# Comparative Performance Study of YOLO Models in Bangladeshi Number Plate Detection

**By**
**Abdullah All Kafi**
**211-15-4040**

**FINAL YEAR DESIGN PROJECT REPORT**

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

**Supervised by**
**Shah Md. Tanvir Siddiquee**
**Assistant Professor**
Department of Computer Science and EngineeringDaffodil International University

**Co-Supervised by**
**Dr. Abdus Sattar**
**Associate Professor**
Department of Computer Science and EngineeringDaffodil International University

**DAFFODIL INTERNATIONAL UNIVERSITY**
**Dhaka, Bangladesh**

**September 17, 2025**

## APPROVAL

This Project titled **"Bangladeshi Vehicle Number Plate Detection: A Deep Learning Approach with YOLO Models"**, submitted by Name: Abdullah All Kafi . ID No: 211-15-4040 to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on **17 September, 2025**.

**BOARD OF EXAMINERS**

------------------------------------------

**Dr. Sheak Rashed Haider Noori** — **Chairman**
**Professor and Head**
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

------------------------------------------

Dr. Fizar Ahmed — **Internal Examiner**
**Associate Professor**
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

------------------------------------------

**Mushfiqur Rahman (MUR)** — **Internal Examiner**
**Assistant Professor**
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

------------------------------------------

**Dr. Md. Manowarul Islam (DMMI)** — **External Examiner**
Professor
Department of Computer Science and Engineering
Jagannath University

# DECLARATION

We hereby declare that this project has been done by us under the supervision of **Shah Md. Tanvir Siddiquee, 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:**

**Shah Md. Tanvir Siddiquee**
Assistant Professor
Department of Computer Science and Engineering
Daffodil International University

**Co-Supervised by:**

**Dr. Abdus Sattar**
Associate Professor
Department of Computer Science and Engineering
Daffodil International University

**Submitted by:**

**Abdullah All Kafi**
Student ID: 211-15-4040
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 Shah **Md. Tanvir Siddiquee**, **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 complete this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting themat all stages have made it possible to complete this project.

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

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

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

# ABSTRACT

The paper is a comparison of the performance of selected YOLO models (YOLOv8m, YOLOv12m, YOLOv11m, YOLOv9m, and YOLOv10m) that are used to detect vehicle number plates in Bangladesh with respect to precision, recall and mean Average Precision (mAP) at various Intersection over Union (IoU) thresholds. Following the hyper-parameter tuning, YOLOv8m yields the best performance with a higher accuracy (0.959) and recall (0.922), mAP50 of 0.946 and mAP50-95 of 0.55, which confirms its suitability to have in urban areas. Even though YOLOv8m is better compared to others, there are other models such as YOLOv12m, YOLOv11m, and YOLOv9m, which perform well and can be used as well in real-time number plate recognition systems. A comparative performance analysis presented in this paper covers a broader spectrum of vehicle detection tasks and reveals the evidence of strengths and weaknesses of each of the models. The results indicate that the YOLOv8m can adapt to the traffic infrastructure and urban surveillance systems in Bangladesh especially effectively, demonstrating the future opportunities of using deep learning-based object detection in addressing complicated issues in urban settings.

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

This chapter will be a summary of the research, which will give the background and importance of conducting the research. It describes the main reason why it is important to solve the vehicle number plate detection problem with the use of deep learning, then states clear goals. The methodology adopted is described briefly including data collection, model development, and deployment. The chapter also outlines the anticipated results of the project and gives the brief of the general organization of the report at the end.

## 1.1 Introduction

One of the most important features of smart transport systems (ITS) is vehicle number plate recognition (VNPR) which uses advanced machine learning and computer vision technology to facilitate the process of identifying vehicles. Automated number plate recognition must be one of the elements of the modern day-to-day traffic flow management of cities, and automatic number plate recognition is crucial to lead the way of road safety improvements, allow implementing traffic and crime laws, and monitor vehicles. As more and more cars fill the world, with Bangladesh not an exception, there is a need to find a system of effective and trustworthy detection to control traffic, curb crime, and secure the city [1], [2]. Improvements in the computer vision and deep learning processes of the recent past have changed the vehicle number plate detection technique quite dramatically. Convolutional neural networks (CNNs) have effectively replaced traditional techniques, feature-based, morphological and edge detection, as they are more precise and robust [3]. Of all deep learning models, the YOLO object detection networks are currently the most popular because they can real-time object detect, are fast, and highly accurate. Among YOLO models, the most notable among them, namely YOLOv8 and YOLOv12, have been reported to achieve high performance to detect a wide range of object types when it comes to applications involving vehicle number plates [4], [5]. YOLO-based models to solve number plate recognition of a vehicle are especially relevant in a city like Dhaka, Bangladesh, where vehicle traffic congestion and crimes against vehicles are a massive challenge [8]. Due to the continuous increase in the amount of vehicles on the roads, the detection and recognition of their license plates must be done accurately and automatically to improve law enforcement and traffic control.

Over the last few years there have been some sophisticated methods that have been considered in detecting the number plate on vehicles. Integration between the deep learning frameworks, including YOLOv3, YOLOv4, and more recently YOLOv8 and optical character recognition (OCR) has improved greatly. The techniques are usually two stage: one stage identifying the number plate of an image and the other stage character recognition with CNN-based techniques [7]. The accuracy of object detection in real-time provided by YOLO has made it an appropriate choice of the model in this application, particularly in dynamic settings such as road networks. Detection and recognition is further complicated by the uniqueness of the vehicle number plates in Bangladesh which not only have numerals but also contain characters of the Bengali language. Most of the literature is devoted to the plate forms popular in the Western world; however, the use of Bengali script on the number plates of cars makes it necessary to develop specially designed models that incorporate regional scripts [9].

The necessity to overcome these difficulties and design a well-developed, effective vehicle number plate detector system specific to Bangladesh is the background of this study. The purpose of this work is to utilize the capabilities of YOLO models, in particular, YOLOv8 to recognize vehicle number plates in real-time with a high level of accuracy and overcome the difficulties with environmental conditions and a variety of plate types [10].

The major aim is to assess the effectiveness of each model in recognizing vehicle number plates in different conditions including lighting variations, occlusions, and the different plate designs used in Bangladesh. It will compare the models based on the YOLO (You Only Look Once) family, namely YOLOv4, YOLOv5, YOLOv8, and YOLOv12, as well as one more model to cover the whole range of comparison [11],[12],[13]. All these models will be trained and tested on a personal dataset consisting of the pictures of vehicles belonging to various number plate formats, taken under various conditions to project real-world traffic situation in Bangladesh. The performance of each of the models will be tested on video streams to make sure that the models satisfy the requirements of real-time detection. This will be a simulation of a real traffic monitoring situation where the model will be required to identify number plates of vehicles that are moving on the camera frame. The real-time processing speed (FPS) will form a crucial parameter, and those models that cannot accommodate the minimum real-time processing speed will not be subject to further analysis [14], [15]. Once the models are compared with the said metrics, a general comparison will be conducted to determine which model is giving the best of the best to trade-off between the accuracy and the speed of car detection in the number plate in Bangladeshi environment. The findings will provide meaningful information regarding the best model to be used in the smart city and traffic management system in Bangladesh where real time number plate detection is the most important aspect in ensuring effective law enforcing and management of transport [16], [17].

## 1.2 Motivation

The rationale behind conducting this work is that three essential needs in the social and technological spheres are met: how to better urban traffic control, how to promote the applicability of deep learning models in real-time systems, and how to promote transparency in AI systems. With the population of vehicles in most of the urban centers like Dhaka continuing to rise, the demand on computerized systems that are able to provide efficient and accurate identification of car number plate to enable law enforcement, ease and simplify traffic management and offer hassle free services like collections of toll plaza and management of parking. The functioning of these systems has a direct impact on the economical activities, the population safety and the overall functioning of the urban infrastructure [18], [19].

Although true, these types of models should be optimized to execute in real-time, particularly in low resource environments with scarce computational resource and storage capacity [20]. The increased need to have real-time systems that perform best on embedded systems including edge devices, smartphones, or even low-cost surveillance cameras is the driver behind the need to design and optimize light-weight, memory-intensive architectures. The potential of using models such as YOLO in real-time in an urban traffic setting without compromising the accuracy of detection is therefore a target of this work [21]. Moreover, as the use of AI-based systems increases, the so-called black box of deep learning models is becoming an issue, which does not encourage users to adopt and trust them. In the case of systems used in publicly facing sectors, e.g., traffic law enforcement and security, both the correct and interpretable nature of the decision process of the AI system is of paramount importance. In this study, employed in an attempt to keep the process of vehicle number plate detection transparent and visible. By so doing, the study hopes to come up with a system that is not only performance-effective but also gives clean and understandable explanation of its forecasts thus promoting trustworthiness and reliability [22]. On a personal scale, this project will provide an excellent chance to solve an urgent urban issue and broaden an understanding of deep learning, and model optimization.

## 1.3 Objectives

This thesis will mainly aim at comparing the performance of various YOLO-based models used to detect vehicle number plates and design an efficient detection application in real-time. The idea is to make sure that the proposed solution is not only correct but also viable to apply in the real-life traffic environment, specifically under conditions when the resources are limited, such as in Bangladesh. The study seeks to meet the requirement of effective number plate detecting systems that are guaranteed to be real-time and available. The overall research process is informed by the following purposes:

- To produce a vehicle image plate dataset that is diverse: The dataset will be photographs in various environmental conditions, i.e. lighting, angle and occlusion. It will have number plates of many varieties with Bengali and Arabic numerals as well to capture the distinct feature of Bangladeshi vehicles registration. Data augmentation methods such as rotation, flip and brightness meticulousness will be used to improve the robustness and generalization aptitude of the model.
- In order to compare the performance of various YOLO models in the detection of car number plates: The benchmarking models will be YOLOv4, YOLOv5, YOLOv8, and YOLOv12. The models will be evaluated in terms of key classification metrics which include accuracy, recall, F1-score and mean Average Precision (mAP) at different Intersection over Union (IoU) levels. The performance of the models in real-time in terms of frames per second (FPS) will also be compared to ensure that it can be applied in the real world in real time traffic monitoring systems.
- In order to compare the efficiency of computation, detection performance as well as model complexity: The study will determine the most computationally efficient YOLO model according to the performance of detection and implementation in real-time. It will be compared based on the trade-offs between the complexity of the model (parameters) and runnability on embedded systems or low-power devices that will be deployed to implement traffic monitoring, e.g., edge devices and mobile platforms.
- To create a number plate recognition application in real time based on image of vehicle: The objective is to build an application that uses images of vehicles in real time to identify number plates using the Streamlit library. It will be easy to upload pictures of vehicles and real-time presentation of number plate detection. The application will be simple, quick to use, so that it will be simple to use in environments with limited

capabilities or network speed, like traffic stop or in the country.

- To apply the chosen YOLO model to the application to the real-time detection: The optimal accuracy YOLO model (both in terms of accuracy and performance) will be deployed to the web application. Real-time detection functionality will be modified to give immediate feedback to the users and, therefore, the system will be adapted to the real-life applications like toll collection, parking, and law enforcement.

## 1.4 Methodology

The method implemented in this thesis in the research is systematic and sequential including the data collection up to the implementation of a real time vehicle number plate recognition system. The processes are in such a way that each of them leads to the creation of an effective, accurate, and understandable model that can be implemented in real urban traffic conditions, especially in a resource-limited environment such as that of Bangladesh.

The initial phase entailed mass data gathering of different vehicles in Dhaka roads in Bangladesh. An iPhone camera was used to take high-resolution images of cars in different conditions such as lighting, the position and orientation of a car, and the environment. Images were used of cars with different number plates, but special attention was given to the Bengali format used as a number plate in Bangladesh, which tends to blend the Bengali script with Arabic numbers. This transformation in car number plates is illustrated below, whereby pictures of a typical car number plate (right picture) are shown. The data set is made of images which depict the dynamic and unpredictable urban traffic. Such images are confirmed manually and annotated to provide consistency and accuracy of the class labeling to 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 robustness and prevent overfitting. These techniques 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. In this study, five models based on YOLO (YOLOv4, YOLOv5, YOLOv8, YOLOv12) were chosen to compare them. The models have been implemented through a high-performance computing system and the performance was compared considering the relevant classification measures of precision, recall and F1-score. Real time processing FPS was also taken into account so that the possibility of the models being applied in the real-world scenario like traffic policing and monitoring could be taken into account. In order to overcome the black-box nature of deep learning models.


Figure 1.1: Expected Result of Detection with YOLO Model

It is also easy to use and the app has a real-time vehicle number plate detection interface with an advantage of interpretability of every decision of the detection, which is essential in boosting confidence by users.

## 1.5 Project Outcome

The results extend beyond the dataset creation, through the model performance, deployment preparedness to the explainability of the system and the research, therefore, is not only scholarly but also feasible to use in actual traffic system of an urban environment. The project solves such urgent problems of urban traffic control as the need to efficiently, scaleably and interpretably implement AI systems to detect vehicles using the synergy of deep learning and real-time performance. The study can be used to enhance effective models to be used in real-time traffic monitoring systems in order to assist in enforcing law, collecting tolls, and administering parking. In this work, a miscellaneous vehicle number plate information set, which encompasses very diverse circumstances such as different illumination, vehicle types, and number plate fonts (Bengali script and Arabic digits), was created. The exact data set is comprised of various annotated vehicle image depicting the distinctiveness of the Bangladeshi vehicle number plates, and a worthy reference resource in further vehicle detection studies.

The comparative systematic review of the five YOLO-based models, i.e., YOLOv4, YOLOv5, YOLOv8 and YOLOv12, was conducted to identify the most appropriate model to be used in order to detect vehicle number plates in real-time. The models have been thoroughly evaluated based on the fundamental measures and key metrics of precision, recall, F1-score as well as mean Average Precision (mAP). The findings provide a reference point by which model choice is measured on and provides a guideline on the future studies on vehicle detection systems and its application in a city setting. This comparison has indicated the trade-offs between the complexity of the models, detection quality and cost of computation, especially in real-time processing with constrained resources. A custom lightweight YOLO-based model was trained and tailored to meet the requirements of real-time, low-power applications in order to run it on an urban traffic environment. The optimized model to execute on the edge with Raspberry Pi and smartphones demonstrated comparable detection performance with very low computational requirements and much lower inference times than larger models. This is what makes the model especially usable in embedded systems to be utilized in traffic monitoring, police surveillance, and security cameras.

## 1.6 Organization of the Report

The report is put forward in six detailed chapters which are formed specifically to continuously address the research problem, summarize primary literature, describe the approach adopted, summarize findings of implementation and provide conclusions and recommendations. The format enables having a uniform presentation of information that would be indicative of the technical level of the study and the practical importance.

Chapter 1: Introduction
The first chapter is the setting of the research. It describes the reasoning behind the work, the technical and social background of locating vehicle number plates, and the necessity to use lightweight deep learning models in real-time applications in urban traffic systems. It also dictates objectives of research, research overview, anticipated results and the structure of the report in terms of chapter.

Chapter 2: Background
The chapter provides a comprehensive literature review of the current methods of vehicle number plate detection, and attention is devoted to YOLO-based models. The chapter discusses the benefits of deep learning models such as YOLO in object detection, summarizes past studies related to number plate detection, and discusses the role of Explainable AI (XAI) in the transparency and interpretability of such models. The chapter also shows the current problems and limitations of vehicle detection systems.

Chapter 3: Research Methodology

This chapter elaborates on the methodology adopted in the research. It begins with data collection for vehicle number plate images, followed by a description of preprocessing techniques such as resizing, normalization, and noise reduction. The chapter then discusses 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, optimized for low-power environments.

Chapter 4: Implementation and Results
The chapter demonstrates the specifics of implementation of the proposed system and a comparative study of all the YOLO models regarding their accuracy, precision, recall, F1-score, and computational efficiency. The other within the results is the inference time of the models with regards to the real time detecting of information. A lightweight YOLO model experimental object is developed and tested according to these standards.

Chapter 5: Engineering Standards and Design Challenges
This chapter gives a summarized version of the ethical and technical considerations that have been employed during the designing of the detection system. It describes the challenges which are faced, such as optimization of the model, dealing with the class imbalance and making the system interpretable with high performance. The chapter also recaps the adherence to engineering and AI ethics, such as the considerations that concern the privacy, transparency, environmental sustainability, and usability of users.

Chapter 6: Conclusion
The last chapter concludes the contributions in the research with the extension of an equilibrated vehicle number plate dataset, assessment of the YOLO model, and architecture of a lightweight but accurate detection system. It talks about the practical implication on the city traffic control, policing, and the development on the real time detection of vehicle number plates. Recommendations on future research would be to increase the dataset, enhance the generalization of the model, and generalize the system to new areas and applications, i.e. to other vehicles or climate conditions.

# Chapter 2

# Background

This chapter introduces a review of the main concepts and technologies in the study of the vehicle number plate detection by automatic systems based on YOLO models. The chapter starts with an overall introduction to machine learning and deep learning mechanisms, and their use in identifying objects in real time i. e. vehicle number plate detection. This chapter addresses the history of object detection models, in particular the YOLO family, because it is fast at real-time and accurate, gaining fame because of this quality.

## 2.1 Introduction

This chapter provides the necessary background information to the research report, which is the basis, which puts into perspective the methodologies and results that are to be further discussed in subsequent chapters. This study is mainly concerned with the creation of an automated vehicle number plate recognition system, which is a vital element in the traffic control of the city and security. Effective and precise vehicle identification is essential in many applications such as the collection of toll, law enforcement, and surveillance in the city. Timely and accurate identification of number plate information is important in the fight against crimes related to traffic, to support security agencies, and to improve on the general traffic systems. The introduction shows the importance of vehicle number plate detection technologies and how they relate to such innovations. Machine learning and deep learning are capable of object detection in recent years, providing an autonomous solution in the situation of vehicle number plate recognition at a very high level of accuracy. Of these, the YOLO models have attracted a lot of attention due to their capability of accomplishing real-time object detection that is highly precise and efficient.

Despite the considerable progress in vehicle number plate detection, there have been numerous issues which have remained, particularly in detection of vehicle models in real traffic environments, scalability and responsiveness to different environmental conditions such as light and obstructions. This is important in practical applications such as traffic control and policing where it is important that model decisions can be explained so as to be accepted and deployed effectively. The uniqueness of the research presented is in the fact that different YOLO models were compared in terms of vehicle number plate detection and that interpretability tools were implemented to enhance the user experience and decision-making process. The chapter will also discuss the difficulties and innovations that have been dealt with in this research providing the foundation of the methodologies and experimentation that will be discussed in the following chapters.

## 2.2 Literature Review

One of the most important elements of intelligent traffic monitoring systems is Vehicle Number Plate Detection (VNPD). As traffic jam is increasingly being experienced within cities like Dhaka, Bangladesh, the need to enhance traffic control, police patrol, and even the security within the city comes in to play as vehicles are identified with. The former number plate recognition techniques are not only fast but they can also not be effectively used in real time besides being very slow and prone to errors. To address this, the use of deep learning, in particular, Convolutional Neural Networks (CNNs), has been widely explored to improve and automatize VNPD. Among such techniques, one of the most effective and real-time processing algorithms is the YOLO (You Only Look Once) algorithm. This review is a critical analysis of several prominent VNPD studies devoted to the YOLO-based detection methods.

Laroca et al. [5] have suggested a very effective Automatic License Plate Recognition (ALPR) system based on the YOLO detector. They emphasized layout-independent number plate detection under a variety of environments to guarantee that the model can be readily integrated to other types of vehicles and regional models. Their emphasis was on the performance of YOLO at a higher frame rate, which detects vehicle number plates, and higher accuracy. Nonetheless, their approach was only centered on recognition stage, such as character segmentation which falls outside this study.A YOLO-based approach to identify license plates of helmetless motorcyclist was suggested by Jamtsho et al. [6] as a way of detecting and monitoring them in real-time. The authors used one convolutional neural network (CNN) to identify license plates in video frame. In order to reduce the number of false positives caused by objects such as helmets, centroid tracking technique was employed. This method had a detection rate of 98.52. The contribution of this paper is the successful use of YOLO in real-world and practical license plate detection under diverse conditions with the minimum human intervention.

Chung et al. [7] proposed an improved version of YOLOv7, YOLO-SLD, which has the addition of the SimAM (Simple Attention Mechanism) to enhance the detection of license plates. Their strategy deals with the problem of different lighting conditions, plate positions among others through the attention mechanism that assists the model to concentrate on the most important features. The experiment results revealed that accuracy increased by 0.44 percent with the default YOLOv7 to 0.98 with the YOLO-SLD. Furthermore, the parameter size of the model was decreased by 1.2 million, which allowed to make it lightweight without deterioration of its performance. In the meantime, Min et al. [8] introduced a novel vehicle license plate detection algorithm based on the Yol-L model that pre-processes the plate to optimize the algorithm in the adverse environment such as weather changes and different viewpoints. They modified the YOLO-L to add k-means++ to cluster the model and choose the candidate boxes covering the license place areas in order to improve the accuracy. The technique had 98.86 recall and 98.86 precision rate, which is better than the current methods. The advantage of this work is the addition of YOLO-L to plate pre-identification, which contributes to the removal of false positives of the objects represented by road signs and billboards.

Jamtsho et al. [9] in a similar study concentrated on the localization of Bhutanese license plates with the aid of YOLO, taking into consideration the peculiarities of Bhutanese license plates. The authors performed vehicle and license plate detection with YOLOv2, which yields successful localization of license plates on video streams with a mean average precision (mAP) of 98.6%. The system used YOLO to recognize vehicles as well as license plates, and reduced false positives due to similar objects in the background, namely signboards. Ravi Kumar et al. [10] also suggested a vehicle number plate recognition system based on machine learning and focused on identifying plates with the help of morphological operations. In their approach, they used diverse image processing methods such as grayscale conversion, bilateral filtering, and edge detection to locate the area of the number plate. Character segmentation and identification was then done by Optical Character Recognition (OCR). Even though their method was found to have high detection accuracy, it was based more on conventional image processing algorithms, and that future solutions built on deep learning should be considered to detect number plates.

In the recent past, Rathi et al. [11] deployed the YOLOv4 deep learning algorithm to detect vehicle number plates in parking lots in real time. YOLOv4 was utilized to identify vehicle number plates and Optical Character Recognition (OCR) was utilized to read the characters on the identified plates. The system was at a high level of detection rate of about 89% thus it is suitably used in urban surveillance and intelligent transport systems. The application of the YOLOv4 to real-time detection in a controlled setting, e.g., parking places, under consideration of the difficulties in the uncontrolled setting (poor lighting and different orientations of cars) is one of the significant contributions made to this work. A YOLOv3 Tiny-based vehicle license plate detection system was proposed by Beratoglu and Toreyin [12], with the model deployed to compressed video data to save on computation time without reducing detection rate. Their model showed a 30 times shorter time to infer than standard YOLO models, significantly increasing the practicability of vehicle number plate detection under resource-constrained conditions, which are appropriate in real-time traffic surveillance.

Ga et al. [13] increased the accuracy of YOLO through the introduction of Super-Resolution Generative Adversarial Networks (SRGANs). This combination worked especially well to improve low-resolution images that are utilized in surveillance systems. An accuracy of detection in their proposed framework was 98.5, much higher than that of traditional detection systems, which tend to fail with low-resolution inputs. The major finding of this paper is the advancement of image enrichment algorithm and YOLO as a remedy to the issues with low-quality images in practice. In the meantime, Min et al. [14] proposed a more advanced feature of the YOLO architecture, namely, the YOLO-L architecture, which featured some changes aimed at enhancing the capabilities of the model to classify and locate vehicle number plates. Their solution resolved such problems as occlusion, changing lighting conditions and the angles of the vehicles. The accuracy of the YOLO-L was 98.8% and thus, it is a very effective solution in detection of number plates in dynamic environments. The primary value of the given work was the creation of the specific version of YOLO, adjusted to the real-life traffic conditions.

The vehicle number plate detection system that Kumar et al. [15] developed was developed particularly, and this is applicable to the roads of Bangladesh. Their system used YOLOv4, which provided great performance in the detection of vehicle number plates in difficult environmental conditions (heavy traffic and low visibility). The experiment obtained a mean average precision (mAP) of 97.3% and has shown the real-time applicability of YOLOv4 in detection, which makes it valuable in development of intelligent transportation system on urban roads in Bangladesh.

### 2.2.1 Similar Applications

With the help of deep learning and mobile technology, vehicle number plate recognition performance has been greatly improved. New research is being carried out through the integration of novel machine learning methods and quick processing, to make such detection systems more efficient, reliable and applicable, especially to urban environments. Vehicle number plate detection systems can also be considered an application of image-based object detection, using deep models to locate the objects in the main roads, parking areas and toll booths.

Recently, Shashidhar et al. [17] proposed a new automatic number plate detection and recognition system for vehicles that employs the Region of Interest (ROI) detection through YOLO-V3 and optical character recognition (OCR) through Convolutional Neural Network (CNN). The system was demonstrated to work with an accuracy of 91.5% for a custom dataset of 6,439 Indian number plate images. Moreover, the major novelty of this paper is the application of the Wiener filter to blur images of moving vehicles to fundamentally improve the quality of the blurry images. Number plate detection was performed using YOLO-V3 and character recognition was carried out using CNNs. The two-step process allowed robust detection even under blurry conditions and a much-improved overall system performance. The proposed system achieves better results than conventional license plate recognition systems by applying state-of-the-art pre-processing techniques and tuning the model for real-time operation. Atikuzzaman et al. [18] suggested a real-time number plate detection and classification system based on CNNs. The system was built in three main stages: plate detection, class letter segmentation and character recognition. A HAAR Based Feature Classifier was used to perform plate detection, character recognition was done using segmentation partitioned class letters, and CNNs. The system achieved a recognition rate of 91.38 and frame rate of 30 frames per second. It also had a plate image detection accuracy of 96.92, a segmentation accuracy of class letter of 94.61 and a real time detection recognition of 90.90. The system was tested using 390 test images and the concept was found to be successful in vehicle classification. Automatic number plate recognition ANPR with this method would be applicable in real life situations such as parking control, tolling and security.

Al-battan et al. [19] suggested that an end-to-end Automatic License Plate Recognition (ALPR) system should use YOLO v2 to detect vehicles (VD) and YOLO v4 to detect and recognize licenses plates with a recognition rate of 90.3%. The system is vehicle and license plate detection, vehicle categorization of emergency vehicles and heavy vehicles with real-life traffic monitoring. This model does not require any additional pre-processing and post-processing procedures that enables simple extension to additional generalization on additional datasets. It performed highly on real-time applications with acceptable FPS even on a low-end GPU. We evaluated our hybrid scheme on five publicly available data sets and it was found to be better than those past systems both in terms of accuracy and speed. Laroca et al. [20] in another work suggested a valid and real-time ALPR system, which is founded on vehicle and license plate detection on the basis of the use of the YOLO v2 that provides the recognition accuracy of 93.53% with a 47 FPS. They use YOLO v2 to detect and CR-NET to recognize characters and their performance is better than that of commercial systems like Sighthound (98.10) and OpenALPR (93.03). Also the UFPR-ALPR dataset is provided which has taken into consideration various real world conditions like the type of vehicles, light, and moving camera angles. Their solution delivered outstanding improvements on vehicle and license plates detection that will be an efficient answer to applying ALPR in real-time.

In the meantime, Hossain et al. [21] suggested an advanced LPR system with the usage of YOLO v8 to detect cars and Optical Character Recognition (OCR) to segment the characters. The system was very accurate with a license plate detection rate of 99% and character recognition rate of 98%. The use of YOLO v8 license plate detection and OCR character extraction together helped this system to defeat challenges like vehicle occlusions and low light. The preprocessing techniques employed by the authors also included K-means clustering, thresholding, morphological operations to enhance the quality of input data so that detecting better results can be achieved. This system was tested and verified on a custom dataset of 270 different images and is therefore a good solution in the real time LPR applications. Swathi et al. [22] introduced a real-time number plate detection system that uses AI and machine learning, namely using Region-based Convolutional Neural Networks (RCNN) and Advanced RCNN as the proficient algorithms to identify vehicles and license plates. Some of the issues that the system would solve in real world conditions are different lighting conditions and environmental issues. The authors implemented the system as a cloud-based system, which increases scalability and access. Their approach was very precise and able to recall, which provides a viable solution to intelligent transportation systems, and their findings revealed that the accuracy was high in a dynamic world. The paper also states the significance of the optimization of the system parameters and the benefits of using clouds to boost the throughput and scalability.

Swapna et al. [23] designed an automated solution to detect motorcyclists without helmets based on vehicle license plate recognition and helmet detection with YOLO v9. The algorithm uses background subtraction, Canny edge detection and morphological operations to divide the vehicles in video content. The non-helmeted riders are detected with YOLO v9 and the license plate numbers are extracted with the help of Optical Character Recognition (OCR). The system had a better performance than the former models and it had 93% precision, 94% recall, and 95.3% accuracy; it is a good solution to real-time implementation of road safety laws in motorcycles.

### 2.2.2 Related Research

Within the last several years the application of deep learning methods, particularly Convolutional Neural Networks (CNNs) and YOLO model has significantly increased the automatic number plate recognition (ANPR) system rendering it more applicable in traffic monitoring systems in real time. In combination with character recognition CNN (Optical Character Recognition - OCR), YOLOv3 detection has been demonstrated to make 91.5% accuracy in number plate detection even with blurry images, when used as a Region of Interest (ROI) detector of vehicles in images. Similarly, Swapna et al. [26] used YOLO v9, which detects both non-helmeted riders as well as car license plates, obtaining 93% accuracy, 94% recall, and 95.3% precision. Mustafa and Karabatak [40] proposed YOLOx for car detection and YOLOv4-tiny for automatic number plate recognition under environmental conditions with different illumination and achieved an accuracy of 97.5%. These studies showcase the efficiency, cost-effectiveness, and accuracy of YOLO models, making them the go-to choice for modern ANPR systems.

The paper of Ottaviani et al. [27] shows a real-time number plate recognition system for traffic monitoring of motorways. Full video sequences are stored without external triggering of image acquisition, and the system is significantly more continuous in operation. Using a fast algorithm, the system processes images in less than 40 milliseconds per frame, which means that it can process video in real time even in high-speed traffic. The system is composed of local processing units by radio coupled to a control station to provide complete coverage of traffic flow. It also consists of temporal post-processing to enhance the recognition accuracy under clear illumination conditions and on-line warning to vehicles on a "blacklist." The system is remarkable in the efficiency of vehicle detection process, taking advantage of video frames to increase accuracy even in low-light conditions, thus proving the practicality of automatic vehicle identification systems. Referring to this topic, Roy et al. [28] presented an improved segmentation technique for Automatic Number Plate Recognition (ANPR) to overcome the difficulties that arise from non-uniform shapes of the plates and font differences across different countries. Through the pixel level segmentation and character recognition using an Artificial Neural Network (ANN), the system achieved 91.59 percent success on 150 number plates of different countries. The research suggests a new region-expanding segmentation algorithm and gives one the chance to re-train the system neural network using the samples of his or her own fonts, which makes the system flexible to diverse international norms. The solution is robust for international applications especially for cross-border traffic management. Sarbjit Kaur [29] proposed the innovative ANPR method with the aim of improving it in low-contrast, blurring, noise, and multi-lighting. The number plate is extracted with the help of the morphological operation and boundary box analysis, and the other pre-processing is the iteration of bilateral filtering, adaptive histogram equalisation technique to enhance the image quality. The method is able to contend plate extraction and segmentation difficulties despite a lack of ideal circumstances with template matching character recognition. The results of this approach were better than the traditional ANPR with regard to the image quality parameters like MSE and PSNR, the success rate in every stage of number plate recognition was higher. This approach is appropriate to the real-time applications which require a high precision even when the environmental load is high.

Sarfraz et al. [42] suggested a real time Automatic License Plate Recognition (ALPR) system for use with CCTV images specifically for non-dedicated cameras in low resolution environments. The proposed method performs well in vehicle localization, tracking and recognition under adverse conditions such as low illumination and non-standard camera views, and low-resolution video. Automatic adaptation to changes in the scene due to background learning, dynamic license plate detection and template matching. It enhances the character recognition by majority voting on multiple frames to reduce the false positives. With an average detection rate of up to 94%, the system is ideal for traffic surveillance and forensic applications where conventional ALPR systems are less than satisfactory. This solution is designed for use in a controlled environment and can be scaled and made resilient for use in un-controlled environments such as public CCTV networks. In another work by M.S. Sarfraz et al. [33] a stable real-time ALPR framework was introduced on CCTV video data (video data usually yielding lower resolution and quality than traditional ALPR systems). The proposed framework overcomes issues such as noisy, low-resolution videos and dynamic light conditions, typical in the forensic video environment. The system supports dynamic background learning for licenseplate detection, tracking and recognition, making it robust for camera view and distance variations, and enabling progressively higher recognition rates. In addition, the use of majority voting decreases the recognition error. This system shows great performance in terms of processing speed and the precision, particularly when processing CCTV video with low-quality streams. The paper presents an important solution for use of real-time license plate recognition in challenging, unconstrained environments, especially for security and forensic applications. Arth et al. [34] first proposed a real-time license plate recognition (LPR) system implemented on an embedded DSP platform and designed to run automatically using the standard video stream without any other sensors. The system is divided into two main modules, namely license plate detection and character recognition. The detection module applies a Kalman tracker-based Viola-Jones-based AdaBoost method to predict license plate motion in future video frames, which can greatly improve the processing speed. Segmentation of the characters is achieved with a region-based approach, which is then followed by support vector classification for recognition. The system overcomes the low resolution via frame integration and high accuracy against poor video quality is thus attained. The embedded platform's real-time nature makes it suitable for traffic management applications, delivering an efficient and cost-effective license plate recognition solution without requiring specialized hardware.

## 2.3 Gap Analysis

In spite of the significant advances in Automatic License Plate Recognition (ALPR) field, several obstacles still prevent the implementation of robust and scalable systems for real-world applications especially in the urban environment. Most currently existing systems, including Otto-Vanni and colleagues [37], are tailored for a specific application, such as highways, and thus urban traffic monitoring has largely been ignored. Also, conventional methods cannot handle different license plate designs and low-quality images due to poor lighting or occlusion as pointed out by Roy et al. [38] and Sarbjit Kaur [39]. While recent studies, for example Sarfraz et al. [42], deal with CCTV applications, these systems are still not enough for low-resolution or non-dedicated cameras. Many models rely on specialized hardware and hence are not suitable for large scale traffic management as indicated by Arth et al. [34]. Further, like Patel et al. [35], the inflexible way of handling the continuously changing nature of the formats of global license plates, environmental conditions limit the scalability of such systems. My contribution is to overcome these challenges by leveraging YOLO-based models, which are flexible, efficient, and robust with respect to different environments, conditions, and license plate formats, which make them a versatile solution for urban traffic monitoring and real-time vehicle number plate recognition.

## 2.4 Summary

This section describes the state of the art of vehicle number plate detection based on deep learning approaches such as Convolutional Neural Network (CNN) and Transfer Learning (TL) using YOLO models. It has been demonstrated that YOLO-based models are highly promising in vehicle number plate detection with high accuracy and efficiency especially in the context of real-time. Nevertheless, some issues remain to be resolved such as when faced with differences in luminance and background, and non-standard plate sizes. Transfer Learning has offered a practical method of addressing the small quantity of labelled data through fine tuning an existing model such as YOLO, but is still hindered by domain shift and is marked by stringent adaptation to plate formats. However, these methods have not been extensively adopted particularly in applications in real time. These limitations point towards the necessity to have more interpretable, efficient and scalable detection systems. To address these problems in this paper, a lightweight vehicle number plate detection system based on YOLO is designed.

# Chapter 3

# Research Methodology

This part outlines the state of art of vehicle number plate detection with the use of deep learning models including Convolutional Neural Network (CNN) and Transfer Learning (TL) using YOLO models. YOLO based models have shown to be highly promising in vehicle number plate detection, both in terms of accuracy and efficiency, especially when running in real time. But there are issues to be addressed yet, such as when faced with differences in illumination and background, and non-standard plate formats. Transfer Learning has offered a practical solution to the scarcity of labelled data through fine tuning an existing model such as YOLO but it is also fraught with domain shift and needs to be carefully adapted to different plate formats. Yet, these techniques have not been widely implemented, especially in real time applications. These constraints highlight the need for more interpretable, efficient and scalable detection systems. In this paper, to overcome these issues, a lightweight YOLO-based system for real-time vehicle number plate detection is developed.

## 3.1 Methodology

### 3.1.1 Overview

This study is organized in a systematic manner to come up with a real-time, precise and light vehicle number plate detection system with the help of YOLO models. The algorithm begins by gathering a number of traffic areas in Bangladesh so that there is a vast variety of vehicles and number plate. This crude data is then pre-processed by resizing, normalising and formatting the images so that they can be uniform. To enhance the dataset, there is data augmentation (rotation, flipping, noise injection, contrast manipulation et cetera) to enhance the ability of the model to generalize.

Once the dataset has been prepared, the vehicle number plate detection system is trained with the help of different YOLO models (v8, v9, v10, v11 and v12). Training and tests of each version of YOLO will be done to identify the most effective version in terms of accuracy, speed, and computation. A comparative analysis is carried out to determine the most suitable YOLO architecture suitable to adopt in real time detection under different traffic conditions. The resulting model is then tried with regard to its performance measures (i.e. precision, recall and F1-score) to verify whether it is useful in real world scenarios. The methodology concludes with the selection of the most appropriate model based on the performance and real-time detection performance in order to be applied in the traffic monitoring and other applications.

### 3.1.2 Proposed Methodology/ System Design

This paper designs a powerful vehicle number plate recognition model based on the application of deep learning models called YOLO, starting with the data gathering in different types of traffic areas like highways and institutions owned privately so that the applicability of the model is not restricted. The gathered data is pre-treated by resizing, augmentation (flipping and rotation), and normalization to improve the performance of the model. The dataset is divided into training, validation, and testing sets, with YOLO models (v8, v9, v10, v11, v12) being trained and compared based on metrics like accuracy, precision, recall, F1-score, and confusion matrix. Techniques like hyperparameter tuning, early stopping, and learning rate scheduling are applied to prevent overfitting. Additionally, a user-friendly UI is developed for real-time detection, allowing users to upload images and receive confidence scores for detected number plates. The system's performance is evaluated through error analysis and model comparison to ensure its suitability for real-world traffic surveillance deployment.

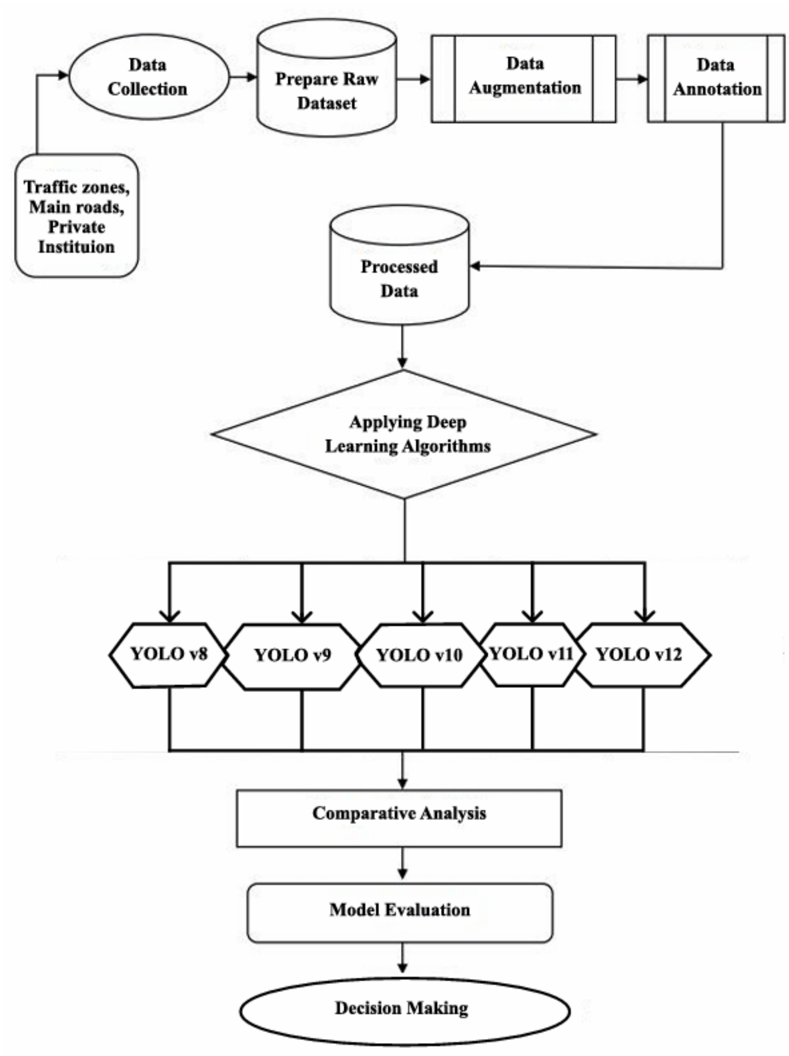



Figure 3.1: Workflow Design for the Proposed Detection System

### 3.1.3 Functional and Nonfunctional Requirements

The system proposed is furnished with both functional requirement and nonfunctional requirement in order to have strong performance and practical usability. The functional requirements deal with the main tasks of the system and the nonfunctional requirements deal with the performance, usability as well as security requirements of the system in the real life applications.

**Functional Requirements**

- Image Upload: The user has options of taking pictures of vehicles using the device camera or gallery in the device.
- Image Preprocessing: The system will automatically downsize, normalize and optimize the uploaded images ready to detect number plates.
- Vehicle Number Plate Detection: The system will identify the vehicle number plate in the uploaded image with deep learning software ( YOLO models ) and

identify the plate area.

- Model Selection and Inference: The system will first be trained on several YOLO models (v8, v9, v10, v11, v12) in order to identify the most optimal model in terms of accuracy, efficiency, and inference speed. Final inference will be performed by YOLOv12.
- Output of Prediction: The system will show the area of the number plate identified and the number predicted as the plate number and a score of confidence (e.g., Confidence: 95.2%).
- Feedback Module: The system will alert users on successful uploads or processing status or errors and send feedback to them in visual or text format.
- Offline Prediction Capability: The system will provide offline feature, whereby its users can upload and process images even in the absence of an internet connection.
- User Interface: The system will have a user friendly web interface that is easy to interact with because it will enable the user to upload images and see the results without difficulties.
- Logging and Storage: The system will have appropriate error management and troubleshooting of unsupported file formats, low quality images or processing errors.

**Nonfunctional Requirements**

- Performance: The system should provide an inference result within 3-5 seconds on the mobile and web platforms.
- Precision: YOLO-based lightweight model must have minimum 90 percent accuracy on vehicle number plates detection with real-world and augmented set.
- Usability: The user interface must be well-designed, reactive, and user-friendly and must be catered to users who might have little to no technical expertise like drivers, traffic officers and security personnel.
- Scalability: The system must be able to accommodate subsequent modifications, such as the inclusion of new vehicle categories, plate design or other traffic related data.

### 3.1.4 Context Diagram

The Context Diagram gives a general view of the vehicle number plate detection system as a coherent system and how it interacts with the outside world. It establishes the boundaries of the system and illustrates how information moves in and out of the system or into the users or external sources of the system. The main external party in the project is the User that communicates with the system uploading vehicle images and receives bounding box predictions and visual explanations. The system comprises all internal operations, which include preprocessing and classification with the help of YOLO models.

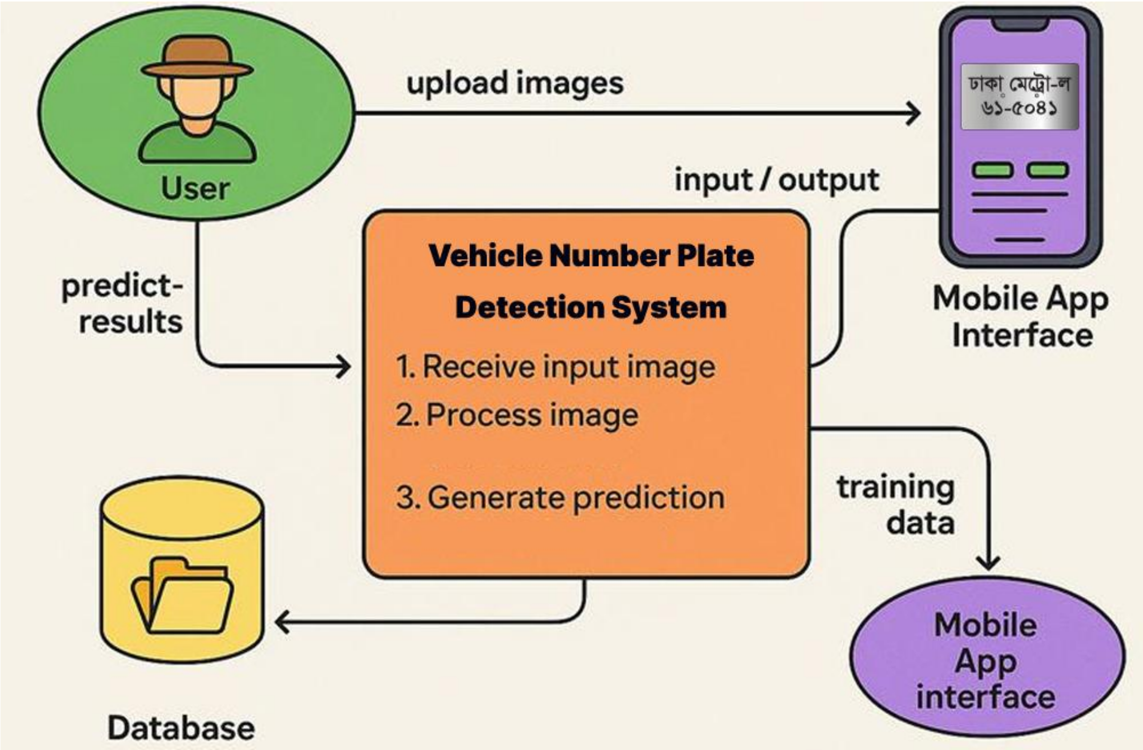



Figure 3.2: Context Diagram

### 3.1.5 Data Flow Diagram Level 1

Level 1 Data Flow Diagram (DFD) shows the internal flow of data in Vehicle Number Plate Detection System and how information flows between different modules of the system. It begins by the user uploading a photo of a vehicles number plate, which is sent to the Image Input Module. This image is then sent to the Preprocessing Module where it is resized, normalized and augmented. The training and testing of the enhanced images are retained in the Augmented Dataset. The image is then forwarded to Model Inference Module, which links the image to the YOLOv8 model to recognize the vehicle number plate. Transfer learning is used to optimize performance of the model as per the dataset. Once inference is done, the number plate that has been predicted with its CL is sent to Explanation Module. Lastly, the result with the estimated number plate, confidence score and explanation is shown to the user in the Result Display Module. This Level 1 DFD provides a modular breakdown of the system, which depicts the succinct flow of the process of raw image input and understandable and explicable predictions to the user.

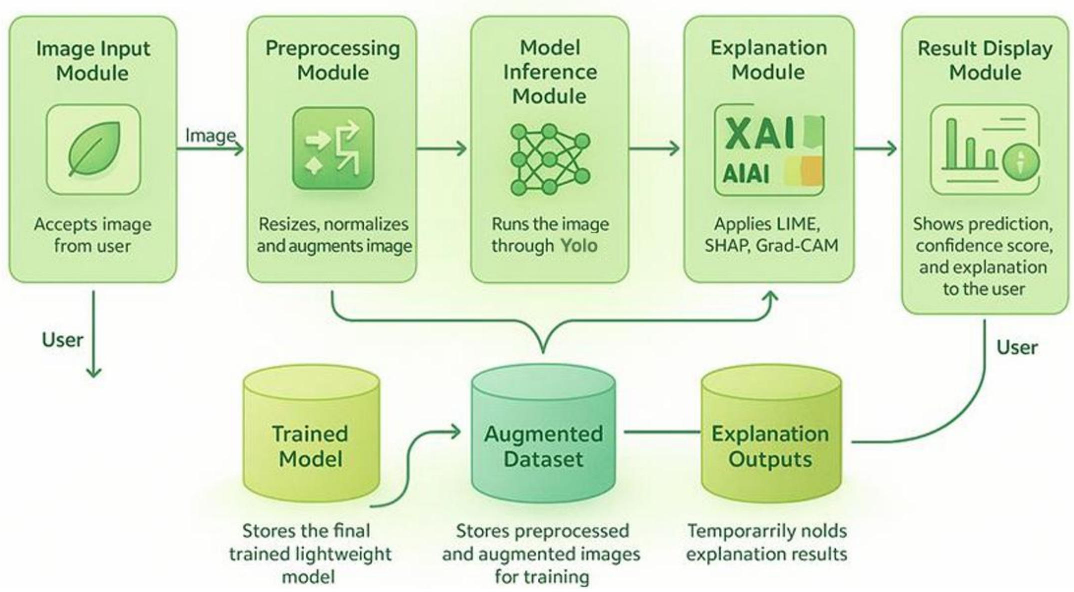


Figure 3.3: System Data Flow Diagram

### 3.1.6 UI Design

The UI screenshot exemplifies the system, Vehicle Number Plate Detection Using YOLOv8, which is simple to use. It has two buttons, one that is called Choose File to upload vehicle pictures, and there is one called Detect that will initiate the detection process, using YOLOv8 to detect the number plate. After processing the system shows the detected number plate with a bounding box and a confidence score (e.g., 0.76) to show to what extent the model is sure. Under the image, the system gives a textual message of the identified object, like license-plate, so that users can know what the result was. This lightweight user-friendly interface is tailored towards individuals who have little or no technical skills, and therefore it is appropriate to operate in real-time vehicle number plate detection in different environments.

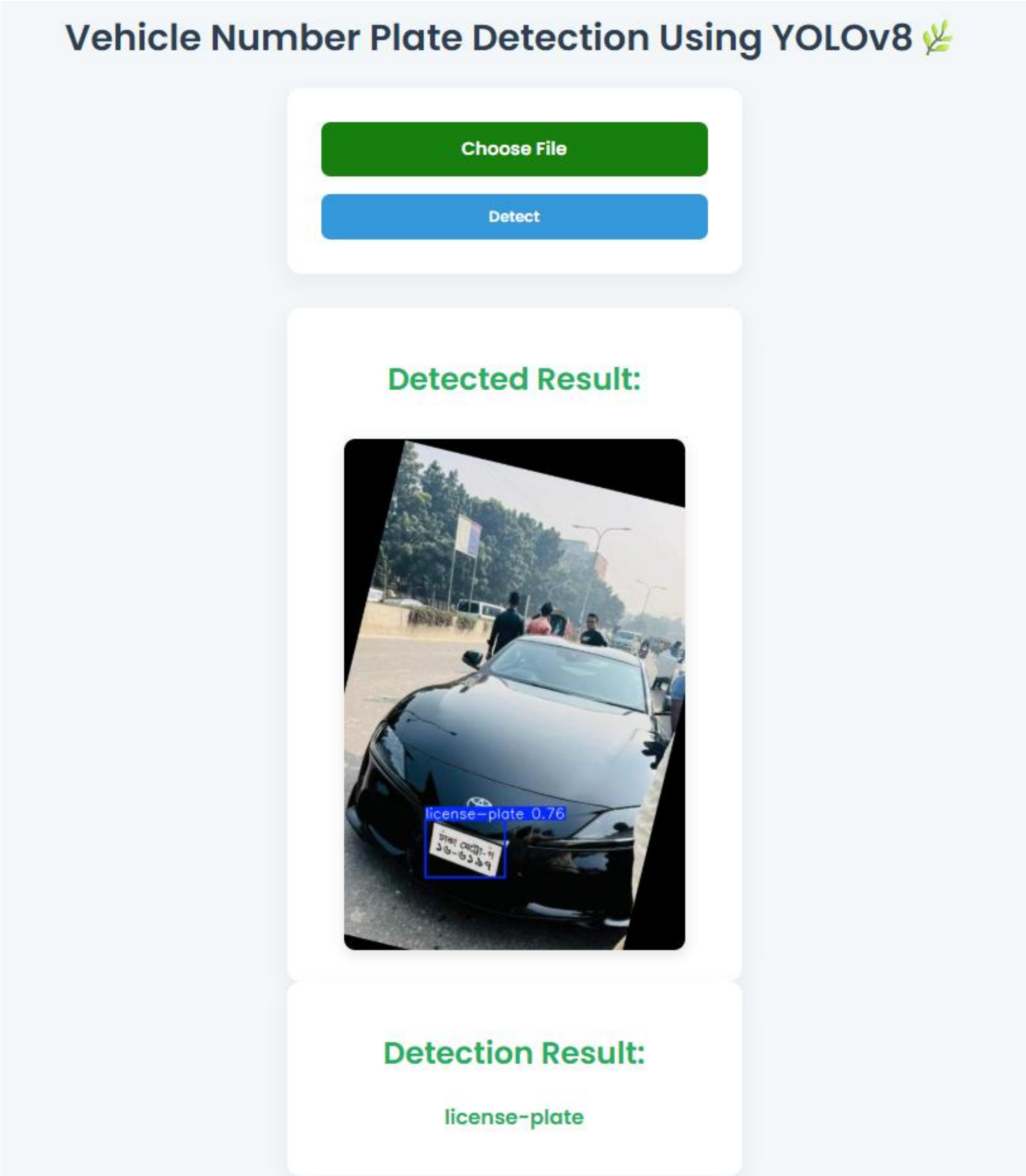


Figure 3.4: UI of the Detection System

## 3.2 Detailed Methodology and Design

Vehicle Number Plate Detection Using YOLOv8 system is created to be functional with regard to real-time identification. It starts by creating a diverse set of vehicle images that contain number plates, and which are captured in varying situations like the weather and the traffic. This data has undergone preprocessing and normalization, rescaling and augmentation (flipping, rotation and brightness adjustments) to increase the variability and the robustness. This long data is split into training, validation and test set so as to train the model, scan hyperparameters and test the model. Phase II: The latest object detector that is found to be state-of-the-art is the YOLOv8. It utilizes CNN structure to derive features, estimate bounding box positions and categorize the objects, in this case, vehicle number plates. The ability to use anchor boxes, as well as multi-scale features, makes YOLOv8 very accurate, even in small number plates. The model would give the bounding box and a confidence score that shows whether the prediction is reliable or not.

### 3.2.1 Dataset

#### 3.2.1.1 Data Collection

A vehicle number plate identification dataset was gathered in different traffic areas such as roads, institutions and busy locations by the use of both smartphones (Realme C25s) and a digital camera (Canon EOS 70D). The data were collected in a period of one month (November to December of 2024), where they recorded the images with various environmental settings, times of the day (morning, noon, and late afternoon) and viewpoints (top-view and side-view). The dataset was initially composed of 3,507 images, which were cleaned and the duplicates eliminated. To increase diversity and generalization of models, data augmentation methods (rotation, flipping, scaling, noise injection) were used to increase the dataset. The pictures were converted into JPG format so that they could be processed easily.

#### 3.2.2.2 Dataset Description

The vehicle number plate detection dataset was first of 3,507 images in different traffic localities, roads, and privately-owned institutions. On cleaning up and elimination of duplicates, 3,045 images were left. Data augmentation strategies included rotation, flipping, scaling, noise injection and contrast manipulation that boosted the size of data to 7,495 images. The data set includes one, "Number Plate" which is to be detected. It is divided into training (86 percent, 6,442 images), validation (7 percent, 524 images) and testing (7 percent, 524 images) sets. This well-balanced distribution helps the model to be able to generalize and be effective during both training and evaluation phases. The pie chart below, which is labeled as the DataSet Distribution, indicates the distribution of training, validation and testing.

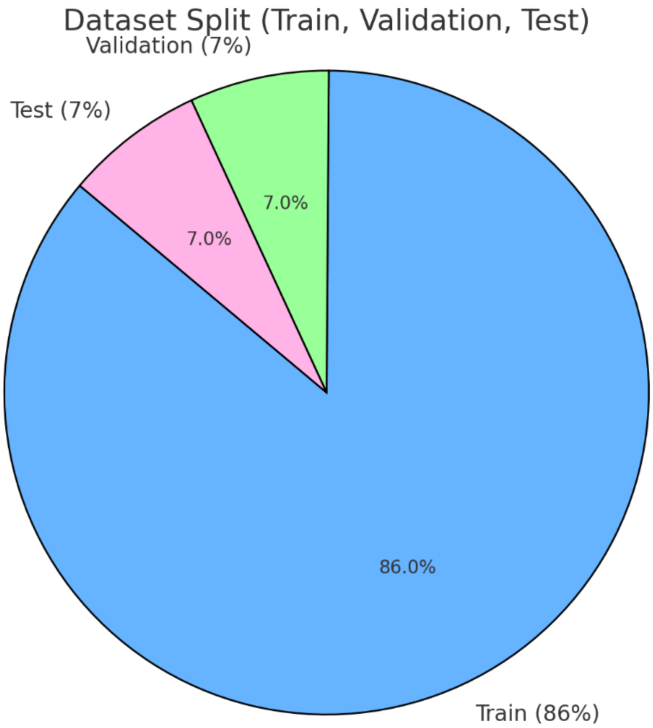



Figure 3.5: Pie-Chart of Dataset Distribution

Figure 3.6 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: Image Samples from each classes

### 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. This resolution standardization helped ensure consistent input data for deep learning models, enabling effective feature extraction and enhancing training efficiency.

- Dataset Splitting: The data was divided into three parts 86 percent training, 7 percent testing and 7 percent validation. This separation has been done based on stratified sampling so that the original distribution of the data set is preserved in every sub-group.
- Data Cleaning: The data cleaning was done manually in order to exclude the duplicate or irrelevant images. This facilitated the fact that the dataset was of only high quality and well labeled images. A thorough review of all the images was conducted, and superfluous samples were eliminated in an attempt to ensure a high degree of quality of the data to be fed into the model.
- Annotation: The free version of Roboflow was used to annotate the dataset.

### 3.2.3 Model Selection

A total of five variations of the YOLO (You Only Look Once) model were selected in this work that are used to estimate the performance of the vehicle number plate detection: YOLOv8m, YOLOv9m, YOLOv10m, YOLOv11m and YOLOv12m. The choice of these models was to compare the performance of the YOLO models with the different versions that are an incremental improvement in model architecture, training optimization and the ability to extract features. YOLOv8m was the most popular and was used as a baseline. The use of YOLOv9m, YOLOv10m, YOLOv11m and YOLOv12m was selected with the purpose of investigating the innovations in the detection accuracy, model robustness, and inference speed with each of the versions offering new approaches such as refined backbone networks, enhanced anchor generation, and real-time applications. The reasons that were imperative in the selection of the models were the following:

#### a) YOLOv8

YOLOv8 is the most recent release in the YOLO (You Only Look Once) lineup, an object detection algorithm, which is a real-time solver and able to perform high speed and precision predictions. YOLOv8 is the extension of the previous ones (YOLOv7, YOLOv6) and includes a number of improvements in terms of speed, accuracy, and architecture. It implements the backbone network architecture provided and maximizes the detection head that leads to the improved feature extraction and detection output accuracy, especially in real-time applications like vehicle number plate recognition. The model can detect objects in real-time, and can identify objects under various conditions including low-resolution images and high-moving objects, and hence is very handy in vehicle number plate recognition. YOLOv8 is built on an efficient architecture, which can process large images as well as a high frame rate and can be deployed to embedded systems and mobile applications.

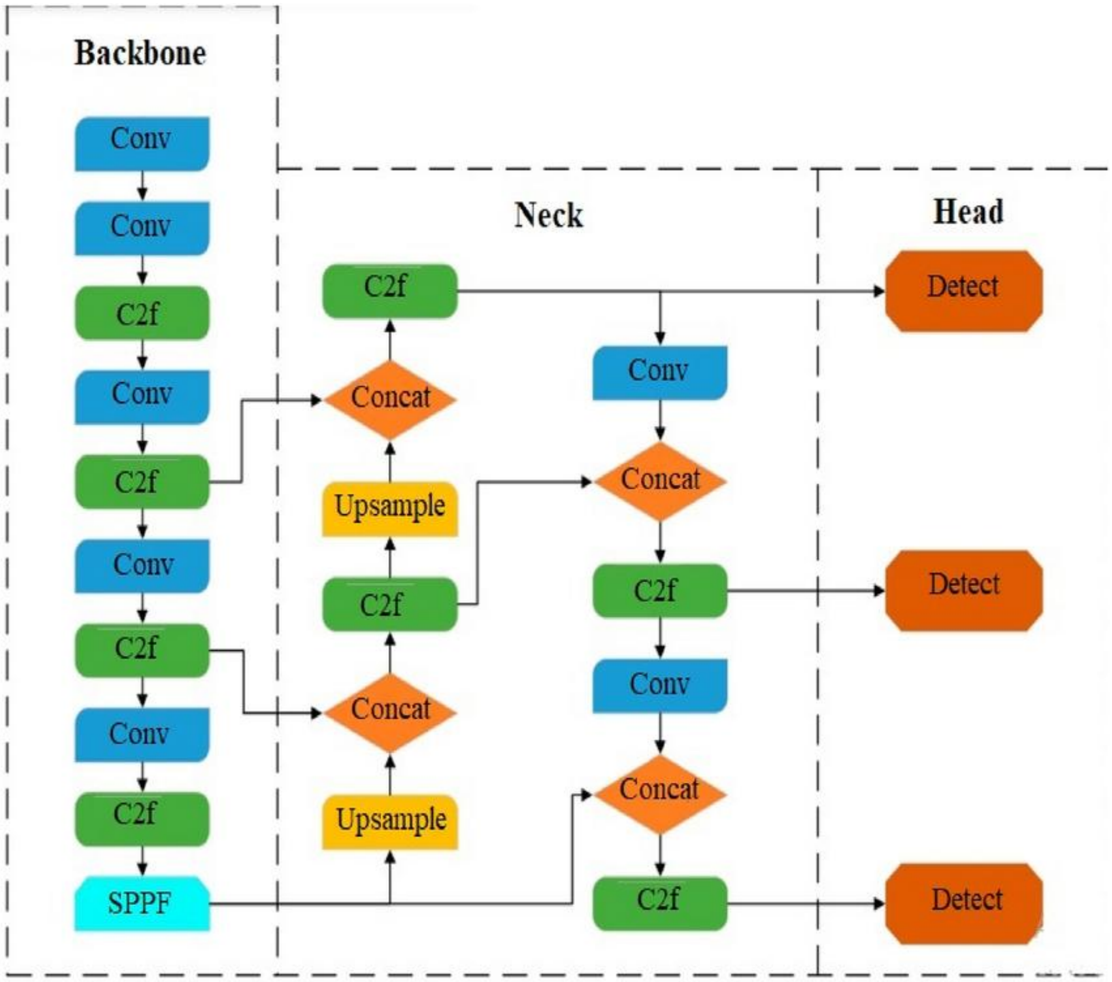



Figure 3.7: YOLO v8 Architecture (Medium)

### b) YOLO v9

YOLOv9 is an expanded version of the YOLO family, and it can operate more effectively in both speed and error, which is most appropriate in practice in real-time to determine the number plates of vehicles. The architecture has three significant elements which are Backbone, Neck, and Head. The Backbone depicts low-level characteristics and dimensionality reduction of space of non-reversible connections which are major visual details. The Neck perfects these features by reversible connections, which enhances flow and improvement of semantic features, which enhances the capacity of the model to learn challenging spatial relationships. The Head then makes the final predictions and the results are bounding boxes and class labels of the detected objects. The architecture has enabled YOLOv9 to be efficient when it comes to processing and identifying vehicle number plates with high accuracy and speed and therefore is a powerful choice as far as real-time detection operations are concerned.

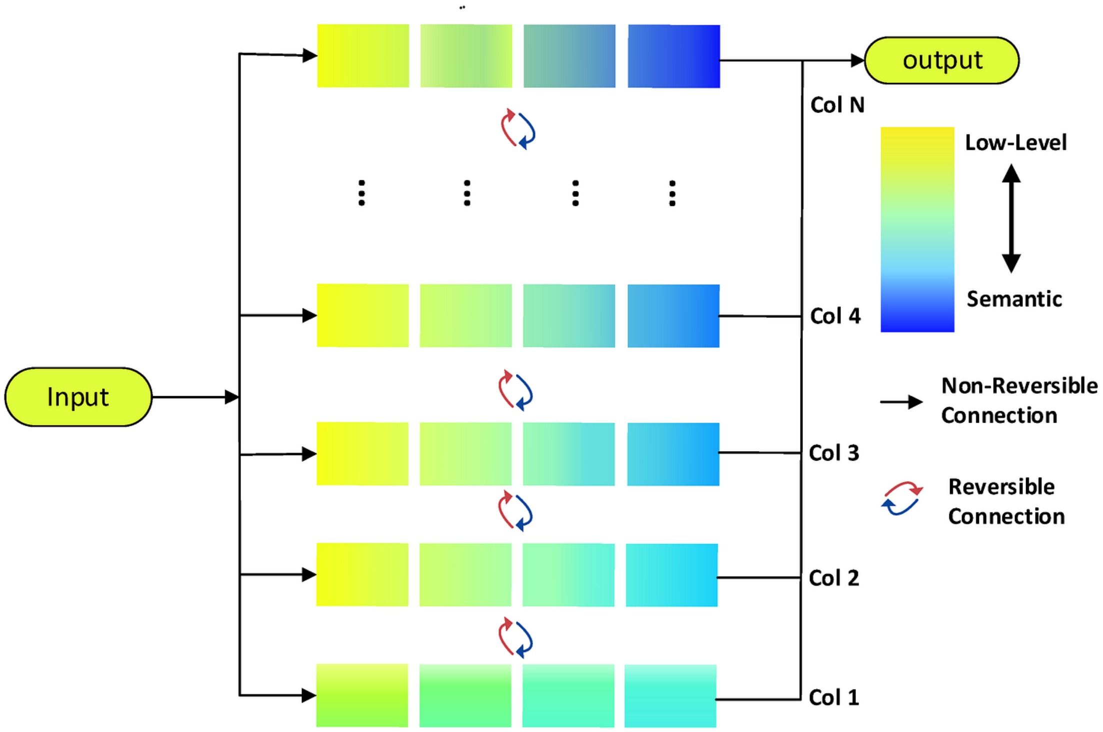


Figure 3.8: YOLO v9 Architecture (Medium)

### c) YOLOv10

In the case of YOLOv10 architecture the approach used is a dual-label assignment strategy, which is more effective in increasing the overall performance of detection through effective handling of both regression and classification. The architecture incorporates Backbone, PAN (Path Aggregation Network) and Heads which perform various tasks like classification and bounding box regression. The backbone is the feature extraction unit and the PAN enhances the flow of feature maps enabling it to efficiently use multi-scale feature maps to perform more accurate localization. The Dual Label Assignment method is unique as it divides the operation into one-to-many head (with multiple bounding boxes) and one-to-one head (with single detection boxes). This makes it very accurate and more effective in dealing with over-lapping objects. The Consistent Match Metric guarantees that the most suitable matching of predicted and ground truth bounding boxes is conducted, depending on the scores and intersection-over-union (IoU) values. YOLOv10 is the best choice because it is capable of offering precise object localization and has a shorter inference time, hence it can be used in a real-time vehicle number plate detection application.

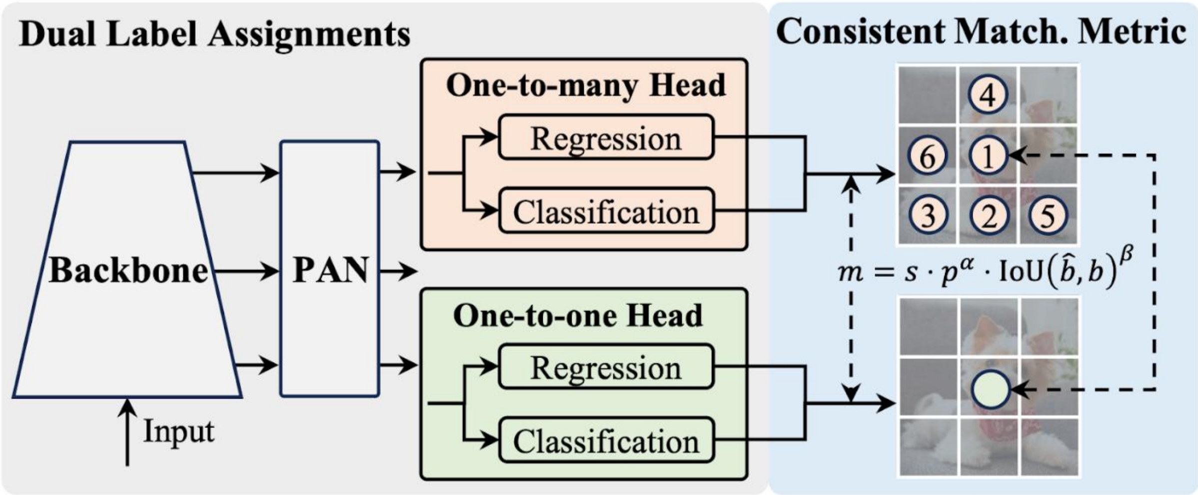



Figure 3.9: YOLO v10 Architecture (Medium)

### d) YOLO v11

The architecture of YOLOv11 is designed in a manner that it may be utilized to obtain the high-level spatial and semantic characteristics at low computational complexity to aid the real-time object detection chores. It uses modified backbone blocks including C2f, C3K2 and C3K blocks. It is also characterized by a convolution layer and division of features and bottleneck operations in the C2f block, which allows learning features to take place. C3K2 block uses stack 3x3 convolutions which are concatenated to boost features and adjust to scale. In the C3K block, layers of bottlenecks are introduced in order to reduce the complexity of calculations and preserve the accuracy. It will enhance localization and classification of images and hence more effective in high-resolution image detection that will be more useful in real-time image detection activities such as vehicle number plate detection.

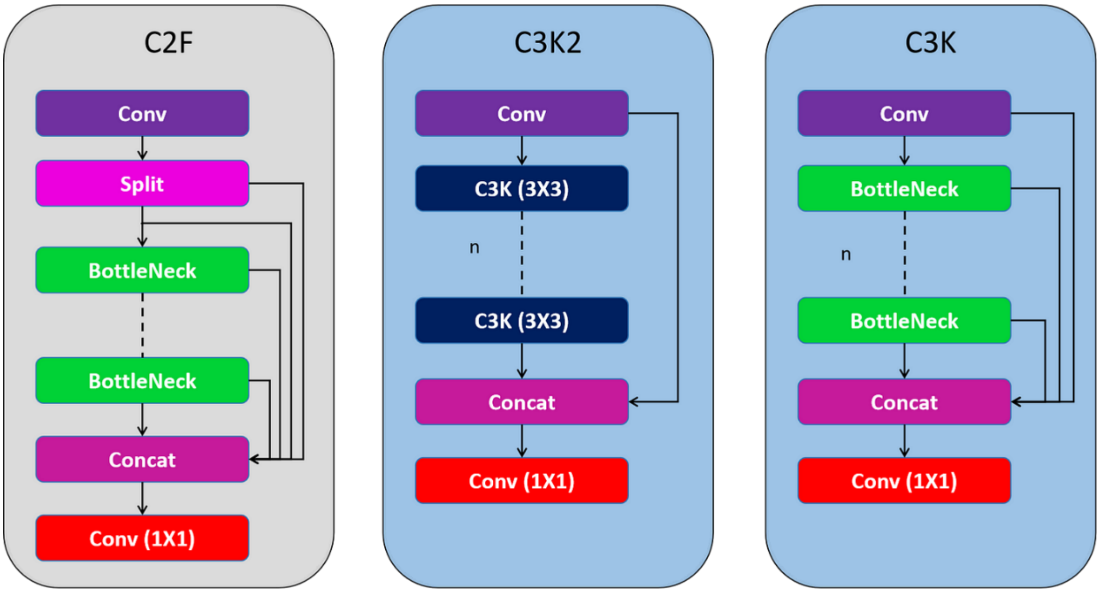



Figure 3.10: YOLO v11 Architecture (Medium)

### e) YOLO v12

YOLOv12 architecture uses an advanced backbone and neck architecture design to optimize outcome with respect to accuracy and efficiency in detection. It has got started with an Initial Conv Layer, C3K2 and R-ELAN blocks, which is necessary in the characteristics of extraction, which extracts the information in all the multi-scales and minimizes the number of parameters. The model also includes A2C2 layers that include Area Attention and convolution, which help the model to be more focused on the relevant areas in the image, particularly in the high-quality detection of the model in real-time usage. The Neck section enhances the properties by applying Upsample layers where they are concatenated as more sophisticated properties in the Detection Head and Class Head. These are supported by other concatenation and upsampling operations to provide efficient processing of the complex image characteristics. It is also incorporated into the architecture that generates both bounding boxes and segmentation masks and thus supports classification and pixel-level analysis, which makes YOLOv12 appropriate in such applications as the detection of vehicle numbers on their plates.

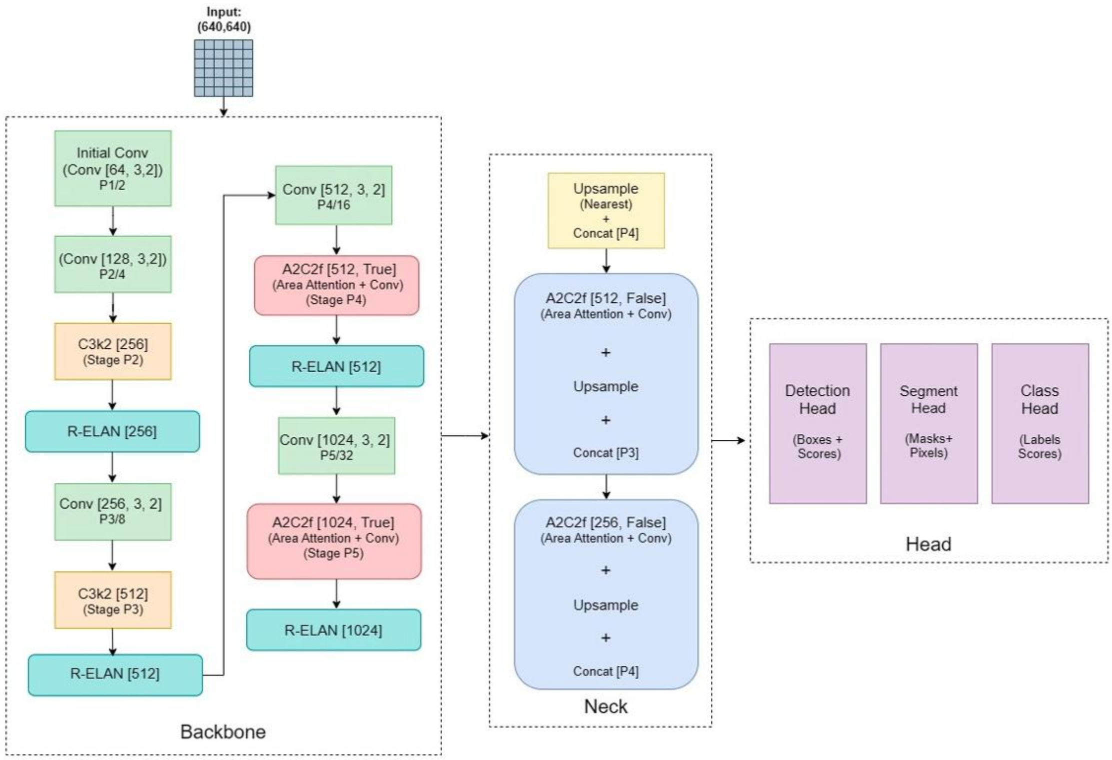



Figure 3.11: YOLO v12 Architecture (Medium)

## 3.2.4 Fine Tuning Strategies

A custom YOLOv8 model was used in this paper to fine-tune directly on this manually curated dataset to detect vehicle number plates. The model had 34 epochs with a batch size of 8 and an image size of 640x640. Since the dataset has been gathered and annotated manually, the model has learnt to identify vehicle number plates starting with no pre-trained weights. The last layers of the YOLOv8 model were trained over this particular dataset with the aim of maximizing the capacity of the model to identify number plates accurately. This method facilitated the specialization of the model in the task of number plate detection under diverse conditions, taking into consideration the peculiarities and differences of the manually annotated dataset.

## 3.3 Project Plan

Table 3.3: GANTT Chart of Project Timeline

| Process | Jan’25 | Feb’25 | Mar’25 | Apr’25 | May’25 | Jun’25 | Jul’25 | Aug’25 |
|---|---|---|---|---|---|---|---|---|
| Working Plan | | | | | | | | |
| Theoretical Study | | | | | | | | |
| Literature Review | | | | | | | | |
| Data Collection | | | | | | | | |
| Data Preparation | | | | | | | | |
| Yolo Model & experiment | | | | | | | | |
| Web-App development | | | | | | | | |
| Report Writing | | | | | | | | |
| Review and Finalization | | | | | | | | |

The project is in the form of a well-defined plan that is carried forward in different phases, all of which are essential for the accomplishment of the goal of building a vehicle number plate detection system on the basis of the YOLO models. Initial work plan and theoretical study will be used to guide the work that follows. The literature review stage will then be undertaken where appropriate existing research and technologies will be reviewed to ensure the project objectives are up to date in the field. Data collection will start in September 2024 and then data preparation for model training will start. The crux of the project will be the YOLO model training and experimentation phase, which is slated for December 2024 to February 2025. In parallel, the Web application development for the testing and validation of the deployment of the model is envisaged for the same period. As the model performance is increasing, reporting will start in March 2025, with final analysis and finalization of the project in April 2025. The Gantt chart below shows the major activities and their estimated 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. The model was implemented through an application of the Streamlit-based web application whereby users could interact with the system in real time to detect the vehicle number plate. Once the web was successfully deployed, the model was brought to TensorFlow Lite and encapsulated into a mobile application to be used in the real world. The results were then compared and assessed and a comprehensive report made. This thesis work entailed a lot of practical participation in all the stages which included the collection of data, developing the model, fine-tuning, deployment and performance analysis, such that every part of the project was completed entirely.

## 3.5 Summary

This research methodology aims at coming up with an effective, explainable, and deployable deep-learning-based vehicle number plate detection system. It starts with the collection of various vehicle number plate data on the roads, traffic areas and institutions to provide variation in lighting, environmental conditions, and the types of number plates. The dataset is processed with resizing of 640x640 pixels, normalization, and data augmentation to enhance the performance of the model and overcome the issues of class imbalance. This data is then divided into training (86%), validation (7%) and testing (7%) datasets. To develop the model, there are five types of YOLO architectures (v8m, v9m, v10m, v11m and v12m) that are implemented and trained to compare their performance in terms of vehicle number plate detection. All models are optimized to detect the accuracy using hyperparameters. M-Net is a customized lightweight CNN model, which is produced to meet the requirements of computational issues and enhance efficiency without accuracy degradation. Lastly, the model is finally made into TensorFlow Lite format and embedded into a mobile application to make it scalable and available to the real world. The methodology is clear and consists of all the steps in the data collection and preprocessing to the development of the models, fine-tuning, and deployment to provide a robust, complete, and useful vehicle number plate detection system.

# Chapter 4

# Implementation and Results

This chapter expounds in a step-by-step format with the environment setup, dataset preparation, number plate detection training and evaluation. This will involve focusing on hardware and software setting necessary, the preprocessing methods that datasets will require and the environments to run in to achieve reproducibility and to maximally use deep learning models.

## 4.1 Environment Setup

To allow the development of image classification deep learning models, training, and testing thereof, the following software tools, platforms, and configuration have been used:

**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 Kernets (Kaggle Notebooks)
- **GPU Used:** NVIDIA Tesla P100 (available on Kaggle platform)
- **Storage:** Kaggle Datasets (for storing datasets during training)

**Version Control & Collaboration:**

- **Git & GitHub:** For version control, collaboration, and tracking changes to the code

## 4.2 Testing and Evaluation

In order to compare the performance of deep learning models in the detection of the number plate in accordance with the usage of our developed interface, deep learning models like YOLOv8m, YOLOv9m, YOLOv10m, YOLOv11m and YOLOv12m were applied and contrasted on the same data and performance indicators. The models were evaluated on the basis of their ability to predict important bounding box in the vehicle picture.

**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%. |

A variety of crucial measures were employed in doing this including but not limited to precision, recall and mean Average Precision. Precision measures how the correct prediction is made by the positive prediction, and recall represents how the model can cover all the significant instances. To estimate multi-scale detection, mAP was measured at the threshold of 0.5 (mAP@50) and 0.5-0.95 (mAP at 50-95). Model performance was examined using additional plots, such as Precision-Confidence, Recall-Confusion and F1-Confidence curves. Combined analysis brings out strengths and weaknesses within the classes with precision-recall trade-offs to each object class and confidence level. The analysis of the five YOLO-based models (v8m, v9m, v10m, v11m, and v12m) in this section was conducted based on standard measures of the effectiveness of the models to recognize vehicle number plates. The following metrics to be used in the evaluation are Precision, Recall, mAP50 and mAP50-95.

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

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

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

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

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 | Images | Instances | Precision | Recall | mAP@50 | 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 displays the results of the five YOLO models (V8m, V12m, V11m, V9m and V10m) that were trained on car number plate recognition. V8m is the most precise (0.959), recall (0.922), mAP-50 (0.946), and mAP-50-95 (0.55) model and it can detect and classify vehicle number plates with greater accuracy. V12m and V11m are close after them with the precision and recall value of 0.943 and 0.941 respectively, and close mAP value of about 0.944 and 0.543, which indicates that they also perform very well. 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.

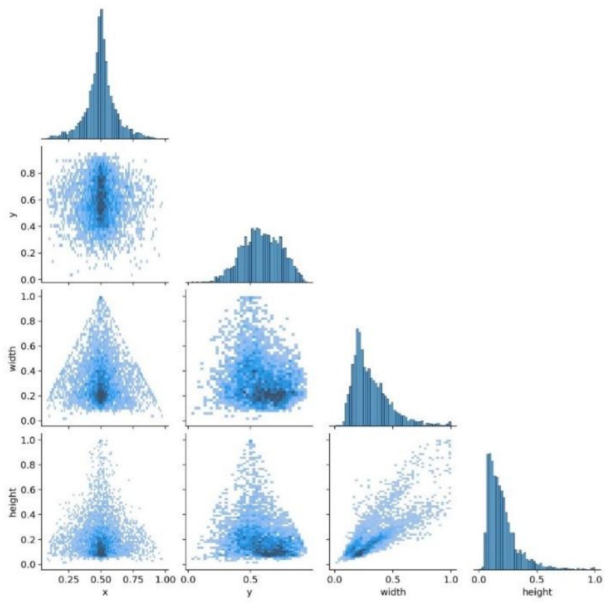
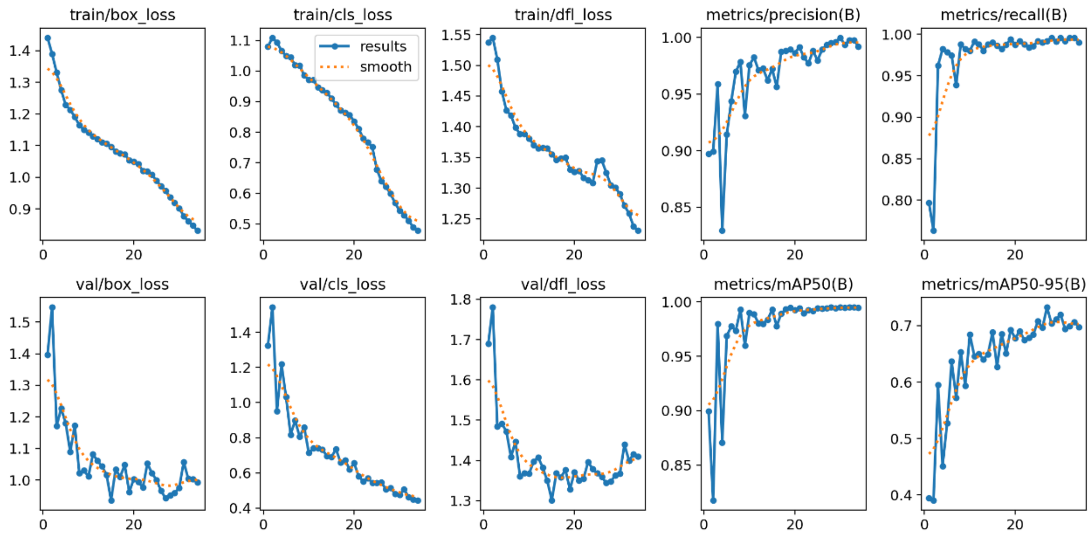
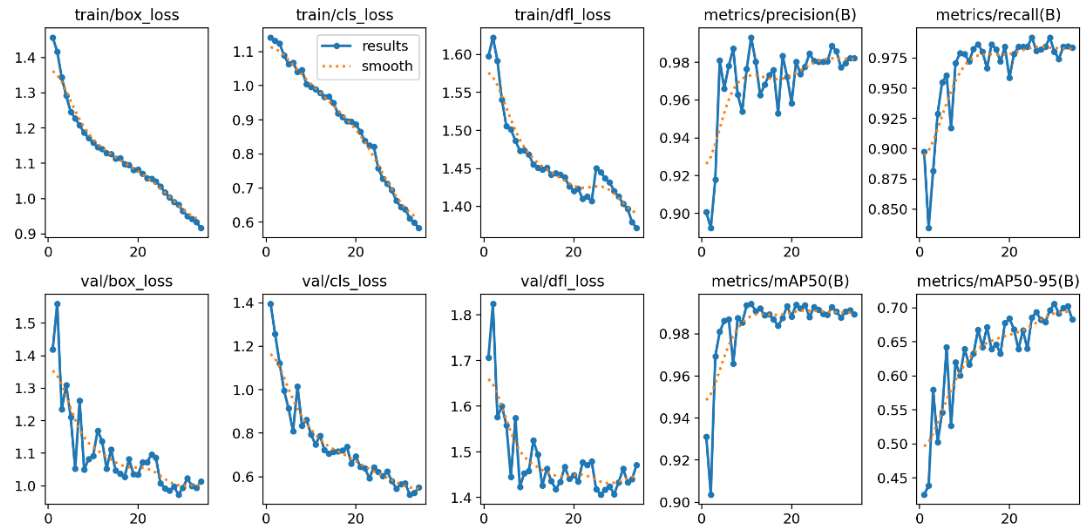



Figure 4.1: Labels Correlogram for the Dataset

The provided correlogram visualizes the relationships between the key variables of 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.



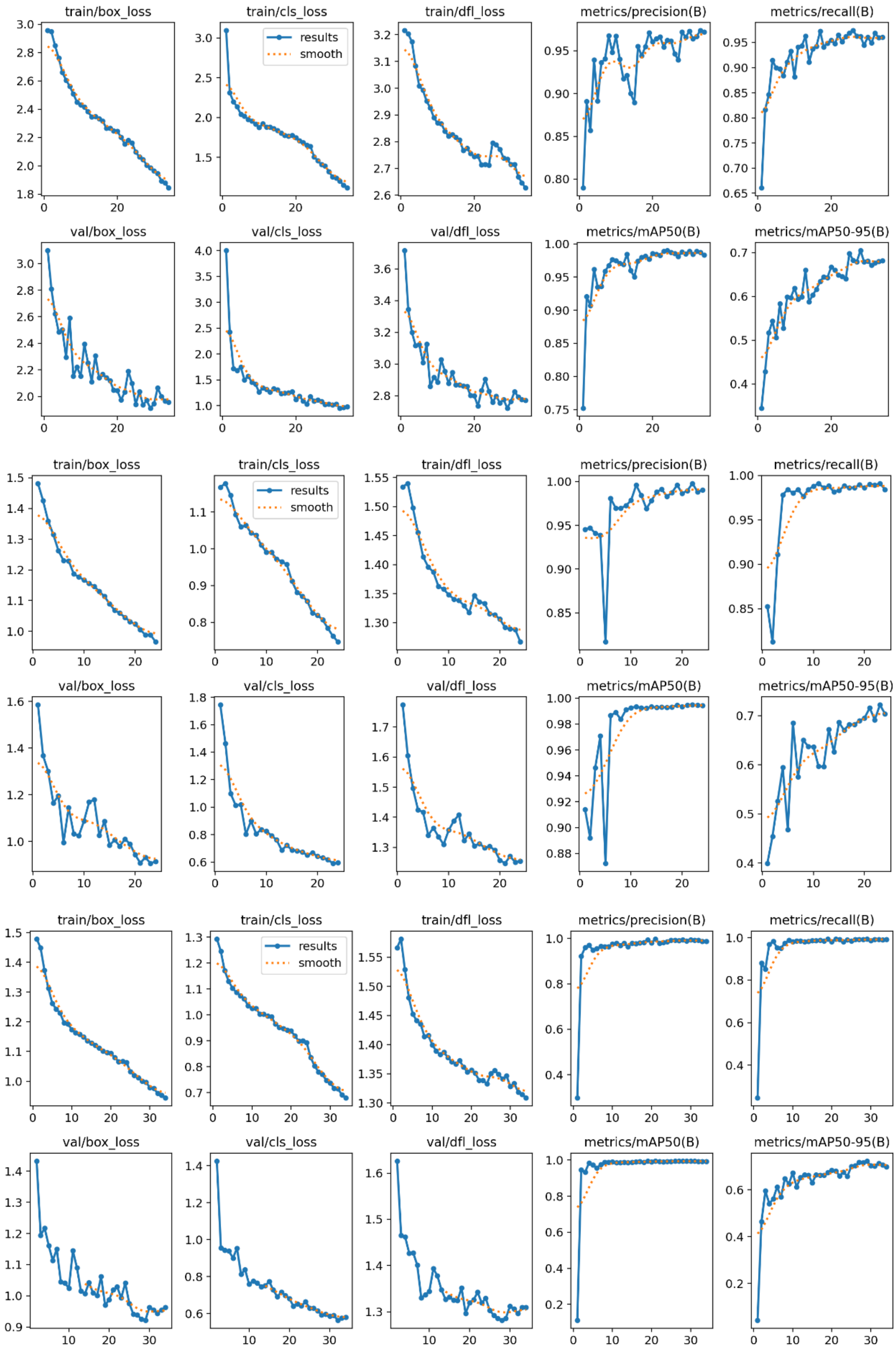
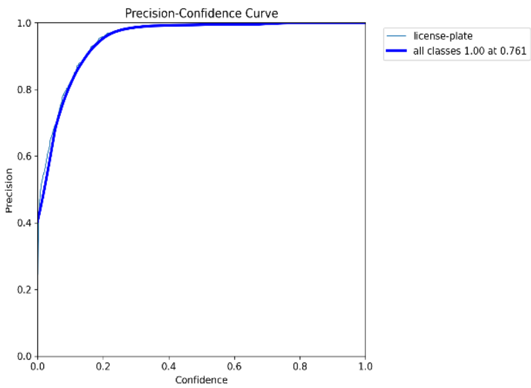
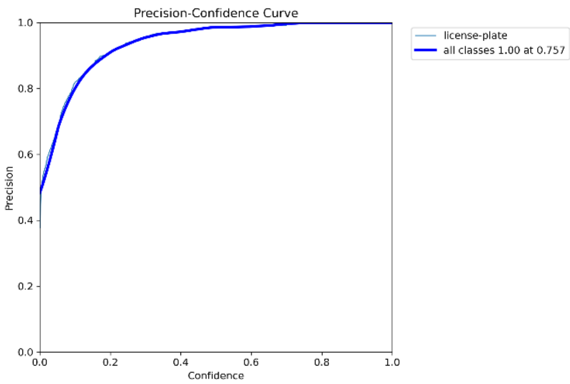


Figure 4.2: Training Loss, Validation Loss, mAP metrics of performed models

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. The extent of the loss reduction smoothness, however, differs among models with YOLOv12m being more stable and consistent in its decline than the previous models (YOLOv8m, YOLOv9m, and YOLOv10m). YOLOv12m has been observed to have more consistent loss reduction during training, which means that the model is more stable and optimized to learn. In comparison of validation losses, YOLOv12m is always more convergent and stable. This action indicates that the generalization of YOLOv12m is more efficient, as it is more effective when working with unseen data and can minimize the chances of overfitting. Meanwhile, YOLOv8m and YOLOv10m exhibit more variation in validation losses which can be interpreted to mean less efficiency in learning and more likely overfitting as training is continued. All models demonstrate improvements over the epochs in terms of the performance measures such as precision and recall. YOLOv12m has the best accuracy and recall that mean that it is more effective in object detection and classification. YOLOv12m is always the best model when it comes to detection, classification, and recall in all the epochs. In the same vein, it can be observed that the mAP 50 and mAP 50-95 metrics, which measure the overall performance of the model in terms of detection, have the greatest values with YOLOv12m at different Intersection over Union (IoU) thresholds. 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 compared to the other YOLO models.



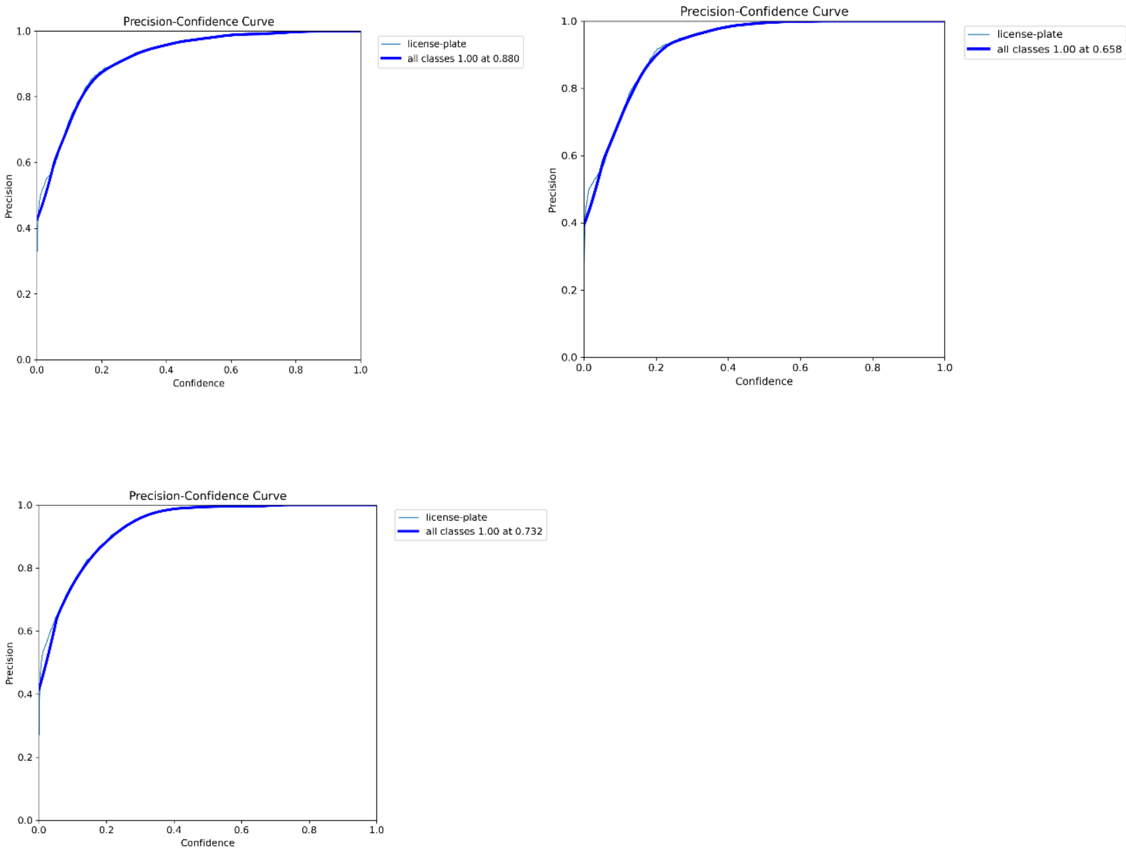



Figure 4.3: Precision-Confidence Curve Training on the performed model. Precision-confidence curves of YOLOv8-12m models represent a trade-off between models on precision and confidence at different thresholds. In the case of YOLOv8m, the curve shows the high precision value at the lower end of the confidence levels (at around 0.76) and slightly plateaus as the confidence levels increase which means that there is a good balance between detecting the true positives and reducing the false positives. In the same way, YOLOv12m is more uniformly performing, with a marginally higher accuracy at the same confidence level (0.76) and a smoother overall curve which may indicate better performance in reduction of false positives at the cost of different thresholds. YOLOv11m is also a good model that has a significant drop in the accuracy of low confidence with a hint of instability at the beginning. YOLOv9m has a lower confidence threshold (0.66) at which the precision is nearly perfect, but the curve decelerates more rapidly than others, indicating a certain degree of drop in consistency of the classification. Finally, YOLOv10m has a smaller precision at 0.73 confidence but grows with the curve though not as strongly as the others. The difference in accuracy in the models infers the difference in accuracy and reliability where the YOLOv12m has the most constant and with higher accuracy at lower confidence levels.

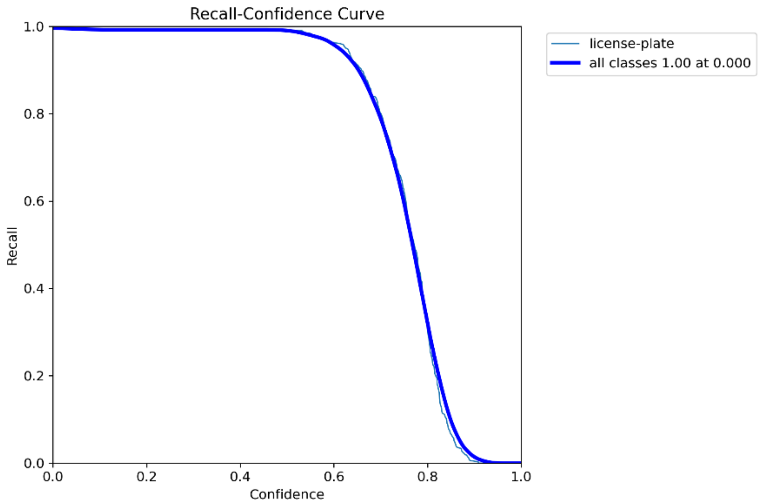



Figure 4.4: Recall-Confidence Curve of YOLO v8m

The recall-confidence curve of YOLOv8m indicates the correlation that the recall has with the confidence threshold, and it can be seen that the model can accurately detect positive cases. As we observe in the curve, the recall will be 1 (i.e. all true positives are called as true) as the confidence rises between 0 and approximately 0.7. Beyond this threshold, the recall begins to decrease very rapidly, which suggests that the higher the confidence level values, the lower the number of true positives are likely to be found, perhaps because of more stringent classification conditions. It implies that the accuracy of YOLOv8m could be in conflict with recall at elevated confidence levels, which is typical of object detection problems. Based on the precision and recall rates of Table 4.2, YOLOv8m has a high precision of 0.959 and recall rate of 0.922, which, once again, confirms its high performance overall in license plate and other classes detection. YOLOv8m has the highest values of both mAP at 50 (0.946) and mAP at 50-95 (0.55), indicating that it is the most balanced model in the series to use in this problem.

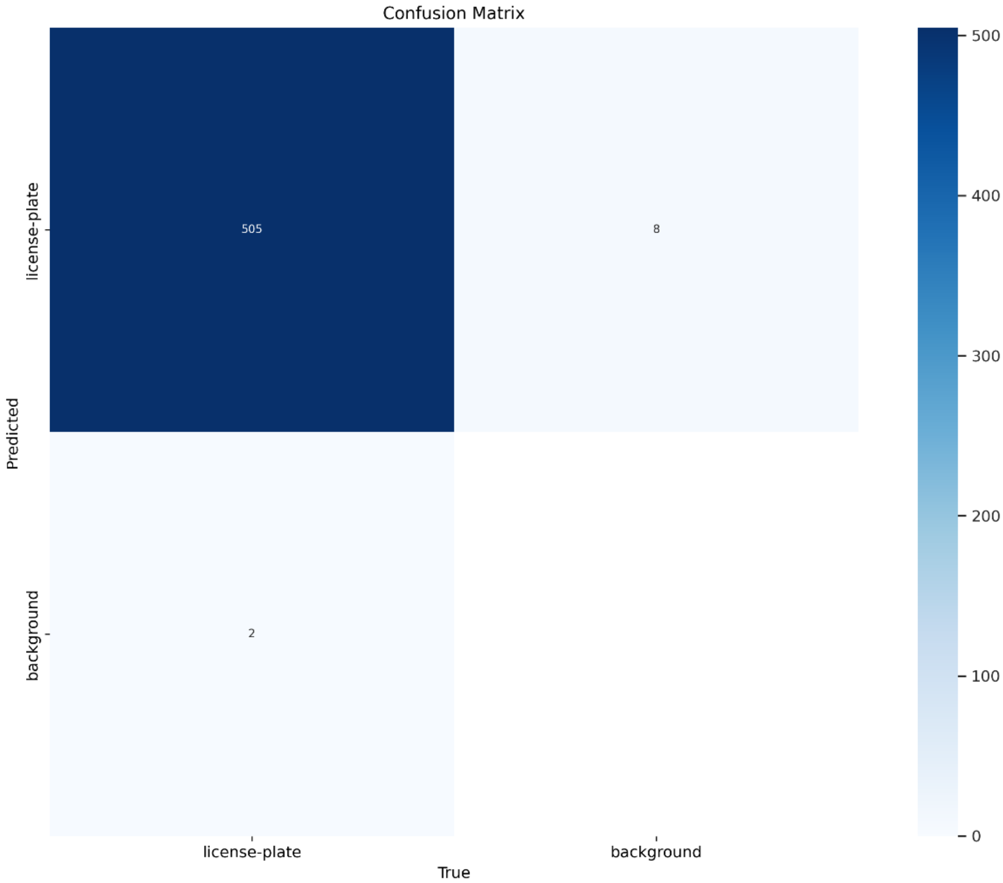



Figure 4.5: Confusion Metrics of YOLO v8m

The confusion matrix of the YOLOv8m model indicates its use in the prediction of the license-plate and background classes. According to the matrix, the model was able to describe 505 cases of license plates as a true positive (the model recognized it as a license plate) and 8 cases of license plates as a false positive (the model accepted the background as a license plate). The model further falsely classified 2 examples of actual license plates as background (false negatives), and none of the background as license plates (true negatives). This table demonstrates that YOLOv8m is very efficient in differentiating license plates and background and that the number of true positives is high and the number of false positives and false negatives is extremely low. With the precision of 0.959 and recall of 0.922, the model has shown a great ability to identify the license plate type with few errors. The results of the overall performance as shown in the confusion matrix are consistent with the large mAP values (mAP@50 of 0.946 and mAP@50-95 of 0.55), which indicate that YOLOv8m is better at correctly identifying and localizing license plates during training.


Figure 4.6: Validation result on sample data of YOLO v8m

As it can be seen in the image, the validation value of the YOLOv8m model is a good sign of the superior quality performance of the model among the others (v9m, v10m, v11m and v12m). The model has identified the license plate in the image and localized it right and has a high level of accuracy and precision. This implies that YOLOv8m will be powerful enough to recognize specific objects which can be license plates in the real world. Although the other models like v9m, v10m, v11m and v12m did not show a poor result, they were defined by the low preciseness and recall when it comes to the license plate recognition. The better results of YOLOv8m are justified by more significant mAP (mAP@50 of 0.946 and mAP@50-95 of 0.55) which indicates that it can be better used in the localization and classification problems. It is also indicated in the picture that the model is effective in recognizing the license plate and labeling it accordingly, which is significant when used in such tasks as the automatic number plate recognition (ANPR). The possibility of the YOLOv8m to be equally effective at both training and validation samples allows it to be used in the real-world scenario, where accuracy and precision are the most important factors of successful object detection.

## 4.4 Summary

Five YOLO models V8m V12m V11m V9m V10m were tested in the research to determine the effectiveness of the models in identifying and categorizing license plates in the real-life context. Table 4.2 displays the accuracy, recall, and mean Average Precision (mAP) values of all the models using the various evaluation metrics. The YOLOv8m model was the best performer with a precision score of 0.959 and recall score of 0.922, to obtain an mAP 50 95 at 0.946 and mAP 50 95 of 0.55. This model was found to be most accurate in classifying as well as localizing license plates in images which implies that it can perform detection tasks with minimum error. The precision-confidence and recall-confidence curves of YOLOv8m also confirm the same conclusion, as precision was high in all the confidence thresholds, indicating that the model is strong in identifying those objects with high confidence. Comparatively, the YOLOv12m model was performing marginally lower than the YOLOv8m, but its results were still good, with a precision of 0.943 and a recall of 0.91 resulting in an mAP@50 of 0.948 and mAP@50-95 of 0.544. Precision-confidence curve of YOLOv12m had the same trends, with an overall low performance comparing to YOLOv8m. In spite of the small gap, the capability of Generalization to different data was evident in YOLOv12m, and the performance was similar across different evaluation measures. YOLOv11m and YOLOv9m models were also effective models, as they had a precision score of 0.941 and 0.937, respectively. The models performed well in terms of recall rates (0.916 and 0.918) and were close to each other in terms of mAP at 50 (0.944 and 0.948). The mAP50-95 of these models was however a bit lower with YOLOv9m getting 0.5337 and YOLOv11m scoring 0.543. It means that both of these models were effective in licensing plates classification, but showed minor difficulties in recognizing smaller or more complicated objects in the photographs. Conversely, the performance of YOLOv10m was the lowest as the model had a precision of 0.923 and a recall of 0.888. Its mAP50 was 0.923 with mAP5095 of 0.486 which indicates that a lot can be improved particularly in the management of recall in case of difficult cases. The poorer accuracy and recall of this model may be explained by the fact that it has a smaller ability to process more complex or partial detections in difficult pictures. With a very low number of misclassifications (2 background misclassifications only), the model has correctly identified 505 instances of license plates in the confusion matrix of YOLOv8m, which is an excellent result. This is also confirmed by the precision-recall curves and training logs which depicts that YOLOv8m continuously attained higher scores of better training losses and quicker convergence as the loss curves portray within the training epochs. The low training and validation loss of YOLOv8m indicates a high performance with the low chances of overfitting. Overall, the results indicate that YOLOv8m offers the best balance between detection accuracy, speed, and robustness, making it the most suitable model for real-time license plate detection applications. Although YOLOv12m, YOLOv11m, and YOLOv9m show comparable results, YOLOv8m's higher precision and mAP scores make it the most reliable choice for deployment in practical scenarios.

# Chapter 5

# Engineering Standards and Design Challenges

The chapter presents the engineering standards that were observed in the tea leaf disease detection system development and describes the overall effects of the project on society, environment and sustainability. It also serves to emphasise on the ethical standards and adherence to the applicable software, hardware, and communication standards.

## 5.1 Compliance with the Standards

This project is also in line with the internationally accepted standards of engineering to provide compatibility, ruggedness, privacy of data, and realistic viability of software, hardware and communication parts. It focused on the use of open-source and available technologies and platforms, including smartphones to obtain data, Microsoft PowerToys to resize images, and Google Colab Pro to test models. This strategic selection supported efficient development, testing, and deployment of the explainable AI-based tea leaf disease detection system.

### 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.

There are a number of alternatives that were factored in the development process. Java was considered as a backend development language because it has strong object-oriented platform and large ecosystem but its wordy syntax and slow experimentation time rendered it less appropriate, particularly in deep learning integration. Kotlin was also considered as the mobile application due to modern features and short syntax which are mainly suitable in Android, but it presents a higher learning curve and less community support when it comes to machine learning-based applications. An iOS app with Swift provided superior performance to the devices and more streamlined user experience; however, it was also concluded to be less user-friendly in the rural regions of Bangladesh because of the expensive Apple devices and their limited market share.

The backend model was developed with Python because it has a broad capabilities of machine learning libraries (Tensorflow, Keras, OpenCV), is simple to debug, and has rapid prototyping. Kaggle was selected as a test based on the availability of GPUs and a ready to go environment at a cost that was affordable. Android Studio was considered to develop an app because Android hardware is dominant among farmers in the targeted areas and the system is more accessible, and the system is viable when applied to the real-life scenario.

### 5.1.2 Hardware Standards

It was even created to be compatible with regular consumer-grade hardware, and was not required to utilize an expensive graphics card or AI-specific hardware. It has been adhering to the main hardware standards to be reliable and efficient. As an example, IEEE 802.11 (Wi-Fi) supports optional internet connectivity which allows activities like updating the model and submitting data in any connected surroundings. Also, compliance with IEEE 1725 aids conscientious management of batteries in mobile gadgets, a crucial aspect when intending to utilize it over a long duration in the field. The system is also compatible with ARM architecture with the custom lightweight convolutional neural network model (M-Net) optimized to execute an efficient Android smart phone running on ARM Cortex-A processors.

To have an image, high-resolution images were taken with a smartphone (iPhone 12), which is a viable and cost-effective option. This approach would comply with IEC 60065 regarding the electronic safety and ISO 12233 regarding the quality of images, and it is guaranteed that the information gathered at the phase of the dataset development was safe and of high quality. The system ensured consistency in the quality of data and did not require costly DSLR cameras because it used the easily accessible smartphone technology, making it more accessible and scalable within limited resource settings.

Several hardware alternatives were considered for image capture and model deployment. A DSLR camera was initially evaluated due to its superior image quality and customizable settings; however, its high cost, bulkiness, and impracticality for frequent field use made it less suitable for this project. Raspberry Pi-based deployment offered an open-source, low-power, and customizable platform, but it posed challenges in terms of portability, setup complexity, and limited processing capabilities. The NVIDIA Jetson Nano was also considered for its edge machine learning potential and GPU acceleration, yet its higher cost and maintenance difficulty for non-technical users made it less feasible, particularly in rural or low-resource settings.

Smartphones were chosen for both image acquisition and app deployment due to their affordability, portability, and ubiquity in rural communities. To run on limited memory and processing power devices, like Android (low to mid-range) smartphones, the M-Net model was specifically optimised to run without special AI hardware. This guaranteed practicality and cost-effectiveness to the end users.

### 5.1.3 Communication Standards

The system adopts a hybrid system of communication, which is both offline and online to ensure there is flexibility in the areas with low connectivity to the internet. Where it can be connected then it communicates under the HTTP/ HTTPS standards of secure web communication. The structure of a RESTful API enables even a scalable communication between the client and the server which is structured in such a way that it provides an efficient exchange of information. To provide privacy to the user and integrity of the data, the TLS/SSL encryption is used during the process of communicating sensitive information such as feedback and uploading of pictures to the cloud centers.

A number of communication options were considered by the system. The MQTT protocol was an option because it could handle lightweight applications, is fast in performance and also does not consume a lot of bandwidth hence suitable in IoT but not in transmitting large payloads such as images and would therefore need a more complicated and tailored configuration. FTP ( File transfer protocol) was easy to implement and simple, but this was not encrypted and was considered insecure by the contemporary standards. The WebSockets offered two-way traffic in real-time, but this was not required in this application, and all the complexity of the approach was additional considering the fact that the system needed to work efficiently even when there is no internet connection.

The HTTP/HTTPS is selected as it is very compatible with mobile and web services, easily can be integrated with REST API, and it provides inbuilt encryption using TLS. It offers an effective and secure way of enabling optional cloud functions and the capability to utilize it in an offline manner. RESTful architecture is extensible and provides easier extension in the future, e.g. the addition of cloud-based analytics or farmer feedback loops.

## 5.2 Impact on Society, Environment and Sustainability

The chapter discusses the compatibility of the proposed License Plate detecting system with the overall objectives of the society and ethical obligations, and also dwells on the long-term sustainability of the system. The creation of an explainable, mobile-based AI case presents both promising perspectives on farmers and other stakeholders in the agricultural industry, as well as environmental protection and sustainable technological development. The system was to be inclusive to be accessible and effective in its non-technical aspects.

### 5.2.1 Impact on Life

We have a great impact on life through our work and in this case, it impacts the Bangladeshi vehicle number plate detection. We have developed a system that can greatly improve traffic surveillance, law enforcement and urban mobility since you have trained a deep learning-based model with YOLO to recognize the number plate in a more precise way. This can be used to its fullest capacity to track the vehicles, prevent illegal parking and facilitate the smooth flow of vehicles in the cities. It would involve a convenient UI, which implies its ease of use by non-technical users, which makes it easy to enable authorities and institutions to use it and practically implement it in their workflow. In addition, the system can assist in reducing the instances of human error and manual labor that is involved in vehicle checks and this will make the governance more effective. It will reduce traffic crimes and help to increase the highways safety at the accuracy supported by AI and real-time detection. Your solution will assist in digitizing the traffic systems in Bangladesh in the long-run, work on smart city programs, improve the adherence to the regulations, and make the process of law enforcement smarter and more automatized. Moreover, it will promote the use of technology in the traffic systems of Bangladesh and will assist in providing better vehicle management solutions to the same, which will also promote technological development in this country.

### 5.2.2 Impact on Society & Environment

You have a great influence on the society and the environment because your work has a direct effect on smarter and more sustainable urban infrastructure. With the use of the vehicle number plate with the application of the YOLO-based deep learning, your system will enhance the law enforcement and traffic control, which will lead to a reduction in the number of traffic offenses, vehicle tracking, and road safety. This is used to make the urban landscape more coordinated and thus ensures fewer congestions and there is smooth movement of transport in the urban places. On the social aspect, the technology improves safety to the common citizen since it enables the detection of stolen cars, illegal parking, and other crimes involving vehicles at a faster rate. It may also help in improving compliance with the parking policies and regulations, which will reduce the congestion in the city areas, and the commuting will be very effective by the general populace. Besides this, convenient UI is making sure that it is easily accessible to various stakeholders such as government agencies and local authorities and citizens, a factor that makes its adoption by such parties very inclusive. On the environmental front, your solution will help in the sustainable development of the city due to promote smart city development, which emphasizes to optimize resources, decrease emissions, and enhance general traffic efficiency. By means of a deep learning model that is executed on an efficient system, there is a potential that your model will reduce the usage of the standard resource-heavy methods of surveillance and thus will reduce the impact of the traditional means of traffic surveillance using manual processes on the environment. Moreover, through better traffic movement and minimization of congestion, your system has an indirect beneficial effect of lowering fuel consumption and consequent carbon emissions, which is a good impact on environmental sustainability.

### 5.2.3 Ethical Aspects

Ethics of your car number plate detector with the YOLO models are based on the concerns of privacy, transparency, fairness, and accountability. The system will be targeted at collecting information on the vehicles with explicit permission, which will ensure privacy as it is anonymized and secure data transfer will be used. Transparency concepts are also advocated with the integration of Explainable AI (XAI) approaches of decision making, i.e., Grad-CAM that provides the users with the understanding of the decision making process of the model and the confidence in the decision making process. The model is also put to test in fairness to avoid possessing biases so as to allow the model to perform uniformly in different types of cars and different environments. The element of accountability comes at an opportune time because the system should be applied within the requirements of the law and not where it is intended to spy on individuals. Lastly, the principle of non-maleficence ensures that the system is not hurtful, i.e. it does not cause wrongful detainments, and that has the measures that will rectify any of the mistakes, which is part of using AI technology responsibly and ethically. In addition, it does not store or transmit any user data without a user-consent and optional cloud connectivity is secured by secure HTTPS protocols. This keeps privacy of data 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. Deep learning libraries like TensorFlow and Keras are used to make sure that future upgrades and enhancements can be easily made without requiring significant changes in the system since it is now possible to add new types of vehicle plates or increase the detection accuracy. The system is modular in nature, which enables the flexibility to update it and add new functionality, including support of multiple country-specific number plate formats.
- Economic Sustainability: The solution also uses resources that are readily accessible including smartphones and open-source software hence the cost of implementing the solution is minimal. The architecture of the system is aimed at achieving such regional and large scale deployments even in resource intensive environments such as rural regions. Offline capabilities minimize data transmission and cloud computing, which are costly, and hence, the total cost of the operation and maintenance is lowered.
- Environmental Sustainability: The system helps in the efficient reading of vehicle number plates using energy efficient smart mobile devices that eliminate the reliance on cloud based infrastructure. This reduces the environmental impact that is associated with data transfer and server management. In addition, the solution will facilitate the use of green technologies in the transportation industry, particularly in the reduction of traffic monitoring emission by competent and proper license plate recognition.
- Social Sustainability: The system is also expected to be user friendly and easy to use even to those who are not techno savvy. This access facilitates the mass consumption making the technological world fairer. Moreover, the system assists in the improvement of road safety, traffic and security in the urban and rural areas through the recognition of vehicle number plate in a myriad of conditions.

## 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) | Jan 7, 2024 | Feb 7, 2024 | 4 weeks |
| Data Collection & Preprocessing | Feb 8, 2024 | Mar 5, 2025 | 4 weeks |
| Methodology Design | Mar 6, 2025 | Mar 30, 2025 | 3 weeks |
| Experimentation with CNN Models | Apr 1, 2025 | Apr 14, 2025 | 2 weeks |
| Experimentation with RNN Models | Apr 15, 2025 | Apr 28, 2025 | 2 weeks |
| Development of fine-tuned YOLO modl | May 1, 2025 | May 20, 2025 | 3 weeks |
| XAI Integration & Model Evaluation | May 21, 2025 | Jun 10, 2025 | 3 weeks |

| | | | |
|---|---|---|---|
| Web Application Deployment | Jun 11, 2025 | Jun 20, 2025 | 1.5 weeks |
| Final Documentation & Printing | Jun 21, 2025 | Jun 28, 2025 | 1 weeks |

The Mobile Application Deployment stage was very important to make the final output in practical and easy-to-understand form. In this phase the trained and validated lightweight YOLOv8m model was implemented into an easy-to-use Android application. This app was first prototyped using Streamlit (web-based demos) then moved to Android Studio (native mobile performance). In particular, the inference time, the user interface design, the offline capability, and the compatibility with low-end smartphones widely available in tea-growing areas were optimized. Usability and performance tests were conducted with many different devices to ensure the usability and performance in real-world conditions. By following this structured timeline and modular project plan the research moved efficiently and smoothly through all critical stages - leading to a robust, efficient, and interpretable mobile-based disease detection system 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. Since the research was directed by the independently designed and academically rather than commercially oriented goals, a cost-efficient strategy was employed, involving the use of personal funds and free open-source software and other cheap cloud-based solutions such as Kaggle. These options saved a considerable amount of money and, at the same time, provided a high quality of experimentation and deployment. But in order to know whether it is financially viable to implement or commercialize on a large scale, another scenario of budget was also provided. The potential costs proposed in this situation include renting high plot, commercial cloud services, buying smartphones to be deployed, and outsourcing professional UI/UX design and labeling datasets.

The motivation behind the economic approach was informed by the fact that a more affordable and scalable solution was to be developed, particularly among rural agricultural populations where the technology interventions were to be affordable. Moreover, a prototype revenue model was considered, and the prospects of a freemium mobile app and licensing were discussed. This two sides view of budgeting makes sure that there is a balanced idea of academic viability and future business viability.

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

| Component | Estimated Cost | Remarks |
|---|---|---|

| | (BDT) | |
|---|---|---|
| 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 shows a high cost efficiency, academic oriented project, personal equipments, open-source solutions and internet sources. The project can be duplicated in other learning settings without having to spend significant amount of capital.

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

In order to achieve the completion of the present research project successfully, a variety of factors of complex engineering problem-solving were touched upon that included the choice of algorithm and model integration, mobile optimization and real-time implementation. The next mapping illustrates the correspondence of the work to the Engineering Problem (EP) framework:

Table 5.4: Mapping with complex problem solving.

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

**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 agriculture 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. 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.

# Chapter 6

# Conclusion

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

Although positive outcomes are achieved, the research has limitations. To begin with, the dataset despite the augmentation remains relatively small when compared to other large-scale global datasets and this could influence the external validity of the model. The models have been trained on photos of particular traffic areas and facilities in Bangladesh, hence they might not be best when they are implemented in different areas with different vehicle number plates styles or environment. Besides, despite the good performance of YOLOv8m, issues like occlusion, changing lighting, and blurred images may affect the model performance in real-life contexts. High quality image data is also required in the system, which is not always found in live surveillance settings. Also, the computational capabilities needed to implement real-time number plate recognition can be a limitation to scaling of the system, especially in resource-intensive environments.

## 6.3 Future Work

Future work in this direction would involve expanding the dataset to include a wider range of vehicle number plates from various locations so that the model can be more robust and generalizable. Incorporation of other environmental factors such as varying weather conditions, varied camera angles, and higher-speed motion of vehicles would also make the model more practical for application in real-world scenarios. To go beyond the computational issues, further optimization of the YOLO models can be done in a manner such that processing efficiency can be improved without compromising the accuracy. Additionally, adding the model within a larger system, i.e., automated vehicle monitoring or violation detection, can expand the productive uses of this technology. Finally, the study of hybrid methods that combine YOLO with other AI tools, such as OCR devices, has the potential to improve character recognition accuracy for challenging 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] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11]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.

[12]Jamtsho, Y., Riyamongkol, P., & Waranusast, R. (2021). Real-time license plate detection for non-helmeted motorcyclist using YOLO. *Ict Express*, *7*(1), 104-109.

[13]Chung, M. A., Lin, Y. J., & Lin, C. W. (2024). Yolo-sld: An attention mechanism-improved yolo for license plate detection. *IEEE Access*.

[14]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.

[15]Jamtsho, Y., Riyamongkol, P., & Waranusast, R. (2020). Real-time Bhutanese license plate localization using YOLO. ICT Express 6 (2): 121–124.

[16]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.

[17]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.

[18]Beratoğlu, M. S., & Töreyin, B. U. (2021). Vehicle license plate detector in compressed domain. *IEEE Access*, *9*, 95087-95096.

[19]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.

[20]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.

[21]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).

[22]Maheshwari, V., & Rajput, A. (2022). Using the Ocr Method with Yolo V2 to Detect Non-Helmeted Riders and Get License Plate Numbers.

[23]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.

[24]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.

[25]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.

[26]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*.

[27]Kanakaraj, S. (2021). *Real-time Motorcyclists Helmet Detection and Vehicle License Plate Extraction using Deep Learning Techniques* (Doctoral dissertation, Dublin, National College of Ireland).

[28]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.

[29]Lubna, Mufti, N., & Shah, S. A. A. (2021). Automatic number plate Recognition: A detailed survey of relevant algorithms. *Sensors*, *21*(9), 3028.

[30]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.

[31]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.

[32]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.

[33]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 (IICAIET)* (pp. 1-6). IEEE.

[34]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.

[35]Patel, C., Shah, D., & Patel, A. (2013). Automatic number plate recognition system (anpr): A survey. *International Journal of Computer Applications*, *69*(9).

[36]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.

[37]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.

211-15-4040

ORIGINALITY REPORT

| 20% | 11% | 14% | 9% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | Submitted to Daffodil International University<br>Student Paper | 4% |
| 2 | www.mdpi.com<br>Internet Source | 1% |
| 3 | S.P. Jani, M. Adam Khan. "Applications of AI in Smart Technologies and Manufacturing", CRC Press, 2025<br>Publication | 1% |
| 4 | irisa.fr<br>Internet Source | <1% |
| 5 | Songhua Zhang, Xiuling Lu, Zhangjie Lu. "Improved CNN-Based CatBoost Model for License Plate Remote Sensing Image Classification", Signal Processing, 2023<br>Publication | <1% |
| 6 | Pushpa Choudhary, Sambit Satpathy, Arvind Dagur, Dhirendra Kumar Shukla. "Recent Trends in Intelligent Computing and Communication", CRC Press, 2025<br>Publication | <1% |
| 7 | 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", Traitement du Signal, 2024<br>Publication | <1% |