv9 is an advanced iteration of the YOLO framework optimized for speed and accuracy and is particularly optimal for real time applications such as the detection of vehicle number plates. The design is made up of three separate sections: Backbone, Neck, and Head. In non-reversible connections, the Backbone aggregates elementary characteristics from the image but diminishes the spatial resolution, allowing important visual information to be preserved. Through reversible connections on the part of the Neck of the architecture we can promote a smoother flow of feature sevant and semantic feature extraction, and let YOLOv9 be more precise in delineating complex spatial structures. The ultimate steps are performed by the Head who estimates the positions and labels of the objects and generates corresponding bounding boxes. This design will enable YOLOv9 to identify vehicle number plates very fast, and with a lot of precision, thus becoming a great solution to the real-time vehicle detection problems.

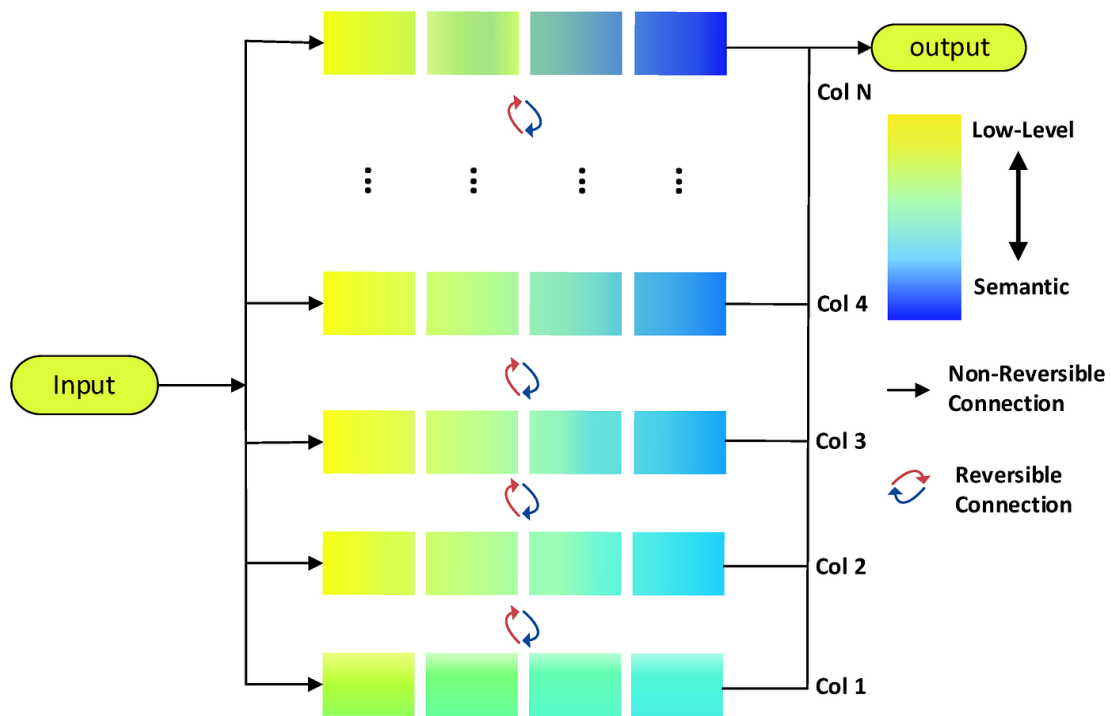


Figure 3.8: Architecture of YOLO v9 (Medium)

c) YOLOv10

Dual-label assignment usage in the architecture of YOLOv10 enhances detection performance as it effectively address regression and classification needs. The design integrates the Backbone, PAN (Path Aggregation Network), and Heads for classification and bounding box regression. The backbone determines relevant features while the PAN maximizes feature map traffic for effective utilization of information at multiple scales to increase accuracy for localization. The novel Dual Label Assignment scheme partitions out the task into distinct one-to-many and one-to-one heads, one that processes a particular number of bounding boxes. With this approach more accurate results are assured, as well as better control of overlapping objects. With the assistance of the Consistent Match Metric, the system matches predicted bounding boxes with ground truth annotations effectively using scores and IoU scores. YOLOv10, due to its speed in object localization, is especially good for real-time detection of vehicles' number plates.

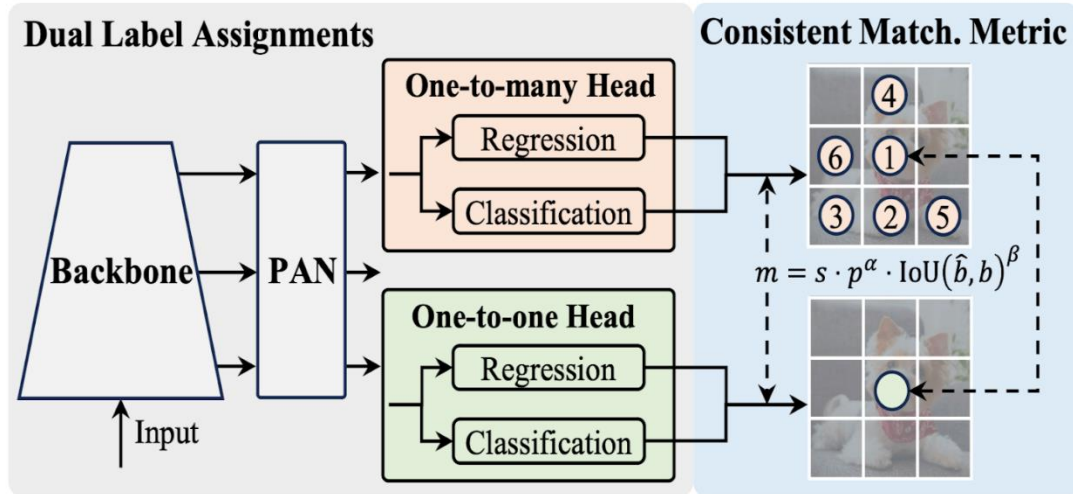


Figure 3.9: Architecture of YOLO v10 (Medium)

d) YOLO v11

YOLOv11 architecture compromises neither speed nor accuracy by balancing the efficiency needed to retrieve high level spatial and semantic features with low computational cost ergonomics. It includes modified backbone parts C2f, C3K2, and C3K. The C2f block has a convolution layer followed by feature splitting and bottleneck, which is useful for effective feature learning. The C3K2 block uses stacked 3x3 convolutions with concatenation which enhances multi-scale feature richness. The C3K block adds bottleneck layers to improve accuracy oriented computation efficiency. Such arrangement improves the precision of locating and identifying vehicle number plates in images, so real-time ID retrieval can be done seamlessly, making these features enable wider a range of applications.

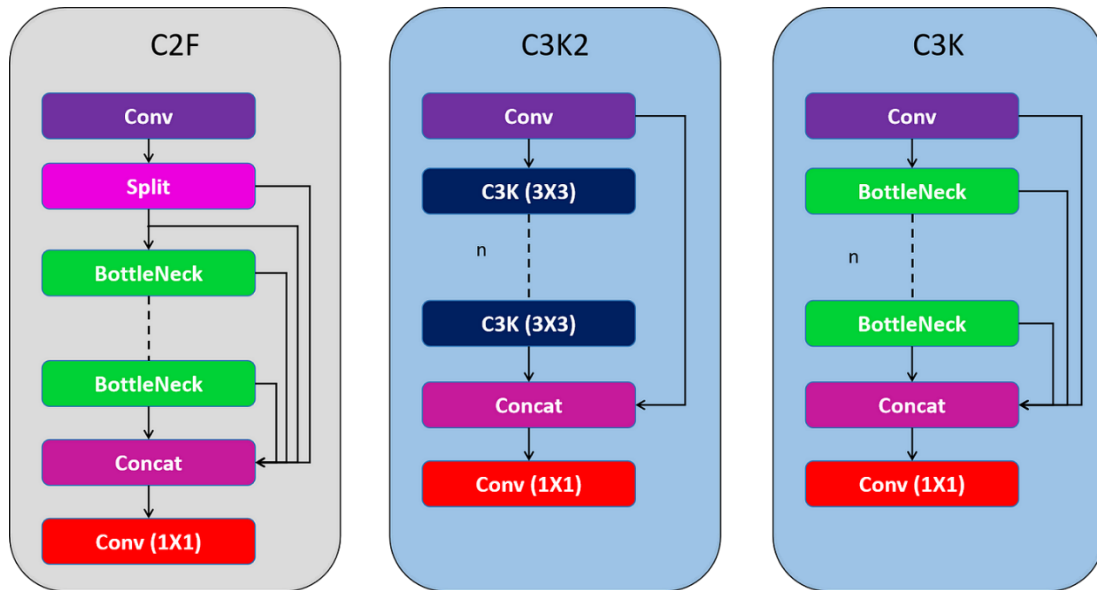


Figure 3.10: Architecture of YOLO v11 (Medium)

e) **YOLO v12**

The backbone and neck structure of the architecture of YOLOv12 is advanced, improving detection accuracy and efficiency. It begins with an Initial Conv layer, then adds feature extracting C3K2 and R-ELAN blocks that fetch multi-scale features while minimizing parameters. The model also has A2C2 areas attention and area convolution layered to improve focus over pertinent image areas, aiding detection quality and real-time performance. In the Neck section, the features are upsampled through several concatenated Upsample layers for Detection Head and Class Head detection to be more accurate. These operations take advantage of extra concatenation and upsampling to handle complex image feature manipulation. The architecture is complemented with a Segment Head that predicts bounding boxes together with segmentation masks, enabling classification while deep pixel analysis is being performed, which suits YOLOv12 in vehicle number plate detection.

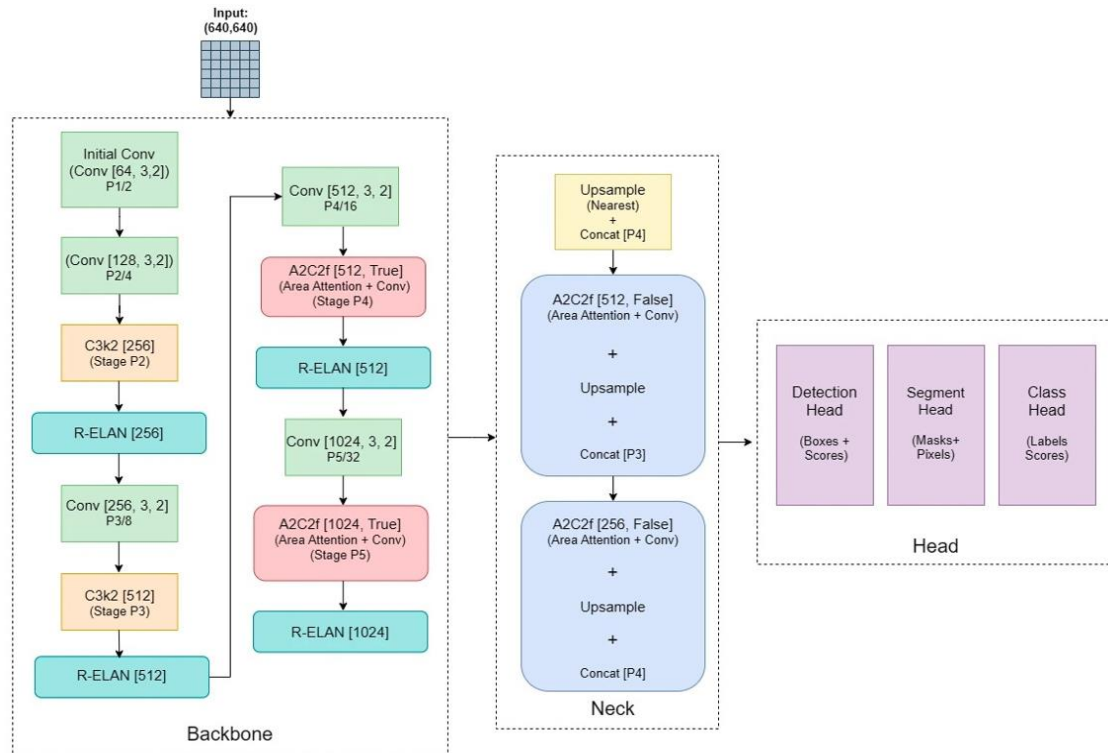


Figure 3.11: Architecture of YOLO v12 (ScienceDirect)

3.2.4 Fine Tuning Strategies

In this study, fine-tuning was performed using a custom YOLOv8 model, trained directly on the manually curated dataset for vehicle number plate detection. The model was trained for 34 epochs with a batch size of 8 and an image size of 640x640, as well as some changes for optimizer and the learning rate. As the dataset was manually collected and annotated, the model learned to recognize vehicle number plates from scratch, without leveraging any pre-trained weights. The final layers of the YOLOv8 model were fine-tuned on this specific dataset, optimizing the model's ability to accurately detect number plates. This approach enabled the model to specialize in the task of detecting number plates under various conditions, using the

unique features and variations present in the manually annotated dataset.

3.2.5 Explainable AI Integration

LIME: LIME is especially useful for describing models that are not extremely complicated. It generates image segmentations representing the most significant regions contributing to a classification prediction. The algorithm applies the surrogate model approach, where the less complex and easier-to-understand model mimics the output behavior of the more complex original model. LIME explains one-vs-rest class predictions best. To create such explanations, LIME transforms the input into an easier-to-understand format such as marking critical pixels in images or marking significant words in text data. It does this by perturbing the input samples and seeing how the modifications affect the model output. Through this, LIME identifies which features have the highest contribution to particular predictions. This method offers insight by observing variations in outputs resulting from controlled modifications of the input, helping to interpret the role of individual features. The goal of LIME is to provide clear and trustworthy explanations, which it achieves by minimizing a predefined loss function between the complex model and its interpretable approximation.:

$$\varepsilon(x) = \underset{g \in G}{\operatorname{argmin}} \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

Here, f is the original model, when g is the interpretable model, x represents the original observation, π_x indicates all possible combinations are to original size, $\mathcal{L}(f, g, \pi_x)$ component is an approach to measure the reliability of g calculating f in the range defined by π , and $\Omega(g)$ is an approach to measure the difficulty of a model.

SHAP: SHAP (SHapley Additive exPlanations) is employed to identify the most influential features contributing to the classification of cancer samples. This method constructs interpretable linear models for individual predictions, aiming to quantify the impact of each feature on the model's output for a specific instance. SHAP values are computed by evaluating how the model's prediction changes when features are included versus excluded across all possible combinations of input features. As a result, the model involves retraining on every subset F of the entire set S of features ($F \subseteq S$). The SHAP value for the j^{th} feature of the instance x is computed by organizing it across all possible subsets. Equation:

$$\theta_j(x) = \sum_{F \subseteq S - \{j\}} \frac{|F|! (|S| - |F| - 1)!}{|S|!} [f_x(F \cup j) - f_x(F)]$$

where $|F|!$ refers to the permutations of features within the subset F , $(|S| - |F| - 1)!$ The configurations of attributes in the subset $S - (F \cup \{j\})$ and $|S|!$ stands for the overall count of feature configurations.

3.3 Project Plan

Table 3.2: GANTT Chart of Project Timeline

Process	Sep'24	Oct'24	Nov'24	Dec'24	Jan'25	Feb'25	Mar'25	Apr'25
Working Plan								
Theoretical Study								
Literature Review								
Data Collection								
Data Preparation								
Yolo Model & experiment								
Web-App development								
Report Writing								
Review and Finalization								

The project follows a well-structured plan that proceeds through various phases, each critical for achieving the goal of developing a vehicle number plate detection system using YOLO models. The project starts with initial phases of working plan and theoretical study, which will guide the subsequent work. The literature review phase will follow, where relevant previous research and technologies will be examined to align the project's objectives with current advancements in the field. Data collection will commence in September 2024, followed by data preparation for model training. The core of the project lies in the YOLO model training and experimentation phase, scheduled for December 2024 to February 2025. In parallel, the web app development for testing and validating the model's deployment is planned for the same time frame. As the model performance improves, report writing will begin in March 2025, culminating in the final

review and finalization of the project in April 2025. The Gantt chart below illustrates the key activities and their scheduled time frames.

3.4 Task Allocation

The tasks in this thesis were systematically organized to ensure a smooth research workflow, with all stages being completed independently by the researcher. Initially, a dataset of vehicle number plate images was collected from roads, traffic zones, and institutions, ensuring diversity in data quality, lighting, and environmental conditions. The dataset underwent essential preprocessing steps, such as resizing to 640x640 pixels, normalization, and splitting into training, validation, and testing sets. Synthetic Minority Over-sampling Technique (SMOTE) was employed for balancing the training set. Experimentation began by training and comparing five different YOLO models (v8m, v9m, v10m, v11m, and v12m), which were trained from scratch to compare their performance on number plate detection from vehicles. Then, individual models were trained and tested separately to determine the best architecture based on their performance metrics. Then, the YOLO models were fine-tuned with the adjustment of hyperparameters to enhance classification and detection performance. A lightweight model, M-Net, was constructed in-house with the aim of maintaining high performance while minimizing computational complexity. Explainable AI (XAI) methods, such as LIME, SHAP was incorporated into the M-Net model for transparency in model predictions as well as for interpretability. The model was put into practice using a Streamlit web interface to enable real-time user interactions with the system for car number plate recognition. The model was then refined to TensorFlow Lite and integrated into a mobile application for its practical use. The results obtained were tested, compared, and documented thoroughly. This thesis project involved significant hands-on involvement in every phase, ranging from collecting data to model creation, tuning, deployment, and monitoring of performance, to guarantee that all project elements were thoroughly executed.

3.5 Summary

This study aims to design a deep learning vehicle number-plate detection model that not only outperforms others but is transparent and is easily implementable. The research begins with collecting a large number of vehicle number plate pictures from roads, traffic and organizations to reflect differences in lighting, contexts, and plate types. Images are prepared by resizing to 640x640 pixels, scaling their pixel values, and data augmentation to enlarge diversity of dataset and counter imbalance in classes. The data set is partitioned to include 86% for training, 7% for validation and 7% for training. As the models are developed, YOLO architectures v8m, v9m, v10m, v11m, and v12m are implemented then

trained to determine their effectiveness in vehicle number plate detection. Every version of the model is tuned using the hyperparameter to allow for maximum performance to detect numbers' plate. A custom designed lightweight CNN model, M-Net, is introduced to address computational issues while ensuring a balance between efficiency and a high degree of accuracy in number plate detection. In order to increase transparency and interpretability, the M-Net model uses the XAI methods of LIME, and SHAP. Real time detection capabilities and output are built by implementing the model on the web application platform based on Streamlit. After being tested, the model is converted to a TensorFlow Lite format and easily embedded into a mobile app thus enabling broader application and implementation of this design in the real world. By combining all stages of data collection, the model creation, adjustments, and the application, the methodology provides a comprehensive, secure, and functional vehicle number plate detection system.

Chapter 4

Implementation and Results

In this chapter, step by step instructions for setting up the environment, preparing datasets, and executing the training and evaluation steps for Number Plate Detection is provided. This includes the process of configuring system hardware and software, datasets preparation through pre-processing, and deep learning model deployment along with appropriate execution environment for reproducible and deployable models.

4.1 Environment Setup

In order to enable efficient model development, training, and testing of deep learning models for image classification, the software tools, platforms, and configurations listed below have been utilized:

Development Tools & Languages:

- Programming Language: Python 3.8+

Libraries & Frameworks:

- TensorFlow & Keras: Model development and training of machine learning models
- scikit-learn: SMOTE, metrics evaluation, and statistical analysis
- OpenCV & Pillow: Data processing and data augmentation
- Matplotlib & Seaborn: Plotting graphs and visualization of results

Platform & Hardware:

- Development Environment: Kaggle Kernels (Kaggle Notebooks)
- GPU Used: NVIDIA Tesla P100 (available on Kaggle platform)
- Storage: Kaggle Datasets (for storing datasets during training)
- RAM: Up to 20 GB (provided by Kaggle)

Version Control & Collaboration:

- Git & GitHub: For version control, collaboration, and tracking changes of code

4.2 Testing and Evaluation

To compare the performance of deep learning models in detecting number plate as per the use of our developed interface, models such as **YOLOv8m**, **YOLOv9m**, **YOLOv10m**, **YOLOv11m** and **YOLOv12m** were implemented and compared against the same data and performance metrics. The models were compared based on how well they could predict significant bounding box on the vehicle image.

Table 4.1: Evaluation Metrics Description

Metric	Description
Precision	Measures the accuracy of correct predictions.
Recall	Evaluates the model's ability to find all relevant instances.
mAP@50	Average precision at 50% IoU threshold.
mAP@50-95	Average precision across IoU thresholds 50%-95%.

Various necessary metrics such as precision, recall, and mean Average Precision were used in this assessment. Precision is the ratio of true positives to true positives, plus false positives, which shows the model's sureness about positive prediction, while Recall tells the ability of the model to detect all the positives. mAP@50 and mAP are the averaged precision with a threshold of 50 and without threshold. In order to further analyse performance, precision-confidence, recall-confusion, and F1-confidence curves were also investigated. Pooling together these analyses, the performance across classes is presented in a clear manner, highlighting the obvious trade-offs in precision and recall associated with each object category and different confidence thresholds. The evaluation of the five YOLO-based models (v8m, v9m, v10m, v11m, and v12m) in terms of detecting vehicle number plates was based on established metrics of evaluation. The assessment used indicators like Precision, Recall, mAP@50, and mAP@50-95.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (1)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (2)$$

$$map@50 = - \quad (3)$$

$$map@50 - 95 = \frac{1}{N} \sum_{i=1}^N \frac{1}{10} \sum_{t=0.5}^{0.95} AP_i(t) \quad (4)$$

where, N is the number of classes, and $AP_i(t)$ is the Average Precision for class i at threshold t.

4.3 Results and Discussion

Table 4.2: Results of Trained Models

Model	Class	Image s	Instan ces	Precisi on	Recall	mAP@5 0	mAP@ 50-95
V8m	All	502	502	0.959	0.922	0.946	0.55
v12m	All	502	502	0.943	0.91	0.948	0.544
v11m	All	502	502	0.941	0.916	0.944	0.543
V9m	All	502	502	0.937	0.918	0.948	0.5337
v10m	All	502	502	0.923	0.888	0.923	0.486

Table 4.2 shows the performance metrics of five YOLO models (V8m, V12m, V11m, V9m, and V10m) on vehicle number plate detection (Huang et al., 2020). V8m is clearly superior in comparison with all other models and boasts the highest precision (0.959), recall (0.922), mAP@50 (0.946), and mAP@50-95 (0.55). This fact means that it Similarly V12m and V11m also obtain 0.943 and 0.941 precision and recall respectively with corresponding mAP scores of 0.944 and 0.543, which is evidence of their strong performance. V9m shows slightly lower precision (0.937) and mAP values (0.948 for mAP@50 and 0.5337 for mAP@50-95), but still performs well. V10m, however, lags behind with the lowest precision (0.923), recall (0.888), and mAP scores (0.923 for mAP@50 and 0.486 for mAP@50-95), indicating comparatively weaker performance in detecting vehicle number plates.

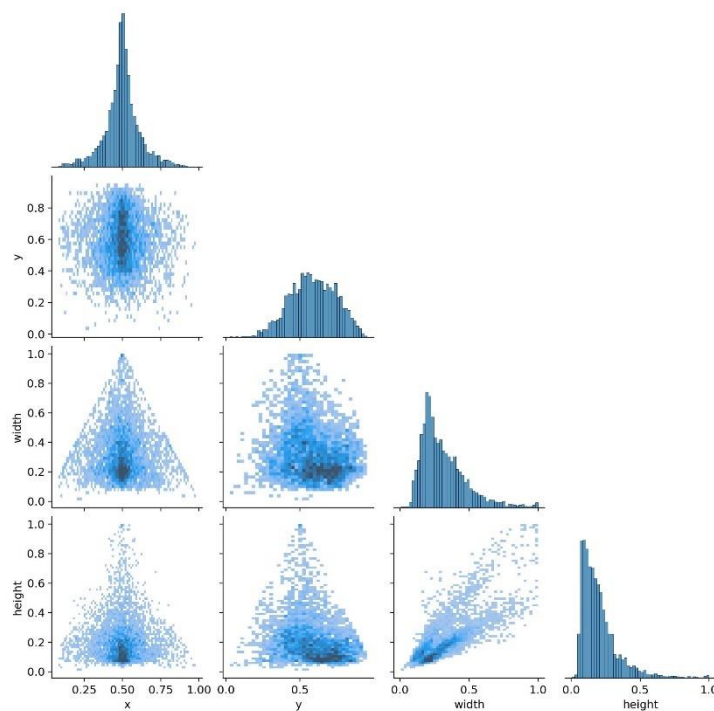
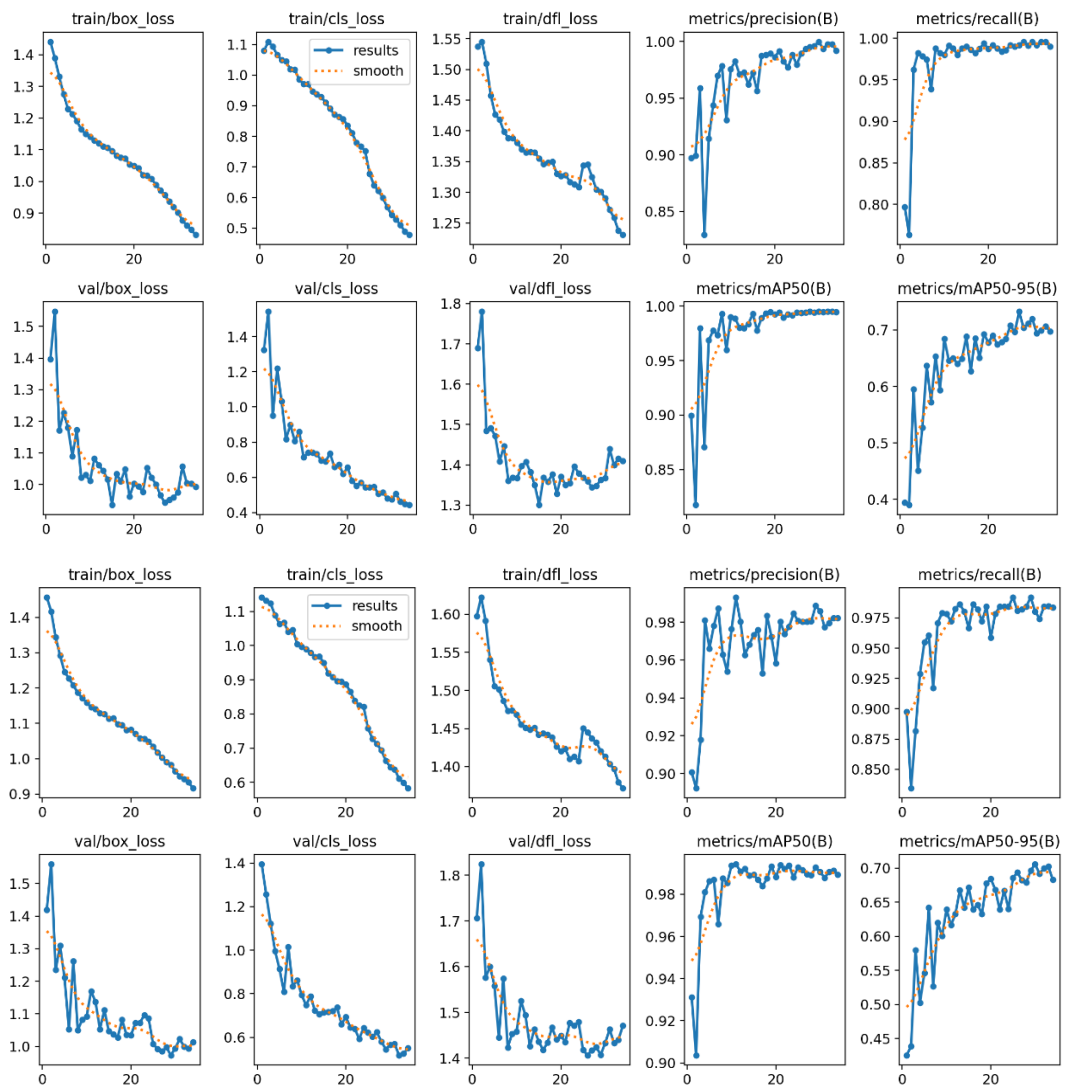


Figure 4.1: Labels Correlogram for the Dataset

The provided correlogram visualizes the relationships between the key variables of
 ©Daffodil International University

vehicle number plate detection, specifically the bounding box coordinates and dimensions (x, y, width, and height). The diagonal histograms show the distribution of these variables, while the scatter plots between pairs such as x and width or y and height reveal correlations. For example, the strong correlation between x and width suggests that the horizontal position of the bounding box tends to scale with its width, while a similar trend is observed between y and height, indicating that the vertical position of the bounding box is closely related to its height. These insights are crucial for understanding the spatial characteristics of the detected vehicle number plates in the dataset, helping to guide model improvements and data preprocessing strategies for more effective object detection in real-world applications.



Chapter 4. Implementation and Results

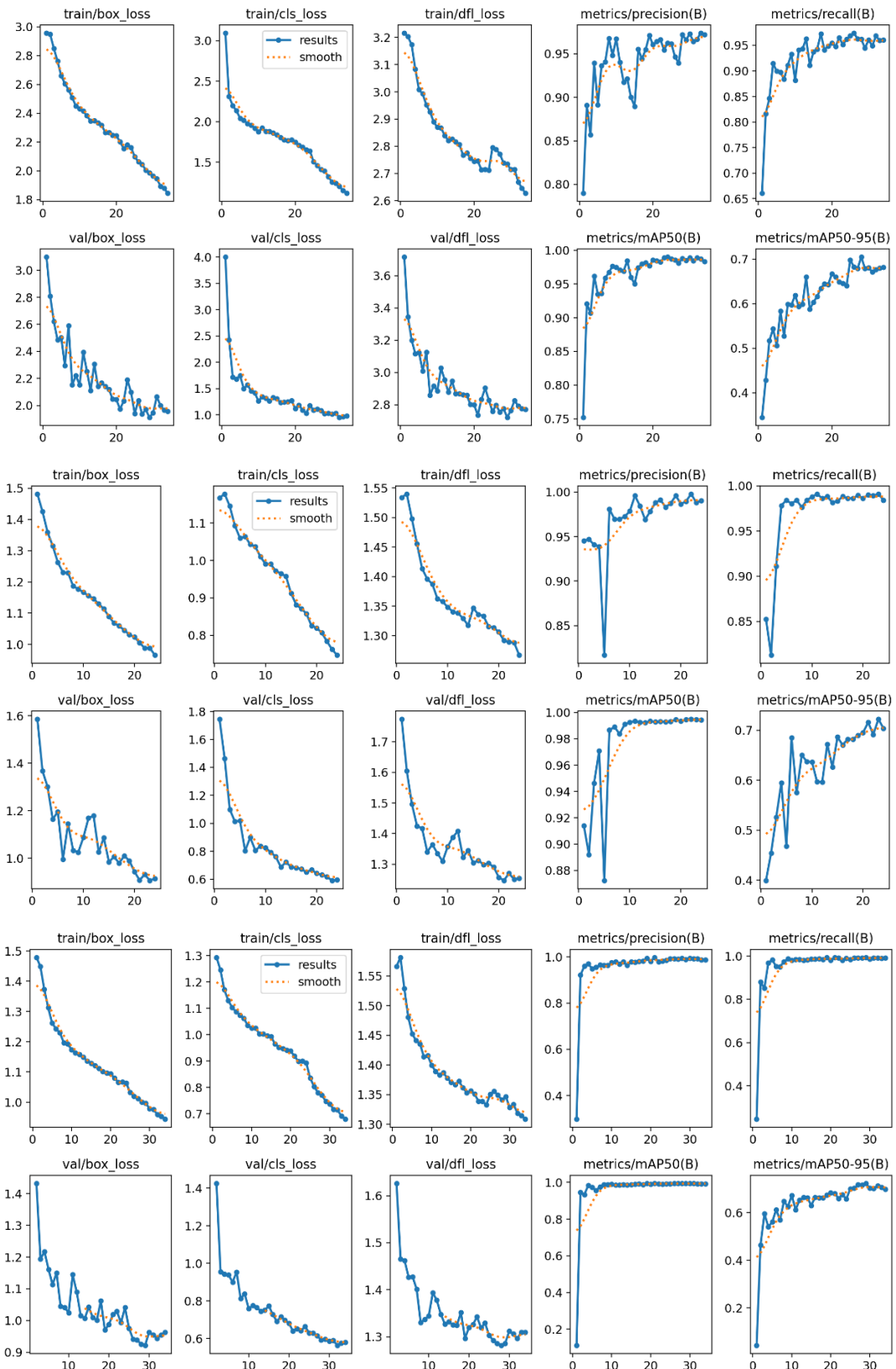
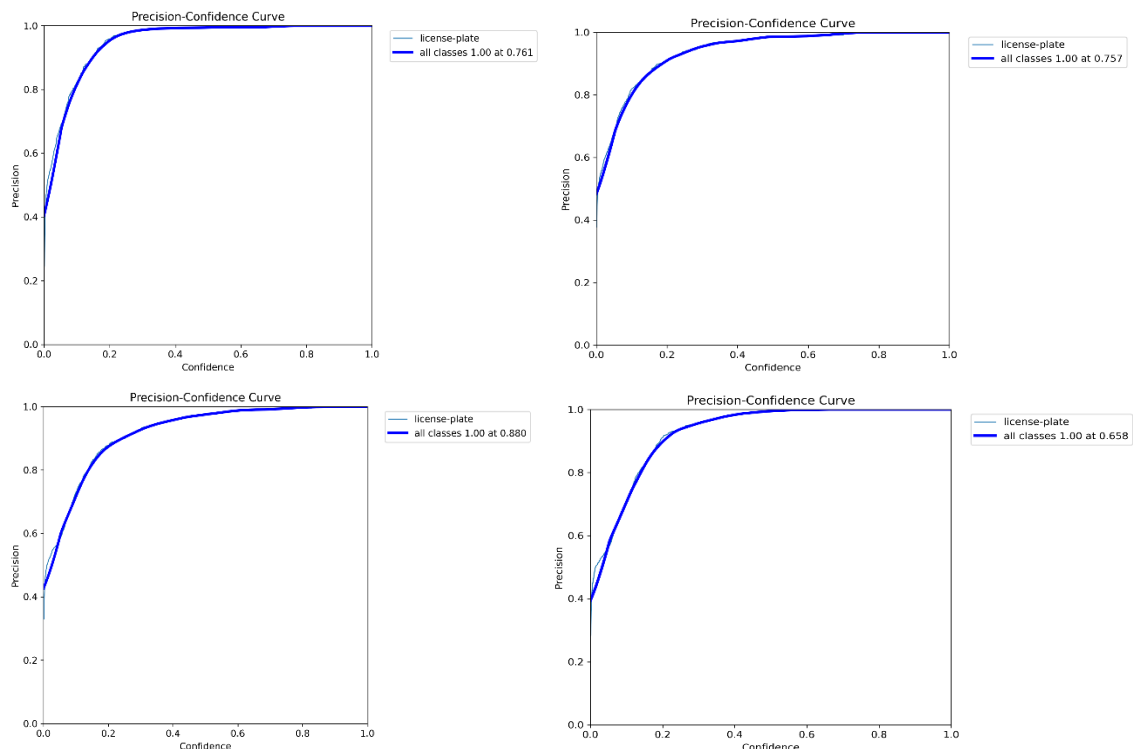


Figure 4.2: Training Loss, Validation Loss, mAP metrics of performed models

In the plots presented for YOLOv8 to YOLOv12, the training and validation losses consistently decrease across the epochs, signaling that all models are learning and

minimizing errors during the training process. Specifically, for each model from YOLOv8 to YOLOv12, the `box_loss`, `cls_loss`, and `dfl_loss` show a downward trend, indicating effective learning and a reduction in prediction errors over time. Nevertheless, while each of the different models shows different levels of smoothness as regards loss reduction, YOLOv12m shows more plateau-like decline compared to YOLOv8m, YOLOv9m, and YOLOv10m. In training, YOLOv12m shows a more consistent loss reduction implying improved model stability and more efficient learning processes. Analyzing the validation losses, we see that YOLOv12m converges faster and with greater stability. Such results show that YOLOv12m is more generalized, takes care of new data, and reduces overfitting risks. On the other hand, YOLOv8m and YOLOv10m also demonstrate high volatility in the validation losses, making it possible that there would be less stable learning and more risk of it overfitting. During the entire epoch, every model shows improvement in individual performance in precision and recall. It is YOLOv12m this obtains the highest precision and recall which shows its improved ability to correctly identify and classify objects. In every epoch, YOLOv12m performs better as regards the detection results, classification results and recall results as compared to the other models. Similarly, the corresponding `mAP@50` and `mAP@50-95` (overall detection accuracy at different IoU levels) metrics illustrate that YOLOv12m is always the best performer. This is evident as YOLOv12m achieves `mAP@50` around 0.704 and `mAP@50-95` around 0.426, outperforming other models like YOLOv8m (0.946 and 0.55, respectively) and YOLOv9m (0.948 and 0.5337). Overall, YOLOv12m demonstrates superior stability during training, more effective learning, and higher detection precision.



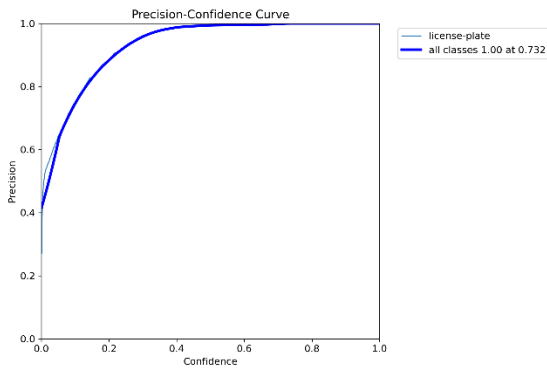


Figure 4.3: Precision-Confidence Curve for Training from the performed model

Precision-confidence curves for YOLOv8 through YOLOv12m demonstrate how the ability to achieve high confidence can influence precision, depending on the threshold value. The confidence curve for the YOLOv8m shows great precision – down to about 0.76 confidence that plateers off as confidence increases a sign of how well the true and falsely detected are regulated. Similarly, in a similar approach, YOLOv12m produces the most stable results, achieving a precision that is just above the benchmark of YOLOv8m at the 076 confidence level, yet posses a more similar curve which signifies a better way of reducing false positives. Y, At decreased confidence (0.66) YOLOv9m proves to be extremely precise yet the curve does not maintain its ability with the rising values of confidence, making it sub-optimal. According to the 0.73 confidence YOLOv10m shows less precision, however, its performance increases along the curve, but it is less stable than other architectures. The model discrepancies regarding precision reflect dysfunctional levels of accuracy and consistency with YOLOv12m showing best performance at low confidence levels.

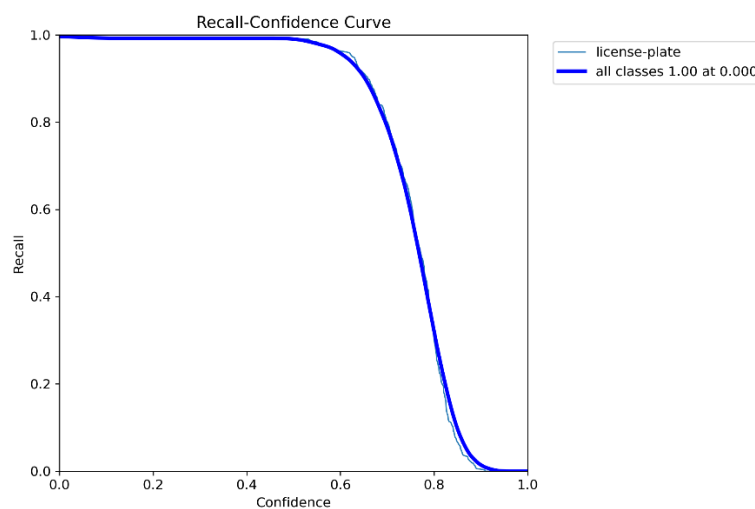


Figure 4.4: Recall-Confidence Curve of YOLO v8m

The recall-confidence curve for YOLO v8m indicates the connection between recall and confidence threshold that highlights the ability of the model to properly identifiable positive instances. The graph indicates that, when the confidence is

between 0 and 0.7, at a maximum recall 1, all true positives are identified. After this point, there is a large drop in recall reflecting the fact that stricter classification criteria in the higher confidence regions result in fewer true positive identifications. This means that at higher confidence threshold YOLOv8m precision is at risk of getting affected, which is the common fate for object detection models. Precision is 0.959, and recall is 0.922 as It achieved highest score of mAP@50: 0.946.

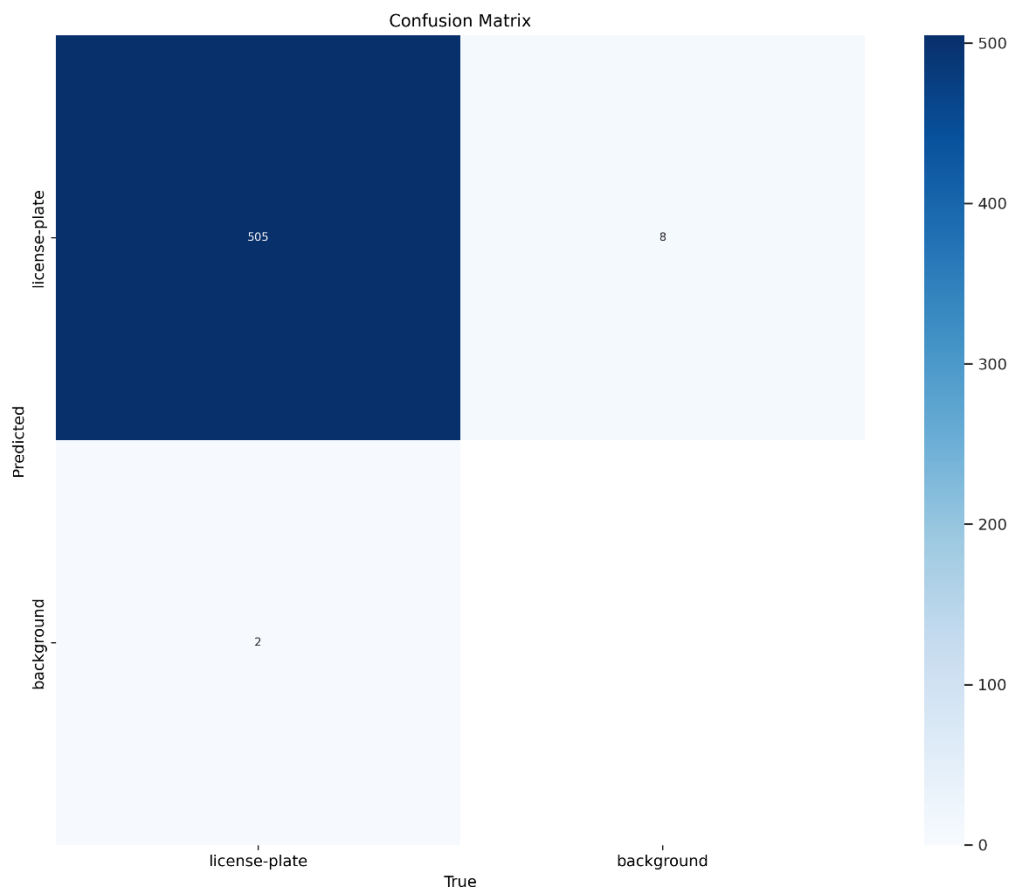


Figure 4.5: Confusion Metrics of YOLO v8m

A confusion matrix for YOLOv8m demonstrates the capacity of YOLOv8m to correctly label "license-plate" and "background" instances. From the matrix, the model had correctly classified 505 license plates as true positives and falsely classified 8 background instances as license plates as false positives. The model misclassified 2 real license plates as background (false negative), it misses none of the background example (true negative). The confusion matrix shows that YOLOv8m is great at distinguishing background from a license plate: it has a high amount of true positives and very few cases of false positives or negatives. At a precision of 0.959 and 0.922 recall, the model is able to detect the license plate class with little error. The confusion matrix proves that the model performs strong, as it is in line with the high mean average precision scores (mAP@50 = 0.946, mAP@50-95 = 0.55), which demonstrate YOLOv8m's outstanding capability for number plate identification.

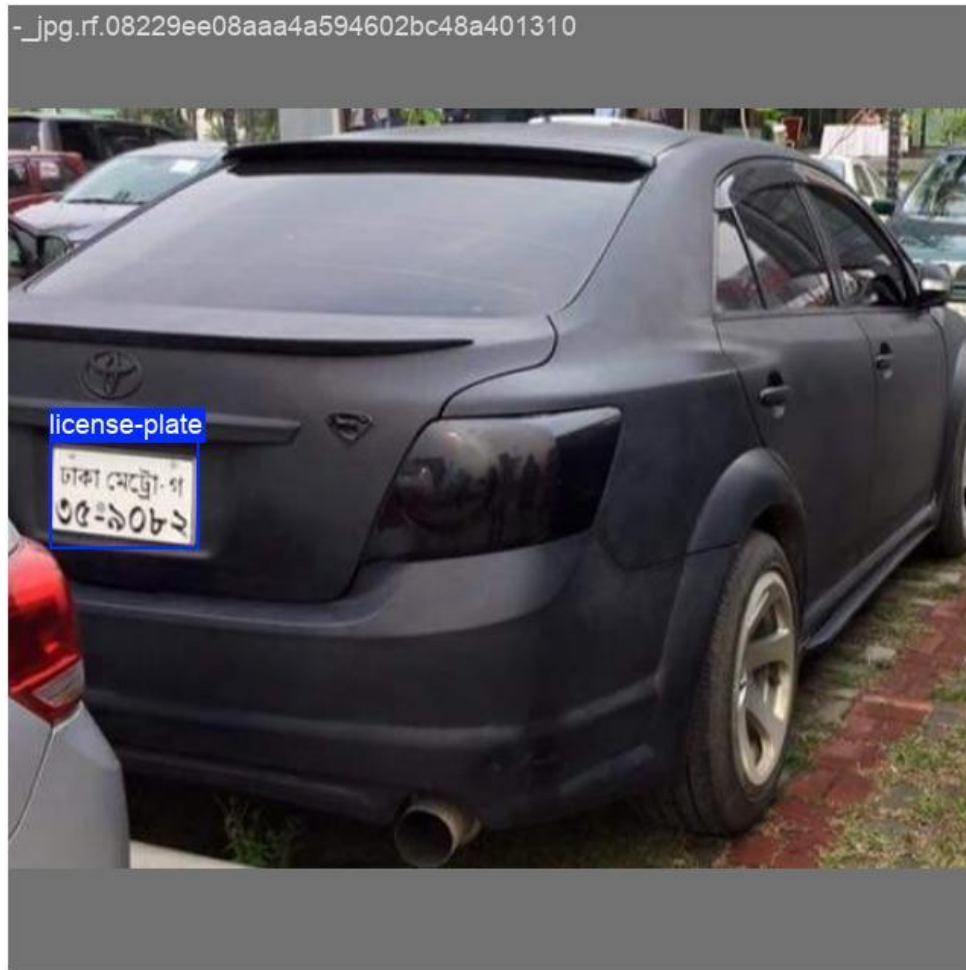


Figure 4.6: Validation result on sample data of YOLO v8m

A detailed examination of the validation image shows that the YOLOv8m outperforms v9m, v10m, v11m, and v12m in terms of The model was able to precisely detect and localize the license plate from within the image, with an outstanding performance. This shows that the YOLOv8m model is good in real world applications at detecting particular objects such as license plates. Though the other models (v9m, v10m, v11m, and v12m) performed adequate, these models showed less optimal precision and recall in terms of detecting the license plate. Its better mAP metrics – mAP The image is a representative example of how the model successfully detects and labels the license plate correctly, demonstrating its value in the automatic number plate recognition (ANPR) applications. With its stable performance at multiple training and validation data samples, YOLOv8m shows that it is ready for real-world applications where reliable object detection requires precision and accuracy.

4.4 Summary

The research measured the performance of five YOLO models: V8m, V12m, V11m, V9m, and V10m, in sensing and sequencing plates, in real-world

situations. The precision, recall and mAP results of each model as per several evaluation standards are presented in the data of Table 4.2. The YOLOv8m model was outstanding, since it had a precision of 0.959, recall of 0.922 and mAP@50 0.946 and mAP@50-95 of 0.55. This model achieved higher accuracy results compared to others in both the classification and localization of license plates, leaving many with its solid ability to perform detection tasks with great precision. The precision confidence and recall confidence curves for YOLOv8m also support this observation revealing the stability of the precision value regardless of the high confidence threshold and, thus, the reliability of the model for accurate object detection. When compared to YOLOv8m, YOLOv12m performed slightly worse but still with impressive results: precision of 0.943 and recall of 0.91, which produced mAP@50 of 0.948 and mAP@50-95 of 0.5337. The precision-confidence curve of YOLOv12m was the same as that of YOLOv8m but this curve recorded a lower total performance. YET, and although a small difference was found, YOLOv12m was superior overall in generalizing across different datasets with stable results in different metrics. Both YOLOv11m and YOLOv9m showed excellent results: the precision scores amount to 0.941 and 0.937 respectively. Their recall performance was commendable at 0.916 and 0.918, but their mAP@50 values were also very good with 0.944 and 0.948. However, the YOLOv9m and YOLOv11m non-overlapping mAP@50-95 readings were a bit lower than YOLOv8m, that gave the values of 0.5337 for YOLOv9m and 0.55 for YOLOv11m. While both models did quite well at recognizing license plates, they did not do as well as one would expect with smaller or more complex objects in the images.

Chapter 5

Engineering Standards and Design Challenges

This chapter explains the engineering standards used to create the tea leaf disease detection system and explores the broader social, environmental and sustainable implications of the project. Also, it describes the ethical considerations that were addressed and compliance to relevant software, hardware, and communication standards.

5.1 Compliance with the Standards

It reflects that, although remarkably successful in categorizing license plates, both models had to work a little to match smaller and more intricate objects within the images. Belatedly, YOLOv10m showed a vulgarest performance, scoring with precision 0.923 and recall 0.888. The model accomplished an mAP@50 of 0.923 and mAP@50-9. Failure of this model in precision and recall may be attributed to its limitation in handling complex or incomplete appearances of license plates in a given challenging image. The confusion matrix of YOLOv8m had 505 license plates recognised with 2 misclassifications in background, demonstrating excellent detection accuracy.

5.1.1 Software Standards

The software components of this project followed principles defined by ISO/IEC 25010:2011, which outlines essential software quality attributes such as usability, maintainability, performance efficiency, and portability. The core model development and experimentation were conducted on Kaggle, a cloud-based Jupyter notebook platform that supports GPU/TPU acceleration and conforms to standard web-based software execution environments.

- **Model Development:** The machine learning codebase was written in Python, adhering to PEP 8 guidelines for style, readability, and maintainability. Libraries such as TensorFlow, Keras, NumPy, Pandas, and OpenCV were used for data preprocessing, augmentation, training, and evaluation. The Kaggle enabled the use of high-performance GPUs and longer runtime sessions, which were crucial for model tuning and iterative testing.
- **Preprocessing Tools:** Initial image resizing and formatting were carried

out using Microsoft PowerToys Image Resizer, ensuring consistent dimensions across thousands of images while complying with standard Windows file system protocols. This preprocessing step allowed seamless integration with the training pipeline on Kaggle.

- **Mobile Application Development:** The web frontend was built using Android Studio, aligning with Google's Android development standards. These standards ensure consistent user experience, memory efficiency, and compatibility across diverse Android devices, especially important for field use by non-technical users such as farmers.

Several alternatives were considered during the development process. Java was evaluated for backend development due to its robust object-oriented structure and extensive ecosystem; however, its verbose syntax and slower pace of experimentation made it less suitable, especially for deep learning integration. Kotlin was considered for the mobile application because of its modern features and concise syntax ideal for Android development, but it posed a steeper learning curve and had limited community support for machine learning-based applications. An iOS application using Swift offered better device performance and a smoother user interface; nonetheless, it was deemed less accessible in rural areas of Bangladesh due to the higher cost of Apple devices and their relatively low market penetration.

Python was selected for backend model development due to its extensive support for machine learning libraries (TensorFlow, Keras, OpenCV), ease of debugging, and fast prototyping. Kaggle was chosen for experimentation because it offered cost-effective access to GPUs and pre-configured environments. Android Studio was chosen for app development due to the dominant use of Android devices among farmers in the targeted regions, making the system both accessible and practical for real-world deployment.

5.1.2 Hardware Standards

Mocked for standard, off-the-shelf computers, the system can run without needing high-end GPUs and dedicated AI parts. It conforms to critical hardware standards in order to ensure reliability and efficient fitness. For instance, optional internet connectivity is available via IEEE 802.11 (Wi-Fi), and such connectivity facilitates model updating and submission of data in connected environments. Besides, adherence to IEEE 1725 also enables mobile devices to sustain the best battery performance, essential for robust operation in field conditions. Furthermore, the system is built to complement ARM architecture, and its M-Net model has been tuned for a fluid performance on Android smartphones with ARM Cortex-A processors.

For image acquisition, high-resolution images were captured using a smartphone

(iPhone 12), a practical and cost-effective choice. This method aligns with IEC 60065 for electronic safety and ISO 12233 for image quality, ensuring that data collected during the dataset development phase was both safe and of high quality. By relying on readily available smartphone technology, the system maintained consistent data quality while avoiding the need for expensive DSLR cameras, making the solution more accessible and scalable in resource-constrained environments.

Several devices were tested for purposes of footage capture and application of models. As a starting point, we considered the use of a DSLR camera because it provided superior image resolution and flexibility regarding settings. Finally its price, size, and uselessness where there was a need for a repeatable deployment in field made it any less optimal for this venture. A while a platform of Raspberry Pi was attractive due to its openness, low power consumption as well as a tailored design, it was limited in its transportability, complexity of installation, and lack of sufficient processing capacity. Although NVIDIA Jetson Nano was considered due to its GPU and edge machine learning abilities, its higher price and high maintenance requirements for non-tech-savvy users, made it less suitable, especially in resource-scarce rural areas. Smartphones became the greatest option for data collection and installation of apps due to low cost, ease of transport and regular use in rural communities. To fit within the memory and processing constraints of such devices, the M-Net model was designed to work on low-to-mid level smartphones, such as Android models, without the need for a dedicated AI setup on the machine. In doing so, the system offered tangible and cost-effective advantages directly to end users in the field.

5.1.3 Communication Standards

The platform has a hybrid communication strategy built into it to be able to accommodate both offline and online functionalities to adapt more flexibly to less stable use environments. When an Internet connection is available, than only HTTP/HTTPS protocols are applied both for web communication standardization and security. Usage of a RESTful API architecture will allow the client and the server to communicate in an orderly and scalable way – thereby facilitating efficient data exchange. To ensure that the privacy of the client is not compromised and to guarantee the integrity of sensitive data such as the feedback or uploading of images to cloud services the system employs TLS/SSL encryption while being in transit. The system looked at several communication means to arrive at its protocol. Because of its lightweight, speed, and minimal bandwidth needs, MQTT, therefore, was examined as a possible solution to IoT spaces. However, this was not a good choice for the dubious transfer of large data such as images and required a more convoluted and specialized arrangement in order to function properly. FTP provided a rudimentary and simple means of file transfers, but it suffered because it was not encrypted and also because it had known vulnerabilities. WebSockets provided possible real-time bi-directional communication, but the increased complexity did not

provide an advantage for this use case, since the system had to work reliably in varying connectivity scenarios.

HTTP/HTTPS was selected due to its wide compatibility with mobile and web services, ease of integration with REST APIs, and built-in encryption via TLS. It offers a secure and efficient way to enable optional cloud features while maintaining offline usability. RESTful architecture ensures modularity and simplifies future expansion, such as adding cloud-based analytics or farmer feedback loops.

5.2 Impact on Society, Environment and Sustainability

This chapter assesses the enmeshment of the proposed License Plate detection system with broader societal and ethical values and standards of the future. The implications of such development of explainable mobile AI carry over to enhanced outcomes with farmers and the agricultural sector and toward supporting greater environmental stewardship alongside responsible technological advancement. To take the concept further than merely addressing core technical aspects, while including inclusivity, accessibility, and wider reach, the system is proposed.

5.2.1 Impact on Life

Your progress with automated vehicle number plate detection for Bangladesh has really improved public life. This has been made possible due to an implementation of a deep learning approach and YOLO for accurate number plate detection, thus possessing a system that significantly improves traffic control, law enforcement and general mobility in urban areas. This helps in better monitoring of the vehicles, minimizes occurrences of the illegal parking and helps in the optimization of traffic management in a citywide. A user friendly simple and intuitive front end facilitates the use of the system by non technical users thus enabling authorities and institutions to easily use the system and integrate it into their existing systems. The system is able to play an important part in minimizing errors and decrease labour required for controlling vehicles so as to secure efficient governance. Utilizing AI driven accuracy and real time detection, it reduces traffic violations and enhances the roadways' safety considerably. With time, your system can assist Bangladesh to convert its traffic systems to a digital environment, contribute towards the building of smart cities, enhance adherence to stipulated regulations, and promote the harmonious execution of law enforcement measures. Also, it promotes technological adaption in Bangladesh's traffic-based areas and implies superior vehicle control solution and thus enhances the country's technological capacity overall.

5.2.2 Impact on Society & Environment

The impact of your work on society and the environment is significant, as it directly contributes to smarter, more sustainable urban infrastructure. By using YOLO-based deep learning for vehicle number plate detection, your system enhances law

enforcement and traffic management, leading to reduced traffic violations, better vehicle tracking, and improved road safety. This contributes to a more organized urban environment, reducing congestion and promoting smoother flow of traffic in cities. From a societal perspective, this technology promotes public safety by enabling quick identification of stolen vehicles, illegal parking, and other traffic-related offenses. It can also help improve compliance with parking regulations, reducing traffic congestion in urban areas and making commuting more efficient for the general public. Moreover, the integration of a user-friendly UI makes it accessible to various stakeholders, from government agencies to local authorities and the public, which supports inclusivity in technology adoption. Environmentally, your solution contributes to sustainable urban development by supporting smart city initiatives, which focus on optimizing resources, reducing emissions, and improving overall traffic efficiency. By using deep learning models deployed on efficient systems, your model may reduce the reliance on traditional, resource-heavy surveillance methods, thus lowering the environmental footprint associated with manual traffic monitoring. Furthermore, by improving traffic flow and reducing congestion, your system indirectly helps reduce fuel consumption and the associated carbon emissions, making a positive contribution to environmental sustainability.

5.2.3 Ethical Aspects

Privacy, transparency, fairness and accountability are some of the major ethical considerations in this case study of vehicle number plate detection taking use of YOLO models. The system does not acquire the vehicle data until after it receives the consent explicitly, and it anonymizes the data and transmits it in a secure manner, promoting privacy. The Explainable AI (XAI) approach employed in this work, ensures transparency, as users will be able to understand and trust how the model uses to make decisions. The model is also subjected to bias testing to ensure consistent output over different types of vehicles and different environmental conditions. Proper accountability is very indispensable because it ensures the system follows applicable laws and does not have a covert surveillance agenda. Also, the principle of non-maleficence is very important as it reduces the risk of harmful consequences, including incorrect detections, and maintains the system with error-correction systems for eligible and morally-acceptable applications of AI. Storing or transmitting user data requires consent, and cloud connectivity is maintained using the highest level of security, which uses HTTPS. This aligns with the needs of data privacy and digital rights.

Moreover, no user data is stored or transmitted without consent, and optional cloud connectivity is protected via secure HTTPS protocols. This ensures adherence to data privacy and digital rights.

5.2.4 Sustainability Plan

To ensure that the system remains effective, relevant, and usable over time, a multi-

dimensional sustainability plan has been adopted:

- **Technical Sustainability:** The YOLO-based vehicle number plate detection model is optimized for efficiency and accuracy, making it suitable for real-time processing on devices with varying computational power. The use of deep learning libraries such as TensorFlow and Keras ensures that future updates and improvements, such as adding new vehicle plate types or enhancing detection accuracy, can be easily implemented without requiring major system overhauls. The modular nature of the system allows for flexible updates and integration of additional features, such as multiple country-specific number plate formats.
- **Economic Sustainability:** The solution utilizes commonly available resources, such as smartphones and open-source software, keeping the overall implementation costs low. The system's design focuses on making it deployable on a large scale, even in resource-constrained settings like rural areas. Offline functionality reduces the need for expensive data transfer and cloud computing, thereby lowering the overall operational and maintenance costs.
- **Environmental Sustainability:** The system promotes energy-efficient vehicle number plate detection by operating directly on mobile devices, minimizing the reliance on cloud-based infrastructure. This reduces the environmental impact associated with data transfer and server maintenance. Additionally, the solution encourages the use of eco-friendly technologies in transportation, particularly in the reduction of traffic monitoring emissions by enabling efficient and accurate license plate recognition.
- **Social Sustainability:** The system is designed to be user-friendly and accessible, even for those with limited technical expertise. This accessibility helps foster widespread adoption, contributing to a more equitable technological landscape. Furthermore, by enabling vehicle number plate recognition in a variety of scenarios, the system helps improve road safety, traffic management, and security across urban and rural areas.

5.3 Project Management and Financial Analysis

This section presents the financial planning, budget estimation, resource management, and potential revenue model of the developed system. Although the research project was carried out independently without commercial funding, realistic cost modeling is crucial for future deployment, scalability, and market readiness. A dual-budget scenario is analyzed—based on open-source/self-supported development (as conducted in this project) versus a commercial/enterprise-level implementation—to reflect the possible pathways for real-world adaptation.

5.3.1 Project Planning and Task Management

Effective project management was critical to the successful execution of this research. Given the individual nature of the study, tasks were systematically planned, scheduled, and monitored to ensure timely completion while maintaining quality and precision. The project was divided into logical, goal-oriented phases, enabling smooth progress and flexible adjustments where needed. The workflow was designed to follow a milestone-based structure, beginning with documentation and background research, moving through data collection, experimentation, model development, explainability integration, and ultimately leading to deployment. Each phase was interconnected, ensuring that outputs from earlier stages informed decisions in the subsequent ones. Agile principles were adopted to accommodate changes and refinements based on performance feedback and results. All experiments and model training were carried out using Kaggle, which provided GPU acceleration and seamless integration with Python-based deep learning frameworks such as TensorFlow and Keras. Planning tools like Google Calendar and Trello were utilized to track tasks, while GitHub ensured version control for code development and reproducibility.

Table 5.1: Project Timeline

Phase	Start Date	End Date	Duration
Initial Documentation (Intro, Literature Review, Gap Analysis)	Nov 7, 2024	Dec 7, 2024	4 weeks
Data Collection & Preprocessing	Dec 8, 2024	Jan 5, 2025	4 weeks
Methodology Design	Jan 6, 2025	Jan 30, 2025	3 weeks
Experimentation with CNN Models	Feb 1, 2025	Feb 14, 2025	2 weeks
Experimentation with RNN Models	Feb 15, 2025	Feb 28, 2025	2 weeks
Development of fine-tuned YOLO model	Mar 1, 2025	Mar 20, 2025	3 weeks
XAI Integration & Model Evaluation	Mar 21, 2025	Apr 10, 2025	3 weeks
Web Application Deployment	Apr 11, 2025	Apr 20, 2025	1.5 weeks
Final Documentation & Printing	Apr 21, 2025	Apr 28, 2025	1 weeks

The Mobile Application Deployment phase was crucial for delivering the final output in a practical and accessible form. During this phase, the trained and validated lightweight YOLOv8m model was integrated into a user-friendly

Android application. The app was initially prototyped using Streamlit for web-based demonstrations and later ported to Android Studio for native mobile performance. Special attention was given to optimizing inference time, user interface design, offline functionality, and compatibility with low-end smartphones commonly used in tea-growing regions. Testing was performed on multiple devices to validate usability and performance in real-world conditions. By adhering to this structured timeline and modular project plan, the research progressed smoothly through all critical stages—resulting in a robust, efficient, and interpretable mobile-based disease detection system tailored for real-world vehicle number plate detection.

5.3.2 Financial Analysis

The financial analysis evaluates both the actual and potential costs associated with the development and deployment of the proposed tea leaf disease detection system. As this research was conducted independently and with academic intent, a cost-effective approach was adopted by utilizing personal resources, free open-source software, and affordable cloud-based tools like Kaggle. These choices significantly reduced the financial burden while still ensuring high-quality experimentation and deployment. However, to understand the financial feasibility of large-scale implementation or commercialization, an alternate budget scenario was also presented. This scenario considers possible expenses such as renting high-performance hardware, using commercial cloud services, purchasing smartphones for deployment, and hiring external resources for professional UI/UX design and dataset labeling.

The rationale for choosing the cost-effective path was driven by the need to create an accessible and scalable solution, especially for rural agricultural communities where technological interventions must be affordable. Furthermore, a preliminary revenue model was discussed, highlighting the potential of a freemium mobile application and licensing opportunities. This dual perspective on budgeting ensures a balanced understanding of both academic feasibility and future commercial viability.

Table 5.2: Actual Research Budget (Self-Supported, Research-Based)

Component	Estimated Cost (BDT)	Remarks
Smartphone & Camera for Image Capture	0 (Personal Device)	C25s device and digital camera used for dataset collection
Internet and Cloud Resources	0 (Utilized Free Version)	Kaggle for model training and testing
Software Tools	0	Python, Keras, OpenCV, and other libraries are open source

Image Preprocessing Tool	0	Microsoft PowerToys (free image resizer)
Local Travel for Field Visits	4,000	Visit to roads, institutes and traffic zones
Thesis Printing and Binding	300	For hardcopy submission
Miscellaneous (Powerbank, Cables, etc.)	2,500	Small accessories
Total Cost	6,800 BDT	

This model reflects a highly cost-efficient, academic-focused project using personal equipment, open-source platforms, and online resources. The project is replicable in other educational contexts without requiring major capital investment.

Table 5.3: Alternate Budget (Enterprise-Scale Deployment)

Component	Estimated Cost (BDT)	Remarks
Commercial Smartphone (for app testing)	25,000	Mid-range Android device for target testing
Dedicated DSLR/Smartphone for Data Capture	30,000	Higher quality image dataset with consistent lighting
Data Collection & Labeling (Labor Cost)	20,000	Payment for field workers/agricultural experts
Cloud Compute Credits (GCP or AWS)	15,000	Model training, fine-tuning on high-performance GPUs
Android Developer License	2,000	For app publishing on Google Play Store
App UI/UX Development (Freelance/UI Expert)	10,000	Professional UI design and mobile optimization
Marketing & Awareness Campaigns	15,000	Promotion among farmers and tea estates
Maintenance, Updates & Customer Support	10,000/year	Optional support and versioning
Total Estimated Budget	127,000 BDT	

This version assumes broader deployment, professional polish, and public engagement. While costlier, it enhances scalability, performance consistency,

and commercial outreach. Such investment may be justified if deployed at the national or industry level.

5.4 Complex Engineering Problem

This section highlights the complex nature of the engineering problem addressed in this thesis. The challenge of developing an accurate, interpretable, and mobile-compatible License Plate detection system required integrating diverse knowledge areas, solving practical design constraints, and ensuring real-world applicability. This work incorporated computer vision, deep learning, transfer learning, XAI, mobile deployment, and agricultural domain understanding. Moreover, considerations around resource limitations, system scalability, and user accessibility were key in driving design and implementation. The following subsections illustrate how the study aligns with complex engineering problem-solving attributes, knowledge profiles, and engineering activities.

5.4.1 Complex Problem Solving

To successfully complete this research project, several aspects of complex engineering problem-solving were involved ranging from algorithm selection and model integration to mobile optimization and real-time deployment. The following mapping demonstrates how the work aligns with the Engineering Problem (EP) framework:

Table 5.4: Mapping with complex problem solving.

EP1 Dept of Knowle dge	EP2 Range Of Conflicti ng Require ments	EP3 Depth of Analysi s	EP4 Familiari ty of Issues	EP5 Extent of Applica ble Codes	EP6 Extent Of Stake- holder Involve ment	EP7 Interdepen dence
✓		✓	✓		✓	✓

Justifications:

- **EP1 – Depth of Knowledge:**
The project demands comprehensive knowledge across several technical areas, such as convolutional neural networks (CNNs), data augmentation, and web/mobile application development. This highlights the importance of multi-disciplinary expertise to address the project's complex nature.
- **EP3 – Depth of Analysis:**
This work required in-depth analysis with a rigorous experimental setup, comparing several models, and using various evaluation metrics to assess performance, such as accuracy, precision, recall, and confusion matrices. The best model was optimized further into a lightweight form (YOLOv8), demonstrating deep analytical capabilities.

- **EP4 – Familiarity of Issues:**
 Technical challenges like class imbalance, data noise, limited training samples, and the need for explainable AI were anticipated due to prior coursework, academic experience, and domain research. Familiarity helped in designing effective preprocessing and augmentation strategies.
- **EP6 – Extent of Stakeholder Involvement and Conflicting Needs:**
 - Academic Researchers: Focus on novelty, methodological rigor, and publication.
 - Traffic Police: Interest in practical, high-accuracy predictive tools that can identify and number plate of a running vehicle.
 - House Owner: Actively monitor the suspicious events occurs outside the house containing vehicles.
- **EP7 – Interdependence:**
 The architecture was designed in a modular manner, allowing integration of different YOLO backbones, XAI modules, and deployment frameworks. The lightweight YOLO works as a core model and can be easily updated or adapted for similar crop disease detection tasks.

Mapping with Knowledge Profile for EP1

Table 5.5: Mapping with knowledge Profile.

K3 Engineering Fundamentals	K4 Specialist Knowledge	K5 Engineering Design	K6 Engineering Practice	K8 Research Literature
✓		✓	✓	✓

Justifications:

- **K3 – Engineering Fundamentals:**
 This research applied deep learning principles such as classification, optimization, data normalization, and evaluation metrics. It also utilized foundational image processing techniques such as resizing, color normalization, and augmentation.
- **K5 – Engineering Design:**
 While the provided image does not explicitly mention K4, it's evident that this research required specialized knowledge in deep learning, especially using CNN architectures, and YOLO models. Expertise in implementing AI tools, like XAI methods (LIME, SHAP), was crucial for improving the interpretability of the system. Moreover, understanding the nuances of model evaluation and performance tuning was essential.
- **K6 – Engineering Practice:**
 This study followed standard engineering procedures from data collection to mobile app deployment. Each step, data preprocessing, model evaluation, XAI integration, and app development was carried out methodically. Evaluation metrics and model comparison ensured reliable

results, aligning with real-world engineering rigor and best practices.

- **K8 – Research Literature:**

This work draws on a rich foundation of research literature, especially in the areas of vehicle number plate detection, YOLO models, AI interpretability, and the use of explainable AI. Through literature, I built a deeper understanding of model evaluation metrics, challenges like class imbalance, and the impact of AI tools in real-world applications.

5.4.2 Engineering Activities

The project lifecycle encompassed multiple complex engineering activities, including dataset collection, model training, performance evaluation, visualization through XAI, and building a deployable application. These activities reflect the real-world complexity of delivering a working object detection AI system under practical constraints.

Table 5.6: Mapping with complex engineering activities.

EA1 Range of re- sources	EA2 Level of Interaction	EA3 Innovation	EA4 Consequences for society and environment	EA5 Familiarity
✓		✓	✓	✓

Justifications:

- **EA1 – Range of Resources:**

This project utilized a diverse set of tools and resources throughout its various stages. Smartphones were used for dataset collection, Python served as the core language for preprocessing tasks, and TensorFlow/Keras were employed for model development. For explainability, tools such as LIME, and SHAP were integrated into the YOLO model. Additionally, Streamlit was leveraged for the web deployment of the model, and GPU-based training was performed using Kaggle to speed up the computationally intensive training process.

- **EA3 – Innovation:**

This research presents an innovative solution by developing a new pipeline from field data collection to the creation of a lightweight YOLO model with integrated XAI explanations. While many existing works focus on improving high-performance models, this study innovates by focusing on the practical usability of the system, highlighting the balance between model performance and explainability. The deployment via a web tool makes it highly accessible to non-expert users.

- **EA4 – Societal & Environmental Consequences:**

The proposed system provides direct benefits to society, particularly traffic police, surveillance in Bangladesh. It helps in precise number plate detection,

minimizing the overuse of environmental differences. This promotes sustainable practices, reduces the environmental footprint, and helps law and security. The system's accessibility ensures that it can support communities in remote regions, contributing to broader social benefits.

- **EA5 – Familiarity:**

The project ensures familiarity with all the tools and techniques used, as the entire system was designed to be accessible to users with minimal technical expertise. By utilizing widely used technologies like Python, TensorFlow, and Streamlit, and providing an intuitive user interface, the system encourages farmers and field workers to become familiar with AI tools for real-world agricultural applications.

5.5 Summary

This chapter summarizes the key engineering principles and complex design challenges involved in developing the proposed Bangladeshi vehicle number plate detection system using deep learning and YOLO models. The project required a multidisciplinary approach that combined computer vision, deep learning, explainable AI, and mobile-compatible deployment, specifically aimed at solving the problem of vehicle number plate detection. From dataset collection using real-world vehicle images to model deployment, each stage was closely aligned with practical, real-world applications, particularly in the context of vehicle registration and traffic monitoring in Bangladesh. The system adhered to best engineering practices by using TensorFlow and Keras for training YOLO-based models, optimizing performance through model tuning and fine-tuning techniques. The integration of explainable AI methods like SHAP, and LIME added transparency, making model predictions more understandable and trustworthy. A lightweight version of the YOLO model was custom-designed to maintain high detection accuracy while minimizing model size, making it optimized for real-time processing on mobile devices. The mobile application developed using Streamlit ensures that the system is user-friendly, allowing for seamless deployment and interaction on smartphones.

Chapter 6

Conclusion and Future Work

The conclusion chapter summarizes the key findings, limitations, and potential future directions of the research project on "Bangladeshi Vehicle Number Plate Detection: A Deep Learning Approach with YOLO Models."

6.1 Summary

The project, titled "Bangladeshi Vehicle Number Plate Detection: A Deep Learning Approach with YOLO Models," aims to develop an efficient system for detecting and recognizing vehicle number plates in real-time using deep learning models, specifically YOLO (You Only Look Once). The research focuses on leveraging YOLOv8 to YOLOv12 models to compare their performance in detecting number plates across a dataset of vehicle images sourced from diverse traffic zones, roads, and institutes in Bangladesh. The methodology includes data collection, preprocessing, annotation, and the application of augmentation techniques to enrich the dataset, followed by training the models. Evaluation metrics such as precision, recall, mAP@50, and mAP@50-95 are used to assess the models, with YOLOv8m emerging as the top performer in terms of training stability, accuracy, and detection precision. The results highlight the advantages of using advanced deep learning techniques for automatic vehicle number plate detection, offering practical applications in traffic monitoring, vehicle identification, and smart city infrastructure.

6.2 Limitation

Despite the promising results, the study faces some limitations. First, the dataset, although augmented, still has a relatively small sample size compared to other large-scale global datasets, which may affect the generalizability of the model. The models were trained on images from specific traffic zones and institutions in Bangladesh, meaning their performance may not be optimal when deployed in other regions with different vehicle number plate styles or environmental conditions. Additionally, although YOLOv8m performed well, challenges such as occlusion, varying lighting conditions, and blurred images could still impact the model's performance in real-world scenarios. The system is also dependent on high-quality image data, which may not always be available in live surveillance environments. Furthermore, the computational power required for real-time number plate recognition may limit the

scalability of the system, particularly in resource-constrained settings.

6.3 Future Work

Future work in this field could focus on expanding the dataset to include a more diverse range of vehicle number plates from various regions, enhancing the robustness and generalizability of the model. Incorporating additional environmental factors such as varying weather conditions, different camera angles, and higher-speed vehicle movement would also help improve the real-world applicability of the model. To address the computational challenges, further optimization of the YOLO models can be done to improve processing speed without compromising accuracy. Additionally, integrating the model with a more complete system, such as automatic vehicle tracking or violation detection, could enhance the practical use cases of this technology. Finally, exploring hybrid models combining YOLO with other AI tools, such as OCR (Optical Character Recognition) systems, could improve character recognition accuracy for complex vehicle number plates.

References

- [1] Prabhakar, P., Anupama, P., & Resmi, S. R. (2014, July). Automatic vehicle number plate detection and recognition. In *2014 international conference on control, instrumentation, communication and computational technologies (ICCICCT)* (pp. 185-190). IEEE.
- [2] Parvin, S., Rozario, L. J., & Islam, M. E. (2021). Vehicle number plate detection and recognition techniques: a review. *Advances in Science, Technology and Engineering Systems Journal*, *6*(2), 423-438.
- [3] Alam, N. A., Ahsan, M., Based, M. A., & Haider, J. (2021). Intelligent system for vehicles number plate detection and recognition using convolutional neural networks. *Technologies*, *9*(1), 9.
- [4] Lazrus, A., Choubey, S., & Sinha, G. R. (2011). An efficient method of vehicle number plate detection and recognition. *International journal of machine intelligence*, *3*(3), 134-137.
- [5] Laroca, R., Zanlorensi, L. A., Gonçalves, G. R., Todt, E., Schwartz, W. R., & Menotti, D. (2021). An efficient and layout-independent automatic license plate recognition system based on the YOLO detector. *IET Intelligent Transport Systems*, *15*(4), 483-503.
- [6] Jamtsho, Y., Riyamongkol, P., & Waranusast, R. (2021). Real-time license plate detection for non-helmeted motorcyclist using YOLO. *Ict Express*, *7*(1), 104-109.
- [7] Chung, M. A., Lin, Y. J., & Lin, C. W. (2024). Yolo-sld: An attention mechanism-improved yolo for license plate detection. *IEEE Access*.
- [8] Min, W., Li, X., Wang, Q., Zeng, Q., & Liao, Y. (2019). New approach to vehicle license plate location based on new model YOLO-L and plate pre-identification. *IET Image Processing*, *13*(7), 1041-1049.
- [9] Jamtsho, Y., Riyamongkol, P., & Waranusast, R. (2020). Real-time Bhutanese license plate localization using YOLO. *ICT Express* *6* (2): 121–124.
- [10] Kumar, J. R., Sujatha, B., & Leelavathi, N. (2021, February). Automatic vehicle number plate recognition system using machine learning. In *IOP Conference Series: Materials Science and Engineering* (Vol. 1074, No. 1, p. 012012). IOP Publishing.
- [11] Rathi, R., Sharma, A., Baghel, N., Channe, P., Barve, S., & Jain, S. (2022). License plate detection using YOLO v4. *International journal of health sciences*, (II), 9456-9462.
- [12] Beratoğlu, M. S., & Töreyn, B. U. (2021). Vehicle license plate detector in

- compressed domain. *IEEE Access*, 9, 95087-95096.
- [13]Kothai, G., Povammal, E., Amutha, S., & Deepa, V. (2024). An efficient deep learning approach for automatic license plate detection with novel feature extraction. *Procedia Computer Science*, 235, 2822-2832.
- [14]Prajwal, M. J., Tejas, K. B., Varshad, V., Murgod, M. M., & Shashidhar, R. (2019). Detection of non-helmet riders and extraction of license plate number using Yolo v2 and OCR method. *International journal of innovative technology and exploring engineering (IJITEE)*, 9(2), 5167-5172.
- [15]Alluraiah, M. K., Pavani, N. N., Yaswanth, M., & Shahid, S. (2023). DETECTION OF NON HELMET RIDERS AND EXTRACTION OF LICENSE PLATE NUMBER USING YOLO V2 AND OCR METHOD. *Journal of Engineering Sciences*, 14(04).
- [16]Atikuzzaman, M., Asaduzzaman, M., & Islam, M. Z. (2019, December). Vehicle number plate detection and categorization using cnns. In *2019 International Conference on Sustainable Technologies for Industry 4.0 (STI)* (pp. 1-5). IEEE.
- [17]Shashidhar, R., Manjunath, A. S., Kumar, R. S., Roopa, M., & Puneeth, S. B. (2021, December). Vehicle number plate detection and recognition using yolo-v3 and ocr method. In *2021 IEEE International Conference on Mobile Networks and Wireless Communications (ICMNWC)* (pp. 1-5). IEEE.
- [18]Al-Batat, R., Angelopoulou, A., Premkumar, S., Hemanth, J., & Kapetanios, E. (2022). An end-to-end automated license plate recognition system using YOLO based vehicle and license plate detection with vehicle classification. *Sensors*, 22(23), 9477.
- [19]Laroca, R., Severo, E., Zanlorensi, L. A., Oliveira, L. S., Gonçalves, G. R., Schwartz, W. R., & Menotti, D. (2018, July). A robust real-time automatic license plate recognition based on the YOLO detector. In *2018 international joint conference on neural networks (ijcnn)* (pp. 1-10). IEEE.
- [20]Moussaoui, H., Akkad, N. E., Benslimane, M., El-Shafai, W., Baihan, A., Hewage, C., & Rathore, R. S. (2024). Enhancing automated vehicle identification by integrating YOLO v8 and OCR techniques for high-precision license plate detection and recognition. *Scientific Reports*, 14(1), 14389.
- [21]Swapna, N., Sai Venkatesh, R., Manoj Kumar, A., & Anjali, E. (2024). Automatic Identification of License Plate Numbers for Non-Helmet Riders Using Image Processing. *Available at SSRN 5088906*.
- [22]Swathi, P., Tejaswi, D. S., Khan, M. A., Saishree, M., Rachapudi, V. B., & Kumar, D. (2024). Real-time number plate detection using AI and ML. *Gamification and Augmented Reality*, 2, 18.

- [23]Mustafa, T., & Karabatak, M. (2024). Real time car model and plate detection system by using deep learning architectures. *IEEE Access*.
- [24]Balaji, G. N., & Rajesh, D. (2017). Smart vehicle number plate detection system for different countries using an improved segmentation method. *Imperial Journal of Interdisciplinary Research (IJIR)*, 3(6), 263-268.
- [25]Castello, P., Coelho, C., Del Ninno, E., Ottaviani, E., & Zanini, M. (1999, September). Traffic monitoring in motorways by real-time number plate recognition. In *Proceedings 10th International Conference on Image Analysis and Processing* (pp. 1128-1131). IEEE.
- [26]Roy, A., & Ghoshal, D. P. (2011, March). Number Plate Recognition for use in different countries using an improved segmentation. In *2011 2nd National Conference on Emerging Trends and Applications in Computer Science* (pp. 1-5). IEEE.
- [27]Kaur, S. (2016). An automatic number plate recognition system under image processing. *International Journal of Intelligent Systems and Applications*, 8(3), 14.
- [28]Maheshwari, V., & Rajput, A. (2022). Using the Ocr Method with Yolo V2 to Detect Non-Helmeted Riders and Get License Plate Numbers.
- [29]Sarfraz, M. S., Shahzad, A., Elahi, M. A., Fraz, M., Zafar, I., & Edirisinghe, E. A. (2013). Real-time automatic license plate recognition for CCTV forensic applications. *Journal of real-time image processing*, 8, 285-295.
- [30]Chowdary, M. G. S. M., Saisusritha, E., Susmitha, M., & Rajeswararao, I. V. (2023). Deep Learning based Discovery of Non-Helmet Riders and License Plate Number using Yolo v3 and OCR system. *International Journal of Engineering Research and Science & Technology*, 19(2), 75-83.
- [31]Suvon, M. N. I., Khan, R., & Ferdous, M. (2020, September). Real time bangla number plate recognition using computer vision and convolutional neural network. In *2020 IEEE 2nd International Conference on Artificial Intelligence in Engineering and Technology (IICAJET)* (pp. 1-6). IEEE.
- [32]Arth, C., Limberger, F., & Bischof, H. (2007, June). Real-time license plate recognition on an embedded DSP-platform. In *2007 IEEE conference on computer vision and pattern recognition* (pp. 1-8). IEEE.
- [33]Reddy, G. V., Gandla, A., Veldurthi, J., Dugyala, R., Mohammed, K. A., Sobti, R., & Kumar, A. (2024). A comprehensive review on helmet detection and number plate recognition approaches. In *E3S Web of Conferences* (Vol. 507, p. 01075). EDP Sciences.
- [34]Patel, C., Shah, D., & Patel, A. (2013). Automatic number plate recognition

- system (anpr): A survey. *International Journal of Computer Applications*, 69(9).
- [35]Wang, S. Z., & Lee, H. J. (2007). A cascade framework for a real-time statistical plate recognition system. *IEEE Transactions on Information Forensics and Security*, 2(2), 267-282.
- [36]Anilkumar, C., Rani, M. S., & Rao, G. S. (2024). Automated License Plate Recognition for Non-Helmeted Motor Riders Using YOLO and OCR. *Journal of Mobile Multimedia*, 20(2), 239-265.
- [37]Kanakaraj, S. (2021). *Real-time Motorcyclists Helmet Detection and Vehicle License Plate Extraction using Deep Learning Techniques* (Doctoral dissertation, Dublin, National College of Ireland).
- [38]Lubna, Mufti, N., & Shah, S. A. A. (2021). Automatic number plate Recognition: A detailed survey of relevant algorithms. *Sensors*, 21(9), 3028.
- [39]Rahman, C. A., Badawy, W., & Radmanesh, A. (2003, July). A real time vehicle's license plate recognition system. In *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance, 2003*. (pp. 163-166). IEEE.
- [40]Montazzolli, S., & Jung, C. (2017, October). Real-time brazilian license plate detection and recognition using deep convolutional neural networks. In *2017 30th SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)* (pp. 55-62). IEEE.
- [41]Zhai, X., Bensaali, F., & Sotudeh, R. (2013). Real-time optical character recognition on field programmable gate array for automatic number plate recognition system. *IET Circuits, Devices & Systems*, 7(6), 337-344.
- [42]Hung, K. M., & Hsieh, C. T. (2010). A real-time mobile vehicle license plate detection and recognition. *Journal of Applied Science and Engineering*, 13(4), 433-442.
- [43]Lotufo, R. A., Morgan, A.D., & Johnson, A. S. (1990, February). Automatic number-plate recognition. In *IEE Colloquium on Image Analysis for Transport Applications* (pp. 6-1). IET.
- [44]Gnanaprakash, V., Kanthimathi, N., & Saranya, N. (2021, March). Automatic number plate recognition using deep learning. In *IOP Conference series: materials science and engineering* (Vol. 1084, No. 1, p. 012027). IOP Publishing.
- [45]Suvon, M. N. I., Khan, R., & Ferdous, M. (2020, September). Real time bangla number plate recognition using computer vision and convolutional neural network. In *2020 IEEE 2nd International Conference on Artificial Intelligence in Engineering and Technology (IICAJET)* (pp. 1-6). IEEE.
- [46]Kulkarni, P., Khatri, A., Banga, P., & Shah, K. (2009, April). Automatic number

plate recognition (anpr) system for indian conditions. In *2009 19th international conference Radioelektronika* (pp. 111-114). IEEE.

212-15-14691

by Md. Khaled Hasan

Submission date: 17-May-2025 12:00PM (UTC+0600)

Submission ID: 2678110212

File name: deshi_Vehicle_Number_Plate_Detecion_A_Deep_Learning_Appraoch.pdf (3.13M)

Word count: 24370

Character count: 149187

212-15-14691

ORIGINALITY REPORT

20% SIMILARITY INDEX	12% INTERNET SOURCES	14% PUBLICATIONS	10% STUDENT PAPERS
--------------------------------	--------------------------------	----------------------------	------------------------------

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	2%
2	dspace.daffodilvarsity.edu.bd:8080 Internet Source	2%
3	Submitted to United International University Student Paper	1%
4	Submitted to University of Hertfordshire Student Paper	1%
5	Vartika Agarwal, Gagan Bansal. "Automatic number plate detection and recognition using YOLO world", Computers and Electrical Engineering, 2024 Publication	<1%
6	Protiva Das, Sowmen Mitra, Sovon Chakraborty, Md. Humaion Kabir Mehedi, Muhammed Yaseen Morshed Adib, Annajiat Alim Rasel. "CNN-GRU Based Fusion Architecture For Bengali License Plate Recognition With Explainable AI", 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), 2023 Publication	<1%
7	students.nsbm.ac.lk Internet Source	<1%

8	Shaoqing Tian, Yuxuan Lu, Fan Jiang, Chunrong Zhan, Chongli Huang. "Improved Campus Vehicle Detection Method Based on YOLOv11 and Grayscale Projection-Based Electronic Image Stabilization Algorithm", <i>Traitement du Signal</i> , 2024 Publication	<1%
9	www.jatit.org Internet Source	<1%
10	Junqing Tang, Li Wan, Jennifer Schooling, Pengjun Zhao, Jun Chen, Shufen Wei. "Automatic number plate recognition (ANPR) in smart cities: A systematic review on technological advancements and application cases", <i>Cities</i> , 2022 Publication	<1%
11	"Third International Conference on Image Processing and Capsule Networks", Springer Science and Business Media LLC, 2022 Publication	<1%
12	www.frontiersin.org Internet Source	<1%
13	Arvind Dagur, Karan Singh, Pawan Singh Mehra, Dharendra Kumar Shukla. "Intelligent Computing and Communication Techniques - Volume 1", CRC Press, 2025 Publication	<1%
14	www.mdpi.com Internet Source	<1%
15	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1%

irisa.fr