# REAL TIME BANGLADESHI VEHICLE TYPE RECOGNITION USING YOLOv9 VARIANTS

**BY**

**SHAKIL MIA**
**ID: 221-15-5595**

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

Supervised By

**Dr. Naznin Sultana**
Associate Professor
Department of CSE
Daffodil International University

Co-Supervised By

**Mr. Narayan Ranjan Chakraborty**
Associate Professor and Associate Head
Department of CSE
Daffodil International University

**DAFFODIL INTERNATIONAL UNIVERSITY**

**DHAKA, BANGLADESH**

**JANUARY 2025**

# APPROVAL

This Thesis titled "**REAL TIME BANGLADESHI VEHICLE TYPE RECOGNITION USING YOLOv9 VARIANTS**", submitted by Shakil Mia, ID No: 221-15-5595 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 13 January, 2025.

## BOARD OF EXAMINERS

______________________ **Chairman**

**Ms. Nazmun Nessa Moon**
**Associate Professor**
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

______________________ **Internal Examiner**

**Dewan Mamun Raza**
**Assistant Professor**
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

______________________ **Internal Examiner**

**Mr. Abdullah Al Mamun**
**Lecturer**
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

______________________ **External Examiner**

**Dr. Md. Manowarul Islam**
**Associate Professor**
Department of Computer Science and Engineering
Jagannath University

## DECLARATION

I, therefore, declare that this undertaking has been finished by me under the supervision of **Dr. Naznin Sultana,** Associate Professor, Department of CSE, Daffodil International University. I further declare that neither an application nor an educational grant has been made anywhere for this thesis or any part of it.

**Supervised by:**

Sultana

**Dr. Naznin Sultana**
**Associate Professor**
Department of CSE
Daffodil International University

**Co-Supervised by:**

for
Sultana

**Mr. Narayan Ranjan Chakraborty**
**Associate Professor and Associate Head**
Department of CSE
Daffodil International University

**Submitted by:**

Shakil

**Shakil Mia**
ID: 221-15-5595
Department of CSE
Daffodil International University

## ACKNOWLEDGEMENT

First, I express my heartiest thanks and gratefulness to Almighty God for His divine blessing, making it possible for me to complete my final year thesis successfully.

I am really grateful and wish to express my profound indebtedness to **Dr. Naznin Sultana**, **Associate Professor**, Department of CSE, Daffodil International University, Dhaka. Her deep knowledge and keen interest in the field of “Deep Learning” have been invaluable in carrying out this thesis. Her endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, and meticulous review of drafts at every stage have made it possible to complete this work.

I would like to express my heartiest gratitude to **Dr. Sheak Rashed Haider Noori**, **Professor & Head**, Department of CSE, for his kind help in completing my thesis. I also extend my gratitude to the other faculty members and staff of the CSE Department of Daffodil International University.

I would like to thank my course mates at Daffodil International University, who participated in discussions during the completion of the course work.

Finally, I must acknowledge with due respect the constant support and patience of my parents.

# ABSTRACT

Vehicle type recognition is important for supporting traffic management and urban planning, as Bangladesh sees rapid growth of vehicular traffic. In this study, I provide a dataset and methodology for building a Bangladeshi vehicle type recognition model using YOLOv9 variant. The dataset gathered from traffic signal points in Bangladesh contains 281 images belonging to 12 vehicle classes which has been augmented to 402 images by techniques such as image augmentation (eg; horizontal flipping, adjusting brightness, etc). I applied preprocessing steps like auto orientation, resize to 256x256 and histogram equalization to improve data quality. I trained Google Colab YoloV9-C, YoloV9-E, and YoloV9-Gelan C with batch size of 32, for 100 epochs, then evaluated them based on mean average precision (mAP). Among all the models, YoloV9-E could achieve the best mAP of 73.66%, which indicates that it was able to perform well in real-time vehicle detection. Based on these insights, the trained models were deployed on Streamlit for testing in real-world Bangladeshi traffic environments.

## TABLE OF CONTENTS

## LIST OF FIGURES

## LIST OF TABLES

# CHAPTER 1

# Introduction

## 1.1 Introduction

In developing countries like Bangladesh where traffic patterns are very diverse and complex, real time identification and classification of vehicles is a critical challenge for traffic management, urban planning and intelligent transportation systems. Despite improvements in vehicular traffic growth, traditional approaches to traffic monitoring and vehicle recognition cannot maintain the speed and accuracy as demanded. However, due to the fact they can perform real time object detection and high precision with advanced deep learning models, particularly in the You Only Look Once (YOLO) family, they have solved the above challenge effectively.

Designed according to distinctive regional traffic conditions, e.g. vehicle detection in wild environments as described by B. Saha et al. [1], demands specialized systems. But because of the presence of a unique mix of cars including bicycles, rickshaws, buses, and trucks in Bangladesh that are very different from the cars that are commonly used in the datasets of developed countries, the generic vehicle detection models that are typically trained can be inadequate to deal with such situations. The development of effective performance calls for the usage of localized datasets and optimized detection algorithms that are tailored to each region.

For license plate recognition and object detection in autonomous driving, YOLO based models have become popular for their real time detection, mostly due to the high computation load. U. YOLOv8 is shown by Saha et al. [2] to be effective for Bangla license plate recognition where it adapts to localized contexts. YOLOv8 was able to detect accurately in situations where traditional models choked on variations within language and vehicle characteristics by utilizing transfer learning. It is a clear indication that YOLO variants may handle complex Bangladeshi traffic systems like none other.

Consistent further advances of vehicle detection have been made through the creation of specialized datasets, for instance the Bangladeshi Autonomous Driving Object Detection Dataset (BadODD) [3] by M. N. Baig et al. Specifically designed for Bangladeshi traffic, this dataset caters for the inherently opposing traffic flows, non-uniform vehicle models and unreliable lighting conditions. Trained with such datasets, these models achieve state of the art high accuracy on real world problems.

YOLO models are great for real-time applications, but resource limitations on edge devices can often become a big issue. A. To overcome this, Saadeldin et al. [4] developed a lightweight YOLOv8n model coupled with DeepSORT for vehicle counting on an edge base. The study showed that computationally efficient YOLO models could achieve high accuracy, and therefore are appropriate for deployment in resource constrained environments like Bangladeshi traffic signal points.

In this work, I develop and evaluate three YOLOv9 variants, YOLOv9-C, YOLOv9-E, YOLOv9-Gelan C, for real-time vehicle recognition using 12 vehicles in Bangladeshi traffic. According to the study, it uses a preprocessed and annotated localized dataset gathered from traffic signal points. I demonstrate the practical applicability of these models through high detection accuracy and their deployment using frameworks such as Streamlit. This work extends the growing field of intelligent transportation systems by proposing a holistic framework for real time vehicle type recognition in Bangladesh taking into account the technological and contextual challenges associated with localized traffic environments.

## 1.2 Motivation

The fast urbanization and excessive vehicular traffic in Bangladesh has caused a requirement of sustaining proper traffic management systems. Vehicle detection and classification in traditional methods fails to address the complexities of Bangladeshi traffic which comprises a wide variety of vehicles, like rickshaw, leguna, easy bike taxi etc. in addition to the conventional vehicles like car and truck. However, most global detection models are trained on data from developed countries, and this diversity together with the chaotic, unstructured nature of traffic makes the detection extremely challenging. The YOLO family of deep learning models that have skyrocketed in popularity given the rise of deep learning models in general, have simultaneously brought real time object detection and classification to life. To date, however, their application to Bangladeshi traffic scenarios has been underexplored. Meanwhile, localized solutions will be the bridge for this gap by leveraging datasets specifically designed for Bangladesh's traffic environment. The reasons motivating this research stem from the ability of the YOLOv9 variants to tackle such problems through their superior ability to detect, use little computational resources, and remain applicable across different traffic conditions. To promote traffic safety, smart city development, and efficient transportation planning, enabling millions of people in Bangladesh to live better lives, i develop a robust vehicle type recognition system.

## 1.3 Rationale of the Study

With increasing urbanization and vehicular congestion, Bangladesh currently requires efficient traffic management and monitoring. Traditional approaches to passenger vehicle recognition have been limited in countering the peculiarities of Bangladeshi traffic as it comprises rickshaws, CNGs, horse carts and modern cars, buses and trucks. The need for a localized, real time vehicle recognition system exists to compensate for inconsistent patterns of traffic and the complicated deployment environment.

Today, deep learning models, especially YOLO variants have shown their capability in object detection and classification. While there has been little explored application of these principles to Bangladeshi traffic scenarios, potential gains are significant. Previous studies have been conducted with generic (global) or international data sets that are not very trustworthy because most of the unique vehicle types and training conditions in Bangladesh are not captured. This gap can be bridged with the development of a localized dataset, integrating state of the art YOLOv9 variants which would offer a robust solution for the condition in the area. The main intention of this study is to deal with the issues of on time vehicle identification in Bangladesh by hitting up a system upgraded for the country's exceptional traffic condition. This research contributes specialised dataset, advanced YOLOv9 models deployment, and ensure real world applicability, which will provide important insights and useful tools for better traffic management and urban planning in Bangladesh.

### 1.4 Expected Output

The objectives of this study are to produce an accurate real time vehicle type recognize system, which can recognize and classify 12 different types of vehicle of Bangladeshi traffic situation. The system is designed to accommodate a wide range of vehicle types including bicycles, bikes, buses, cars, CNGs, easybikes, horsecarts, legunas, rickshaws, trucks, vans, and wheelbarrows, commonly found on Bangladesh' s urban and rural roads.

Specifically, i believe that the YOLOv9 variants (YOLOv9-C, YOLOv9-E and YOLOv9-Gelan C) will achieve very good detection accuracy and efficiency in vehicle classification under different traffic conditions. Preliminary testing suggests that the models will vary in performance, with YOLOv9-E likely to produce the best mean average precision (mAP). It is expected to have real time performance, images are processed efficiently with negligible latency that would allow the system to be deployed at traffic signal points for monitoring purposes.

In terms of accuracy, I expect 70% mAP across vehicles in all categories, with additional fine tuning and testing on the way to potentially increasing this accuracy. It is anticipated that deploying trained models using Streamlit will enable seamless integration and real-time testing, making the models ready to serve in urban traffic management and intelligent transportation systems for Bangladesh.

### 1.5 Report Layout

The distinctive features of my endeavor are as follows:

**Chapter 1:** The intro presents the historical and contextual information of the research topic and also investigates the challenge or query of the research and goals and relevance of the study. Finally, 1.5 introduces the format of this paper, 1.1 the introduction, 1.2 the inspiration of the subject under study, 1.3 the justification for the study to be conducted and 1.4 the expected results of this article.

**Chapter 2:** The background study contains an initial appraisal which gives a brief recap of the research carried out on this topic. The literature describing the applicable intelligence technology research is given here. In addition, the size of the issue was shown by the challenges i encountered performing this study. For the purpose of the paper 2.1 describes the areas i will look at for the paper, 2.2 shows the related works of the scientist that display the scientist's prior work, 2.3 is a comparative evaluation and summary of the topic, 2.4 talks about the overall goal of the paper and 2.5 describes the problems i will face.

**Chapter 3:** This section has covered the general concepts of how to handle a data set and also how to generate the model. In 3.1, the research approach is described, 3.2 - the structure of assembly of the dataset, 3.3 - the sterilization of the dataset, 3.4 - the preliminary dataset processing strategy, 3.5 - the recommended approach to the dataset and 3.6 – machine and human implementation prerequisites.

**Chapter 4:** In this section i assessed and looked into the output of my predictive framework. It is all the results from the graphical description for ease of understanding. The evaluation of this paper and experimental finding are covered in this section. experiment results and analysis in 4.2, training and validation precision and loss curve in 4.3, and result analysis in 4.4, and the discussion in 4.5,and the introductory part in 4.1.

**Chapter 5:** Finally, 5.1, 5.2, and 5.3 address impact on society, impact on environment, ethical aspects.

**Chapter 6:** This section provides the accomplished study in accordance with 6.1, 6.2 and 6.3, its conclusion, limitations and future studies.

# CHAPTER 2

# Background Study

## 2.1 Terminologies

For understanding of methodologies and systems used in this work of vehicle recognition using deep learning models, several key terminologies are used in the context of vehicle recognition. Object detection is a fundamental concept, which consists of the task to spot and pinpoint objects in an image or video. This is used in vehicle recognition, where i need to not only classify each vehicle type, but also localize this within the frame perfectly. Closely related to the YOLO (You Only Look Once) is one of the most popular deep learning architectures for real time object detection. The idea is to perform all of these at once, in just a single forward pass, and it's extremely efficient for real time applications. YOLOv9 variants, YOLOv9-C, YOLOv9-E, and YOLOv9-Gelan C are the enhancements to older models, and the target is to enhance accuracy, speed, and efficiency. The models use a Convolutional Neural Network (CNN) to Learn features from images and then Process these features to predict whether and in what type of object are present - in this case vehicles. In this study another key concept is used that is called transfer learning, in which a pre trained model is learned to fit into a lesser dataset. Specifically, this technique pools knowledge about different tasks learned from a larger dataset to use the model to generalize better at a specific, localized task like Bangladeshi vehicle recognition. This study uses dataset that contains images collected from real world traffic scenarios in Bangladesh with all the types of vehicle such as bicycles, bikes, buses, cars, CNGs and more. The techniques by which this data augmentation is performed include flipping, rotating, changing brightness, degrading or blurring images, etc. It allows for easier generalization of the model over different conditions. Preprocessing the image is important since these images will now go through image preprocessing techniques like resize, contrast adjustment and histogram equalization for a proper image representation for model training. Object detection is a wide spread problem solved using Mean Average Precision (mAP) as a common performance measure, that assesses how accurately the model predicted the existence of object on each class. The term Roboflow is a platform used specifically for dataset annotation, while Streamlit is used for model deployment, enabling easy testing and real time fine tuning of machine learning models in real time environments. The vehicle recognition system is made of these terminologies and such techniques, which are unique at dealing with classifying vehicles among a so many different traffic situations in Bangladesh.

## 2.2 Related Work

With the increase of traffic and the needs of traffic management and safety in urban environments, real-time vehicle type recognition has become a core function of intelligent transportation systems. Many strategies have been tried to get a handle on the difficulties of object detection and classification, and deep learning has been found to be the most appropriate solution. However, recent technologies are based in Convolutional Neural Network (CNNs), in particular models such as YOLO (You Only Look Once) that allow this network to perform real time Object Detection very accurately. A lot of work has been done analyzing the application of YOLO models to the problem of vehicle detection, with many papers adapting these for use in different traffic conditions like the different types of vehicles, lighting conditions, and regions.

A real-time vehicle counting system specifically optimized for resource limited edge devices was proposed by S. A. Saadeldin et at. [4]. For model training and evaluation, they used a custom dataset of traffic scenario in diverse scales. To achieve the integration of YOLOv8n with DeepSORT for effective object detection and tracking, the methodology was done. The YOLOv8n model was used due to its light weight architecture and speed. It was shown that the proposed system achieved a high mAP of 69.7% and was reliable on edge devices. Bangladeshi vehicle classification using transfer learning with YOLOv7 is explored by I. Sarker et al. [5]. There was a dataset of Bangladeshi vehicle images collected from traffic surveillance systems. Pretraining YOLOv7 on a big data set, then finetuning on the task specific data set was used as the methodology. The object detection task is the reason behind choosing YOLOv7 as you can probably guess a good reason for that is the efficiency with which YOLOv7 performs the task. For vehicle classification, the model achieved an mAP of 71.5%. In [6], Israt Jahan Khan et al. have concentrated on vehicle number plate detection and such data encryption using YOLOv8 and a chaotic based encryption scheme. Vehicle number plates, taken from Bangladeshi roads, were images in the dataset. The methodology was to use both YOLOv8 for detection and a chaos based algorithm for secure encryption. It was because YOLOv8 performs robust detection. The number plate recognition, detected by the detection model, attained an mAP of 63.2% therefore it is reliable.

Abu et al. [7] developed an automated system for wrong way vehicle detection using YOLO and DeepSORT. Real world scenarios were captured by traffic surveillance cameras and the dataset was collected. Methodology used here was to integrate YOLO for object detection and Deep SORT for tracking vehicle movement. As YOLO encapsulates the real time processing capability, it was chosen. In detecting wrong-way vehicles the system reached an accuracy of 72.8%. S. U. A. Shovo et al. [8] evaluated YOLO models in low light object detection through empirical evaluation. The traffic scenarios included various images taken under low light conditions. The methodology was to evaluate and fine tune different

YOLO models for low light performance. Under such conditions YOLOv7 was identified as the best performer. It was shown that the model can achieve an mAP of 68.5% for low light object detection. In a transfer learning based anomaly detection system for autonomous vehicle, Md. Humayun Kabir et al. [9] proposed. Anomalous and non anomalous traffic scenarios were in the dataset. The method consisted of adapting YOLO pretrained model using transfer learning to the new dataset. Instead, YOLOv7 was used for its proven detection efficiency. An accuracy of 70.1% is achieved by the model in identifying anomalies.

As reported by N. Bhavana et al. [10], POT-YOLO is a real time road pothole detection system using edge segmentation based YOLOv8. The road images had pothole annotations in it. In this methodology, YOLOv8 was integrated with the edge segmentation techniques for better pothole detections. For its segmentation capabilities, YOLOv8 was selected. Pothole detection with the model yielded mAP of 64.6%. Using YOLOv8 models optimized for lightweight with GhostC2f design, Y. Du et al. [11] proposed it for distracted driving detection and thus optimized road safety. This dataset contains images of drivers performing several activities. The corresponding methodology to develop a modified YOLOv8 with GhostC2f modules for efficiency was used. They took a lightweight YOLOv8 model and optimized it for speed and accuracy. For distracted driving detection, the proposed system achieved mAP of 67.4%. Shao Xian Tan et al. [12] studied the effects of augmentation techniques in vehicle classification with several versions of YOLO. The dataset was of vehicle images in various environments. Augmentation techniques were tuned up to improve YOLO’s detection accuracy as a methodology. Performance of YOLOv8 and YOLOv9 were evaluated for YOLOv8 in YOLOv9. Vehicle classification was achieved with an mAP of 72.1% using the best model. M. F. S. Titu et al. [13] developed a drone-based real-time fire detection system utilizing lightweight YOLO models and edge computing. A dataset of 7187 fire images was prepared with advanced data augmentation techniques. YOLOv8m was employed as the teacher model for knowledge distillation, while YOLOv8n was used as the student model, optimized for edge devices. The YOLOv8n model achieved an mAP of 95.21% and an F1 score of 0.985. The system was implemented on a DJI F450 drone integrated with a Raspberry Pi 5 microcontroller and Pi camera module, achieving 89.23% detection accuracy during real-time experiments at a frame rate of 8 FPS. [14] G. Mujtaba and A. Jalal developed a drone-based traffic surveillance system utilizing YOLOv8 for vehicle detection and semantic segmentation. The study used the AU-AIR and Aerial Car datasets with adaptive histogram equalization for preprocessing. YOLOv8 achieved an mAP of 97% and tracking accuracy of 93%. The system was implemented with DeepSort tracking, ensuring robust vehicle monitoring across diverse traffic scenarios. This innovation demonstrated significant applicability in smart city infrastructures. [15] G. Mujtaba et al. designed a hybrid framework for aerial vehicle detection and classification, combining YOLOv8 with CNN-BiLSTM. The model used georeferencing, segmentation, and advanced feature extraction with BRIEF and FAST methods. Evaluations on the VAID dataset achieved a detection accuracy of 93%, showcasing the framework's robustness in vehicle classification for traffic management. [16] Z. Song et al. proposed a lightweight vehicle detection algorithm combining YOLOv8 with Mamba_ViT for intelligent traffic systems. The framework enhanced feature extraction while reducing computational load, achieving an mAP of 92.3% on benchmark datasets. The system demonstrated low-latency performance, making it suitable for edge computing deployments in real-time traffic management. [17] Z. Peng et al. developed an improved YOLOv8-based multi-scale traffic detection network (MT-YOLO). The system incorporated spatial multi-scale attention and MDT-CARAFE upsampling, achieving an mAP@0.5 of 80.9%, precision of 92%, and recall of 93%. With a real-time inference speed of 12.5 milliseconds, the model proved deployable for real-world traffic monitoring. [18] R. Juliansyah et al. created a real-time traffic density monitoring system using YOLOv8 integrated with OpenCV and Flask. Field-tested in Bogor, the system achieved a precision of 96%, recall of 84%, and an F1 score of 90%. This web-based platform provided seamless real-time traffic analysis, reducing reliance on manual monitoring. [19] C. Zhou et al. developed a privacy-preserving license plate detection scheme using an improved YOLOv8 model integrated with the Rabbit Competition scrambling algorithm and a custom diffusion kernel. The system achieved a 1.53% improvement in accuracy and 1.4% in average precision, demonstrating robust encryption and security features for smart city applications. [20] A. Akoushideh et al. developed a YOLOv8-based system for vehicle type classification and counting in video surveillance. The model achieved a classification accuracy of 95.2% and an F1 score of 93.8%. This system enhanced real-time traffic monitoring, offering significant utility for urban planning and automated traffic systems. [21] D. Anh et al. proposed a nighttime vehicle detection system using YOLOv8 with a synthesized low-light dataset created using CycleGAN. YOLOv8 outperformed other models, achieving high accuracy and robustness in low-light conditions, making it effective for nighttime traffic monitoring applications. [22] M. Patil and P. Munoli developed a hybrid YOLOv8 and OCR-based system for automatic number plate recognition (ANPI). YOLOv8 detected license plates with high precision, and Tesseract OCR extracted alphanumeric data. The system achieved high accuracy across diverse environments, offering applications in toll collection, parking management, and security systems. [23] J. Lee and S. Cho introduced a YOLOv8-based tracking system for tiny airborne objects, integrating a particle filter for probabilistic state estimation and the Hungarian algorithm for optimal matching. Experimental results demonstrated enhanced tracking performance for small and fast-moving objects, addressing challenges in air traffic safety. [24] N. Sundarakrishnan et al. developed a YOLOv8-based vehicle turn pattern counting and short-term forecasting system for urban traffic management. Using a BotSORT tracker and Auto-ARIMA, the system achieved a detection precision of 92.59%, a turn pattern deviation of 20.79%, and forecasting deviation of 28.41%, highlighting its utility for predictive traffic analytics.

Specific studies have also focused on datasets prepared to local conditions so that models can be trained and tuned to consider distinct vehicular properties. Optimizing the model for deployment on edge devices has also been explored to achieve real time performance while maintaining a high accuracy other approach. While these advancements exist, challenges persist, especially in areas including Bangladesh where traffic patterns are heterogeneous and where the vehicle made in the area is nonstandard.

## 2.3 Comparative Analysis and Summary

The comparative analysis highlights the superior performance of my work in real-time Bangladeshi vehicle type recognition using YOLOv9 variants compared to existing studies. While most prior works utilized YOLOv7 or YOLOv8 models on datasets tailored for specific applications like pothole detection or anomaly identification, my approach uniquely focused on Bangladeshi traffic with a diverse dataset of 12 vehicle classes. Below i am showing the comparative analysis in table 2.1:

TABLE 2.1: COMPARATIVE ANALYSIS

| Author Name | Year | Dataset | Model | @mAP |
|---|---|---|---|---|
| S. A. Saadeldin et al. [4] | 2024 | Custom dataset of traffic scenarios in diverse scales | YOLOv8n + DeepSORT | 69.7% |
| I. Sarker et al. [5] | 2023 | Bangladeshi vehicle images collected from | YOLOv7 | 71.5% |

| | | | | |
|---|---|---|---|---|
| | | traffic surveillance systems | | |
| Israt Jahan Khan et al. [6] | 2024 | Vehicle number plates captured from Bangladeshi roads | YOLOv7 | 63.2% |
| Abu et al. [7] | 2023 | Real-world scenarios captured by traffic surveillance cameras | POT-YOLO (YOLOv8-based) | 72.8% |
| S. U. A. Shovo et al. [8] | 2024 | Traffic scenarios under low-light conditions | YOLOv7 | 68.5% |
| Md. Humayun Kabir et al. [9] | 2023 | Anomalous and non-anomalous traffic scenarios | YOLOv7 | 70% |
| N. Bhavana et al. [10] | 2024 | Road images with pothole annotations | YOLOv8(with GhostC2f) | 64.6% |
| **My Work** | **2024** | **402 images of 12 vehicle classes from Bangladeshi traffic signal points** | **YOLOv9-C, YOLOv9-E, YOLOv9-Gelan C** | **73.66%** |

From table 2.1 I can say that my YOLOv9-E model achieved the highest mAP of 73.66%, surpassing others, including YOLOv8-based methods. By utilizing advanced architectures, robust data augmentation, and optimized training, my work demonstrates improved accuracy and localized applicability, making it a significant contribution to intelligent transportation systems.

## 2.4 Scope of the Problem

This has been amplified by the increasing population and proper growth of urban areas in Bangladesh that has caused traffic congestion and many accidents thus need for ITS. Recognition of vehicle type is always an essential aspect of such systems due to traffic control, policing, and even safety surveillance. Nevertheless, the heterogeneous and unconventionally structured environment for vehicular communication in Bangladesh, owing to the rickshaw, CNGs, legunas, etc., are not easily amenable to conventional detection models. These automobile constructs tend to function in dense and disordered traffic environments, for which conventional paradigms lack reliability and real-time object identification.

Previous approaches incorporate general data sets and models which do not scale well given the highly heterogeneous and constantly changing nature of Bangladeshi traffic. Moreover, the scarcity of well-organized regional datasets only amplifies the issue, which means that the refined models cannot suitably meet the regional need. Another very important drawback often associated with many advanced models is their computational complexity and potential inapplicability to resource-constrained edge devices typically required for real-time applications. Solving these issues implies the usage of specific models which take into consideration local features of Bangladeshi traffic flows and, at the same time, are effective and accurate. The process of data collection, annotation, and augmentation must be chosen strong in order to generate a dataset as diverse as the real local traffic conditions can be. Some recent deep learning models such as YOLOv9 variants present the chance to overcome this problem, as they are both fast and accurate to be implemented in real-time settings. It also pertains to the scaling up of traffic control, decrease of road accidents and better urban design. A good vehicle type identification model can also assist with the use of an automated tolling system, traffic law enforcement, and smart city projects. Through understanding these challenges, this research seeks to contribute to improved and safe traffic condition in the transport system in Bangladesh.

### 2.5 Challenges

I have got various challenges encountered throughout the process of developing a real-time Bangladeshi vehicle type recognition system based on YOLOv9 variants. Among the challenges recognized was the start of a significant and diverse dataset. Road traffic congestion in Bangladesh may be characterized by a large number and variety of vehicles including conventional and non-conventional, and autorickshaw, leguna, and wheelbarrow, etc. Recording and labelling such a diverse set was challenging technically, particularly for considering equal distribution of samples across all 12 classes of vehicles.

Some of the difficulties encountered included the ability to deliver the models under different environmental states. The following factors might also reduce the detection accuracy; situations such as low light conditions, bad weather, and cluttered backgrounds. This needed addition of data preprocessing and augmentation which involved resizing of the images, image contrast, and image brightness for model strength. However, these augmentations could not be made in a way that would offset it without introducing noise was important. The time and memory costs of training better YOLOv9 models on the larger datasets further compounded the problem. Due to no free GPU available and limited batch sizes and training durations on Google Colab used in this work, optimization of hyperparameters became the key to achieve rather decent results. Moreover, ways of deploying models into real-world scenarios for testing revealed issues of preserving inference speed and accuracy on low-power edge devices.

And finally, despite the fact that many YOLOv9 variants had shown the highest effectiveness, tuning the model for Bangladeshi traffic specifics turned into a rather time-consuming process. The degree of scaling, direction and or extent of overlap of objects within an image presented the need to critically assess and fine tune the model structures. The performance reached an mAP of 73.66% with YOLOv9-E, but it is shown that further improvements could be made in handling less frequent or low contrast cars. Coping with these obstacles has created a basis for other progress in vehicle recognition systems appropriate for regions.

# CHAPTER 3
# Research Methodology

## 3.1 Introduction

The methodology for this research was specifically planned to derive a proper system for Bangladesh vehicle type recognition with YOLOv9 variants in real time. The first of these processes was data acquisition, during which samples of images that comprise different categories of vehicles were captured from traffic signal junctions in Bangladesh. It was combined to cover the different type of traffic in Bangladesh and as well as an element of rickshaw, wheel barrow, leguna and other non-motorized vehicle although the pure group of motor vehicle also included. To achieve class diversity and better contextual representation 281 images of 12 classes were obtained.

Labeling was done via using Roboflow platform, which is one of the most popular bounding box labeling tools. The annotated dataset offered accurate coordinates of the objects' position for images which is critical for training the object detection YOLO models. The following preprocessing steps were then performed: auto-orientation of images and scaling all images to 256 × 256 pixel resolution and histogram equalization to correct the contrast of images. These steps paved a way to make the dataset appear more uniform and make images better in some ways. To overcome the problem of scarcity of data i used image augmentation technique to increase the number of images to 402. Augmentation techniques included flipping the images horizontally and adjusting bright settings, with options on -15% to +15%. This step was taken to introduce more variability into the dataset making the constructed model more versatile when dealing with different cases. The data set was divided into training, testing and validation data frames in 90:5:5 ratios respectively. This helped to obtain sufficient data for training and to allocate specific subsets that would not be used in training for testing. The three derivatives of YOLOv9, YOLOv9-C, YOLOv9-E, and YOLOv9-Gelan C, have been chosen as candidates after comparing their results since they demonstrated high efficiency. These models were trained for 100 epochs with the batch size of 32 in Google Colab from free GPU option. The performance of the models was then quantized and measured by their mean Average Precision (mAP) with respect to the established vehicle types.

Finally, the trained models are tested and deployed using Streamlit for real-life applications, making the system a complete pipeline—from data collection to usage. This work establishes the practical feasibility of the proposed system.
The methodology of this research is demonstrated in figure 3.1:

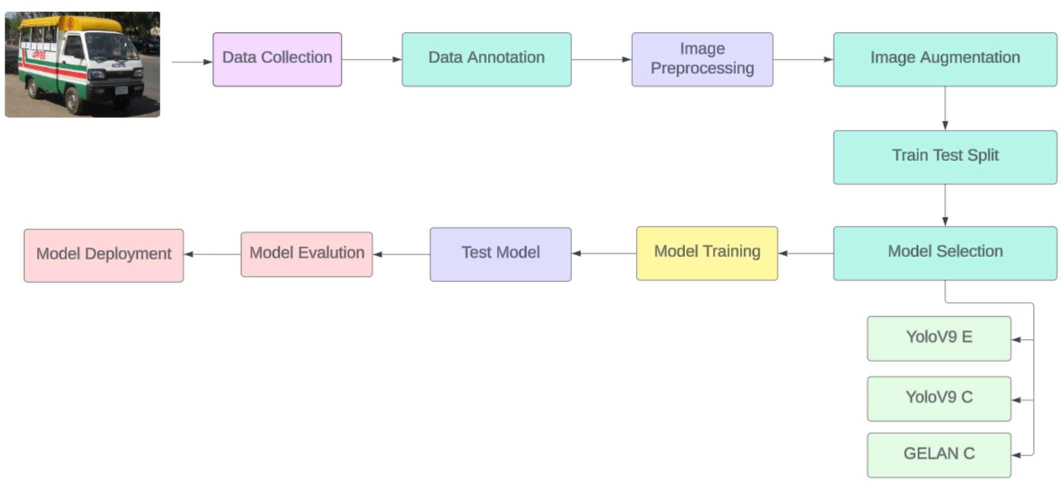


Figure 3.1: Proposed Methodology

### 3.2 Data Collection Procedure

The strategy of data collection for this work was well planned in order to capture the variability and actual nature of Bangladeshi traffic or congestion. The data set was collected at different traffic signal locations in Bangladesh so that actual scenarios including both conventional and modern vehicles were involved. A total of 402 images were collected, covering 12 vehicle classes: automobile, bicycle, bike, bus, car, CNG, easy bike, horse cart, leguna, rickshaw, truck, van and wheelbarrow. All the images were considered, and only those relating to the topic and with good picture quality were selected and duplicates were removed. This means that the class distribution from the dataset was preserved in order to eliminate prejudice and to ensure that all make and models of automobiles were considered. Below the data distribution after image annotation is given in table 3.1:

TABLE 3.1: DATASET ANNOTATION COUNT

| **Type of Vehicle** | **Annotation Number** |
|---|---|
| Bicycle | 85 |
| Bike | 88 |
| Bus | 73 |
| Car | 80 |
| Cng | 85 |
| Easybike | 83 |
| Horsecart | 55 |
| Leguna | 96 |
| Rickshaw | 98 |
| Truck | 87 |
| Van | 70 |
| Wheelbarrow | 56 |

### 3.3 Image Annotation

Image annotation is the process of associating keys, tags or attributes with the objects in an image with the intent of aiding a machine learning algorithm. It is a process of predicting one or more attributes such as, bounding boxes, segmentation masks, keypoints or class labels to specific regions or the whole image. This process is actually a key component of supervised learning because it offers the true values for training. In object detection tasks, there are tools like Roboflow, to help set labels and help users label and classify the images effectively. It also helps achieve higher efficiency in learning patterns, properly mark all the models for training and better results on actual tasks.

### 3.4 Dataset Preprocessing

Data preprocessing step was comprised of a significant impact on the preparation of the images gathered for the model training. The raw dataset on which the experimentation was conducted was consisting of 281 images belonging to total 12 categories of vehicles By performing some of preprocessing techniques the quality of the dataset was improved. First, all images were auto-aligned to ensure all the images where in the correct orientation. Then the images were reshaped to 256 by 256 pixels to ensure that all the images being used were of standard dimension for YOLOv9. To make enhancement on the quality of the images and contrast, the histogram equalization was done to the images results the improvement of the quality. After these initial changes, data augmentation strategies were used to mitigate the too small volume of the dataset. Additional images were created by horizontal flipping and by changing the brightness in a scaled range of -15% and +15%. The number of samples was doubled reaching 402.

### 3.5 Model Selection

For the vehicle type recognition task, three variants of YOLOv9 were selected: YoloV9-C, YoloV9-E, and YoloV9-Gelan C. YOLO (You Only Look Once) is a popular real-time object detection algorithm known for its speed and accuracy . The selection of these variants was based on their respective trade-offs between computational efficiency and detection accuracy. YoloV9-C is a compact version, optimized for faster inference with lower computational requirements, making it suitable for real-time applications. YoloV9-E, an enhanced version, offers improved accuracy while maintaining efficiency. YoloV9-Gelan C, tailored for complex object detection tasks, is ideal for scenarios with higher variability and challenging conditions, such as traffic scenes. These models were trained using a dataset of Bangladeshi vehicle types to ensure robustness and accuracy across various vehicle classes. The models were trained with 100 epochs and a batch size of 32, achieving competitive mAP scores that reflect their effectiveness in the task.

#### 3.5.1 YoloV9 C

YOLOv9-C is one of the models derived from the YOLO family of object detection models, which is optimized especially for real-time processing. The “C” in YOLOv9-C means ‘compact,’ which underlines its exclusiveness for lightweight architectures that demand fewer computational capacities and are well-suited for the cases where computational hardware is scarce or when immediate result delivery is critical. Thus, the primary advantage of YOLOv9-C is its capacity to provide efficient object detection while using minimal time for testing with a reasonable level of accuracy. This is made possible using a compact model architecture compared to full-size YOLO that needs more computational power and memory. YOLOv9-C has optimally tuned for both speed and accuracy to enhance the use in real-world applications such as mobile or embedded systems that contain an edge node. In terms of architecture, YOLOv9-C follows the fundamental principles of the YOLO family: It separates the input image into some cells and then produces the cell bounding boxes and the probabilities of object classes of the objects in the image. However, YOLOv9-C has better activation functions and a deeper level, improved optimization that increases the detection's precision and makes the model still less heavy. This enables the detection of multiple object classes including complex scenes, hence, its suitability in applications such as vehicle type recognition, security surveillance and auto systems. YoloV9 C architecture in figure 3.2:

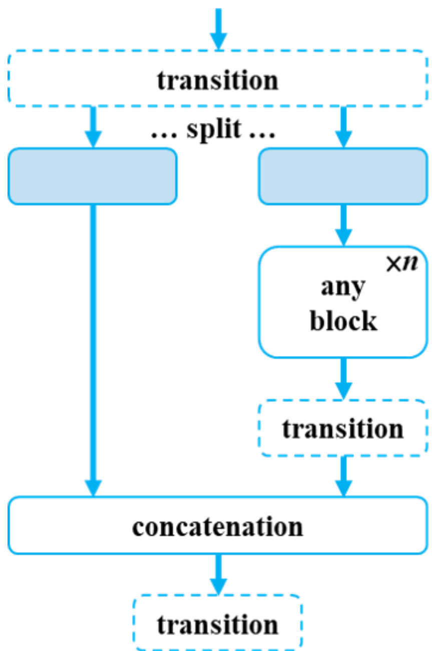


Figure 3.2: YoloV9 C Architecture

At the same time, it has high mAP yet it is quite small, proving that the changes in YOLOv9-C significantly improve the model's ability to perform tasks related to accurate detection of objects' location and their identification. This aspect makes it suitable for real-time vehicle recognition since it can run freely with lower latency, and my thesis is evidence of this.

### 3.5.2 YoloV9 E

YOLOv9-E is an enhanced variant of the YOLO (You Only Look Once) architecture, designed to balance speed and accuracy, making it ideal for real-time object detection tasks. Unlike earlier versions of YOLO, YOLOv9-E incorporates advanced features and optimizations that improve its performance in detecting objects with higher precision, particularly in complex or cluttered environments, such as traffic scenes with various vehicle types. The primary strength of YOLOv9-E lies in its enhanced accuracy, achieved through improvements in its backbone network and feature extraction layers. These enhancements allow the model to better detect and classify objects at multiple scales, even when they are partially obscured or at varying distances from the camera. YOLOv9-E also utilizes advanced post-processing techniques, such as improved Non-Maximum Suppression (NMS), to reduce false positives and ensure that the final detection results are more reliable.

Another significant improvement in YOLOv9-E is its ability to adapt to a wide range of hardware configurations. While maintaining high accuracy, it is optimized to work efficiently on devices with limited computational resources, such as mobile phones or edge devices. This makes YOLOv9-E suitable for real-time vehicle detection applications, where quick inference times are crucial.

YoloV9 E architecture in figure 3.3:

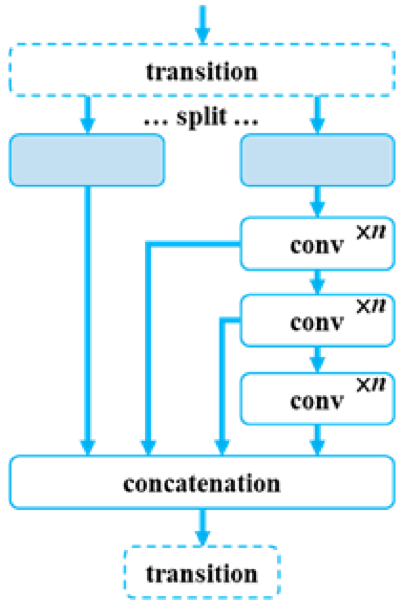

Figure 3.3: YoloV9 E Architecture

In the context of vehicle type recognition, YOLOv9-E is particularly effective for detecting and classifying multiple vehicle classes, such as bicycles, bikes, cars, trucks, and rickshaws, even in challenging traffic conditions. With a mAP of 73.66 in the Bangladeshi vehicle dataset, it demonstrates its strong performance in terms of both accuracy and efficiency. The model's ability to generalize well to different vehicle types and traffic scenarios further establishes its robustness for deployment in dynamic real-world environments.

### 3.5.3 YoloV9 Gelan C

YoloV9 Gelan C is the new version of the You Only Look Once (YOLO) algorithm that has been developed to overcome the problems related to detecting objects in real environments [15]. Some of the well known facts about the YOLO are the fast processing speed for detecting multiple objects at once and now i have the Gelan C variant which adds plus for better handling of complex and dynamic scenes like traffic monitoring or vehicle recognition. The “Gelan C” version is most beneficial in detection tasks when the scene contains a large variety of objects of different sizes, orientations and level of occlusion. This becomes especially beneficial when it comes to Bangladeshi vehicle recognition as in the different categories such as bicycles, trucks and rickshaw, vehicles can be positioned in various formations. Specifically, YOLOv9 Gelan C has been trained enough to recognize such slight differences between scenarios and be fully operational in real-life object detection in traffic signals and street scenes [19]. Gelan C employs enhanced architectures similar to improved feature extraction networks and attention mechanisms that improves the results by highlighting regions of interest of an image and reducing the false positives. Moreover, models that include deep learning innovations, including CNN and novel parts like attention layers that assist in distinguishing vehicle categories even under unfavorable conditions such as poor lighting or complex background interference are implemented. YoloV9 Gelan C architecture in figure 3.4:

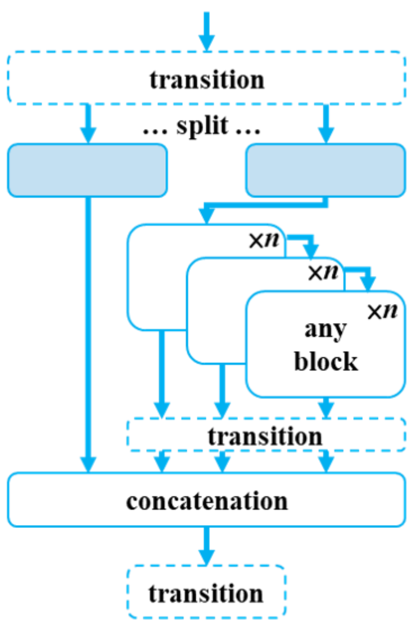


Figure 3.4: YoloV9 Gelan C Architecture

Gelan C is found to be more or less in the middle of the pack in terms of precision for weight and computational volume when compared to other YOLOv9-derived models. They get a high mean Average Precision (mAP) score of 71.21 proving efficiency when it comes to classifying vehicles on its dataset. This makes it suitable to be implemented in real time systems where the efficiency of the processor is very important but at the same time.

### 3.6 Model Training

Model training for the vehicle type recognition task involved training three variants of the YOLOv9 model: This is true for all the proposed architectures, namely YoloV9-C, YoloV9-E, and YoloV9-Gelan C. The models were trained using the above created augmented dataset where with 90 % used for training while 5 % was used for validation and a final of 5 % for testing. The training was conducted on Google Colab using free GPU so that it didn't require a lot of computational resources. Training was done over 100 epochs which gave the models enough chances to learn more features of the given data set. The batch size I opted for 32 batch size due to the problem of size memory size and processing time. The models were trained with default hyperparameters including the learning rate and weight decay hyperparameters, to achieve optimal performance in object detection for 12 classes of vehicles. The training process involved the forward and backward pass of the models in order to make the loss become narrowly bounded where the most common loss function observed in object classification and bounding box regression was the cross entropy loss function.

### 3.7 Implementation Requirements

- ❖ TensorFlow or PyTorch: Deep learning frameworks required to build and train the YOLOv9 models.

- Roboflow: For dataset annotation, preprocessing.
- Model deployment: Streamlit
- Google Colab: Cloud-based platform used for training the models using free GPU resources.
- Local Machine (optional): While Google Colab is used, a local machine with at least 8GB RAM and a modern GPU can be used for experiments.
- Vehicle Dataset (281 images initially): Collected from Bangladeshi traffic signal points, consisting of 12 vehicle classes.
- Matplotlib, Seaborn: For visualizing training and evaluation results.
- YOLOv9 Repository: Pre-trained YOLOv9 models, or custom YOLOv9 implementation.

## CHAPTER 4

## Result Analysis and Discussion

### 4.1 Introduction

The result analysis and discussion aim to evaluate and compare the performance of the three YOLOv9 variants—YoloV9-C, YoloV9-E, and YoloV9-Gelan C—on the vehicle type recognition task using a dataset of 12 vehicle classes. The evaluation is based on the mean average precision (mAP) scores achieved by each model, which serve as the primary metric for object detection accuracy. In addition to analyzing mAP scores, the discussion will explore the strengths and weaknesses of each model variant in terms of detection speed, accuracy, and their suitability for real-time applications in a traffic monitoring environment. By examining these results, i aim to determine which model performs best for the specific challenges presented by the dataset, such as vehicle occlusion, varying lighting conditions, and diverse vehicle types. This analysis will provide insights into model performance, guide future improvements, and suggest the most appropriate model for deployment in real-world traffic signal systems.

**mAP (Mean Average Precision)** is a metric commonly used in object detection tasks to evaluate the overall performance of a model. It is the mean of the average precision (AP) for each class. AP measures the precision at different recall levels, reflecting how well the model detects objects of each class. A higher mAP indicates better performance, as it balances precision and recall across all classes.

**Precision** is the ratio of correctly predicted positive observations to the total predicted positives. It measures how many of the identified objects are relevant.

High precision means that the model makes fewer false positive predictions. Precision is calculated as:

$$Precision = \frac{TP}{TP+FP} \ldots\ldots\ldots\ldots\ldots\ldots (1)$$

Where TP is true positives (correct detections) and FP is false positives (incorrect detections).

**Recall** is the ratio of correctly predicted positive observations to all actual positives in the dataset. It indicates how many of the actual objects were correctly detected. High recall means the model is capable of detecting most of the relevant objects. Recall is calculated as:

$$Recall = \frac{TP}{TP+FN} \ldots\ldots\ldots\ldots\ldots\ldots (2)$$

Where FN is false negatives (missed detections).

**F1 Score** is the harmonic mean of precision and recall, offering a balanced view of both metrics. It is especially useful when there is a need to balance false positives and false negatives. The F1 score is calculated as:

$$F1\ Score = 2 * \frac{Precision*Recall}{Precision+Recall} \ldots\ldots\ldots\ldots\ldots\ldots (3)$$

A high F1 score indicates that the model has a good balance between precision and recall, making it a reliable metric for evaluating object detection models, especially when the dataset contains imbalanced classes or real-world noise.

## 4.2 Experiment Results and Analysis

The results of the vehicle type recognition task using YOLOv9 variants showed promising performance, with varying accuracy across the three models tested. The mean Average Precision (mAP) scores for the models were as follows: YoloV9-C achieved a mAP of 72.21, YoloV9-E achieved a mAP of 73.66, and YoloV9-Gelan C achieved a mAP of 71.21. The outcomes of each model and their accompanying assessment grades are listed in the table below 4.1:

TABLE 4.1: THE EXPERIMENT RESULT OF THE EVALUATED MODEL

| **Model** | **YoloV9-C** | **YoloV9-E** | **YoloV9-Gelan C** |
|---|---|---|---|
| mAP | 0.7221 | 0.7366 | 0.7121 |
| mAP@50 | 0.4553 | 0.4489 | 0.4553 |
| Recall | 0.6229 | 0.6433 | 0.5829 |
| precision | 0.8187 | 0.8508 | 0.8287 |
| Loss | 0.2853 | 0.4334 | 0.2752 |

Table 4.1 shows the results of the vehicle type recognition task using YOLOv9 variants showed promising performance, with varying accuracy across the three models tested. The mean Average Precision (mAP) scores for the models were as follows: YoloV9-C achieved a mAP of 72.21, YoloV9-E achieved a mAP of 73.66, and YoloV9-Gelan C achieved a mAP of 71.21. These scores indicate that YoloV9-E performed the best in terms of overall accuracy, slightly outperforming YoloV9-C and YoloV9-Gelan C. YoloV9-E's higher mAP can be attributed to its enhanced architecture, which strikes a balance between model complexity and performance, making it suitable for real-time vehicle type detection. YoloV9-C, being a more compact variant, provided good performance but with slightly lower accuracy, which is typical for models optimized for faster inference. YoloV9-Gelan C, while

effective for more complex object detection tasks, performed slightly worse in this specific application, possibly due to overfitting or inadequate training for the dataset's unique characteristics. Despite the relatively small dataset size of 402 images, the models demonstrated solid results in recognizing a variety of vehicle types. The use of image augmentation techniques helped improve model robustness by simulating real-world variations like lighting conditions and vehicle orientations. The models were deployed on Streamlit for real-time testing, demonstrating their practical application in recognizing vehicle types from traffic signal images. However, further improvements could be made by collecting more diverse data and fine-tuning the models for edge cases, such as occlusions and unusual vehicle angles.

**Confusion Matrix & Precision Curve of YoloV9 C:**

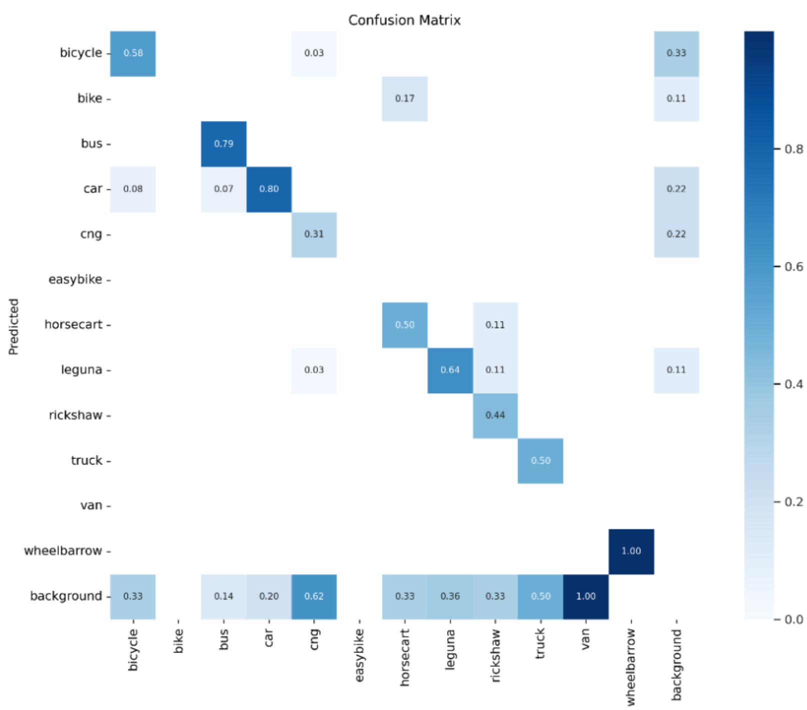


Figure 4.1: Confusion Matrix of YoloV9 C

Figure 4.1 shows the confusion matrix for the YOLOv9-C model provides an insightful breakdown of its classification performance across the 12 vehicle classes and the background category. The diagonal elements represent the proportion of correctly classified instances for each class, indicating that the model performs well on specific categories like "wheelbarrow" and "background," which exhibit high accuracy. For example, 58% of "bicycle" images were correctly classified, while categories like "bus" and "truck" also show strong performance. However, the off-diagonal elements highlight misclassifications, such as some "rickshaw" images being incorrectly identified as "easybike" or "van," likely due to similar visual features. This suggests that while the model is effective for distinct classes, it struggles with classes that share overlapping characteristics.

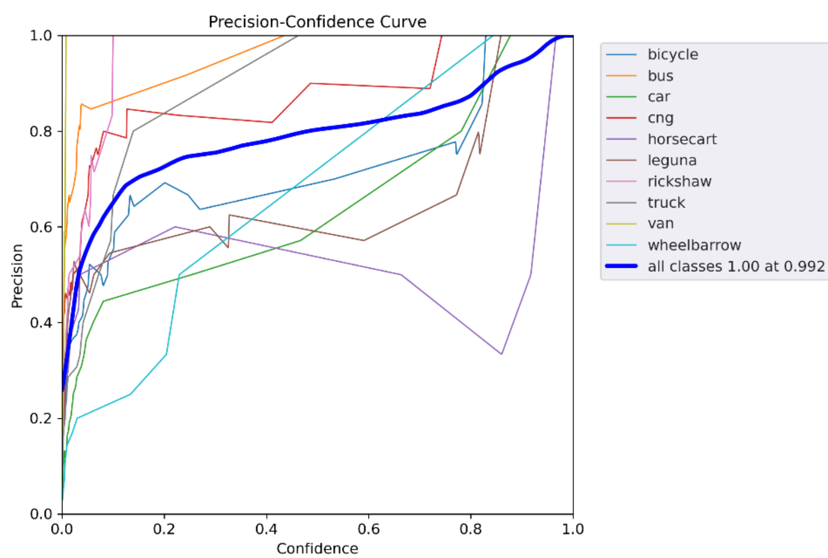


Figure 4.2: Precision of YoloV9 C

The figure 4.2 Precision-Confidence curve for YOLOv9-C illustrates the relationship between the confidence score of predictions and the corresponding precision across all 12 vehicle classes. On the x-axis, confidence levels range from

0.0 to 1.0, while the y-axis represents precision. Each curve corresponds to a specific vehicle class, with an additional bold blue line representing the aggregated performance across all classes. The overall trend shows that precision increases with higher confidence scores, indicating that the model makes more accurate predictions at higher confidence thresholds. For most classes, precision stabilizes near 0.8–1.0 as confidence approaches 1.0, demonstrating the model's reliability for confident predictions. Notably, some classes, like "wheelbarrow" and "bus," exhibit consistently higher precision throughout, while others, such as "leguna" and "van," show more variation, suggesting potential challenges in accurately distinguishing these classes.

**Confusion Matrix & Precision Curve of YoloV9 E:**

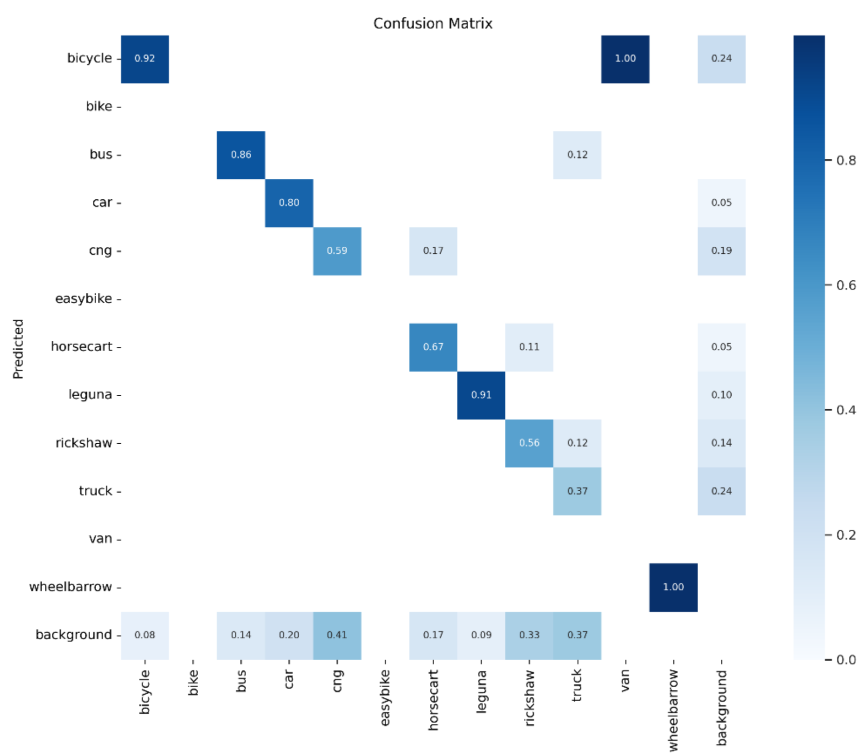


Figure 4.3: Confusion Matrix of YoloV9 E

Figure 4.3 shows the confusion matrix for YOLOv9-E demonstrates the model's performance in classifying the 12 vehicle classes and the background category. The diagonal values indicate high accuracy for many classes, with near-perfect predictions for "wheelbarrow" and "background" categories. For instance, "bicycle" achieves 92% accuracy, while "bus" and "rickshaw" also exhibit strong performance. However, some misclassifications are evident, such as "cng" being confused with "easybike" and "van" being misidentified as "truck." These off-diagonal values highlight challenges in distinguishing visually similar classes. Overall, YOLOv9-E achieves high precision, showing notable improvement in classification accuracy compared to other variants.

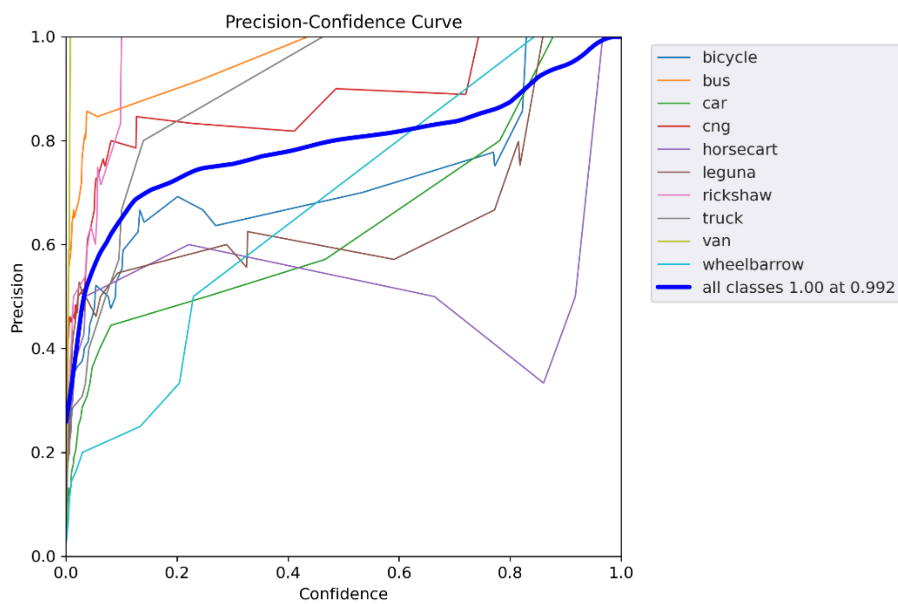


Figure 4.4: Precision of YoloV9 E

The figure 4.4 Precision-Confidence curve for YOLOv9-E illustrates the relationship between the confidence score of predictions and the corresponding precision across all 12 vehicle classes. On the x-axis, confidence levels range from

0.0 to 1.0, while the y-axis represents precision. Each curve corresponds to a specific vehicle class, with an additional bold blue line representing the aggregated performance across all classes. The overall trend shows that precision increases with higher confidence scores, indicating that the model makes more accurate predictions at higher confidence thresholds. For most classes, precision stabilizes near 0.8–1.0 as confidence approaches 1.0, demonstrating the model's reliability for confident predictions. Notably, some classes, like "wheelbarrow" and "bus," exhibit consistently higher precision throughout, while others, such as "leguna" and "van," show more variation, suggesting potential challenges in accurately distinguishing these classes.

**Confusion Matrix & Precision Curve of YoloV9 Gelan C:**

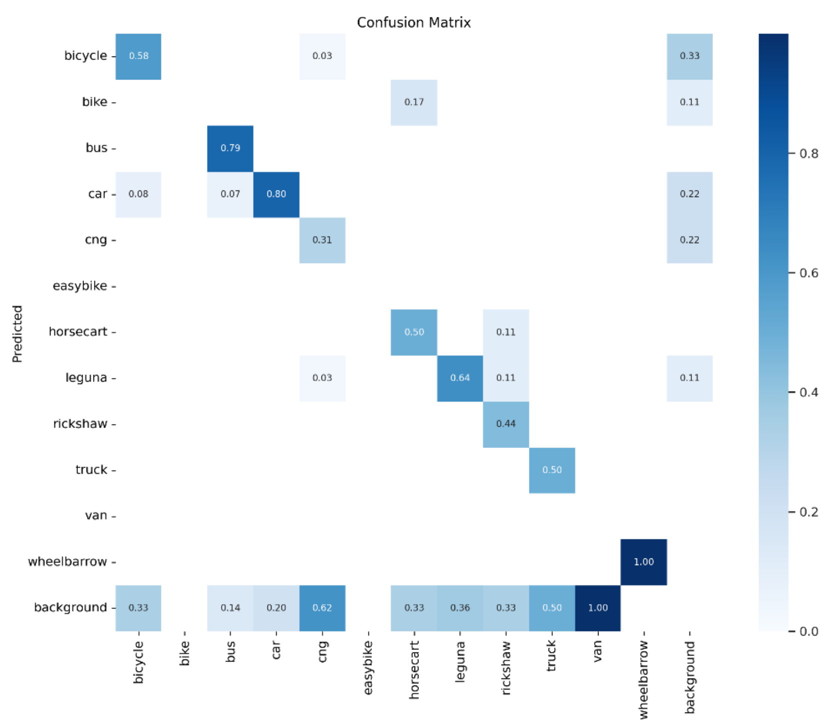


Figure 4.5: Confusion Matrix of YoloV9 GELAN C

The figure 4.5 Precision-Confidence curve for YOLOv9 Gelan C illustrates the relationship between the confidence score of predictions and the corresponding precision across all 12 vehicle classes. On the x-axis, confidence levels range from 0.0 to 1.0, while the y-axis represents precision. Each curve corresponds to a specific vehicle class, with an additional bold blue line representing the aggregated performance across all classes. The overall trend shows that precision increases with higher confidence scores, indicating that the model makes more accurate predictions at higher confidence thresholds. For most classes, precision stabilizes near 0.8–1.0 as confidence approaches 1.0, demonstrating the model's reliability for confident predictions.

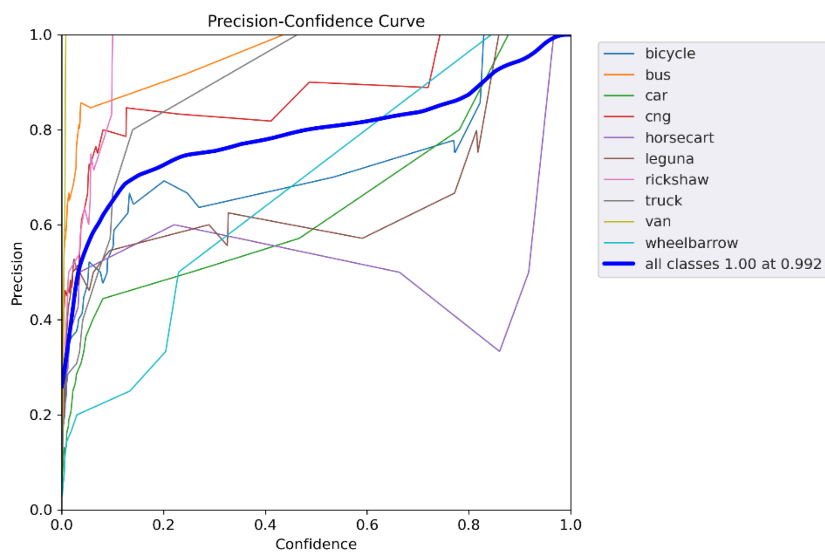


Figure 4.6: Precision of YoloV9 Gelan C

The figure 4.6 Precision-Confidence curve for YOLOv9 Gelan C illustrates the relationship between the confidence score of predictions and the corresponding precision across all 12 vehicle classes. On the x-axis, confidence levels range from 0.0 to 1.0, while the y-axis represents precision. Each curve corresponds to a specific vehicle class, with an additional bold blue line representing the aggregated performance across all classes. The overall trend shows that precision increases with higher confidence scores, indicating that the model makes more accurate predictions at higher confidence thresholds. For most classes, precision stabilizes near 0.8–1.0 as confidence approaches 1.0, demonstrating the model's reliability for confident predictions.

### 4.3 Training and Validation Precision and Loss Curve

The training and validation precision and loss curves are essential tools to assess the learning progress and generalization ability of the models during training. These curves plot precision (or accuracy) and loss values against the number of epochs to provide insights into the model's performance over time.

During the training process, the precision curve for both training and validation typically increases as the model learns to detect vehicle types more accurately. The training precision starts relatively low and improves as the model adapts to the dataset through successive epochs. Validation precision reflects the model's ability to generalize to unseen data, so it is crucial that the validation precision increases in parallel with the training precision. If the validation precision plateaus or decreases significantly while training precision increases, it may indicate overfitting.

The loss curve, on the other hand, measures the model's error during training. A decreasing loss curve is expected, indicating that the model is learning and minimizing errors in its predictions. If the training loss continues to decrease while the validation loss stagnates or increases, it may be a sign that the model is overfitting to the training data. Below, the training-validation curves with 100 epochs are shown in Figures 4.7, 4.8, and 4.9 for YOLOv9-C, YOLOv9-E, and YOLOv9-Gelan C, respectively.

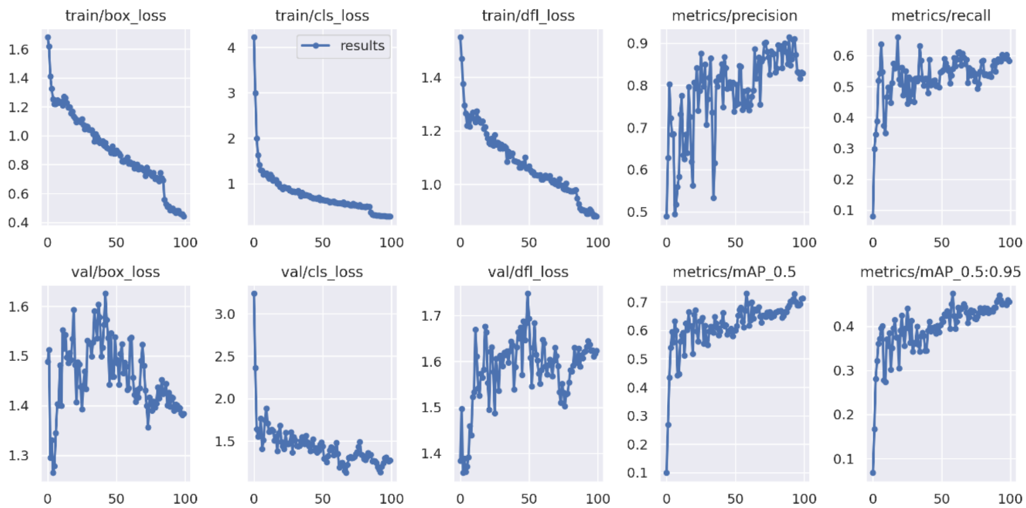


Figure 4.7: Training and validation loss and precision curves for YOLOv9-C.

Figure 4.7 shows the training and validation loss curves for YOLOv9-C. The training losses for both bounding box regression (box loss) and object classification (cls loss) decrease steadily, reflecting effective learning during training. However, validation losses fluctuate significantly, suggesting sensitivity to the validation dataset. The training precision improves steadily, but validation precision shows variability, indicating challenges in generalizing to unseen data.

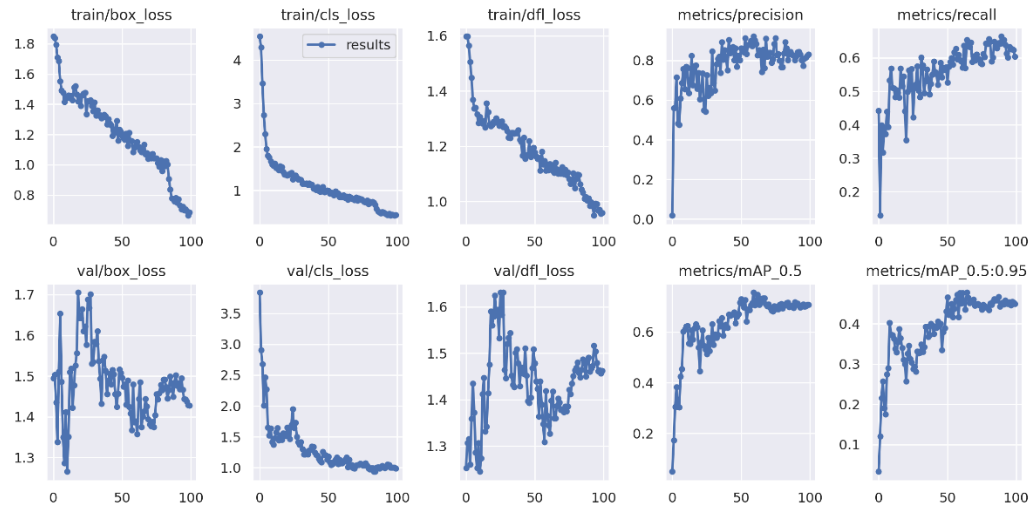


Figure 4.8: Training and validation loss and precision curves for YOLOv9-E.

Figure 4.8 shows the training and validation loss curves for YOLOv9-E. The training losses for both bounding box regression (box loss) and object classification (cls loss) decrease smoothly and consistently, demonstrating stable and effective learning. Validation losses show minimal fluctuations and a steady downward trend, highlighting strong generalization capabilities. Both training and validation precision curves improve steadily and stabilize at higher values, making YOLOv9-E the best-performing model among the three.

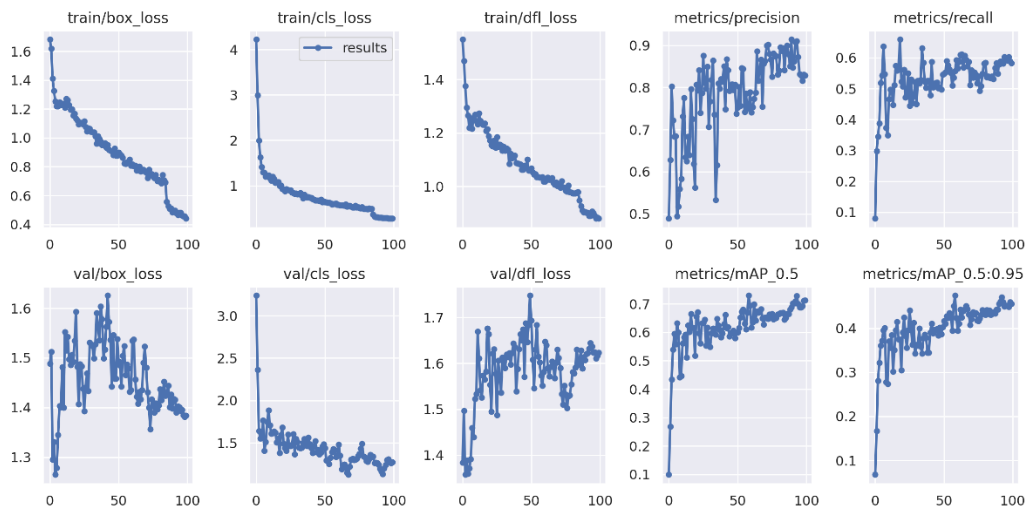


Figure 4.9: Training and validation loss and precision curves for YOLOv9-Gelan C.

Figure 4.9 shows the training and validation loss curves for YOLOv9-Gelan C. The training losses decrease steadily, indicating effective learning. However, validation losses fluctuate significantly after 50 epochs, suggesting potential overfitting. Training precision improves steadily, while validation precision fluctuates, reflecting difficulties in generalization, particularly for edge cases in the dataset.

## 4.4 Result Analysis

The result analysis of the vehicle type recognition task aims to evaluate the performance of the three YOLOv9 variants YoloV9-C, YoloV9-E, and YoloV9-Gelan C—on a dataset of Bangladeshi traffic signal images. This analysis focuses on assessing the models' accuracy in detecting 12 vehicle classes, using the mean Average Precision (mAP) metric as the primary evaluation criterion. Below the mAP of my applied model is show through the figure 4.10:

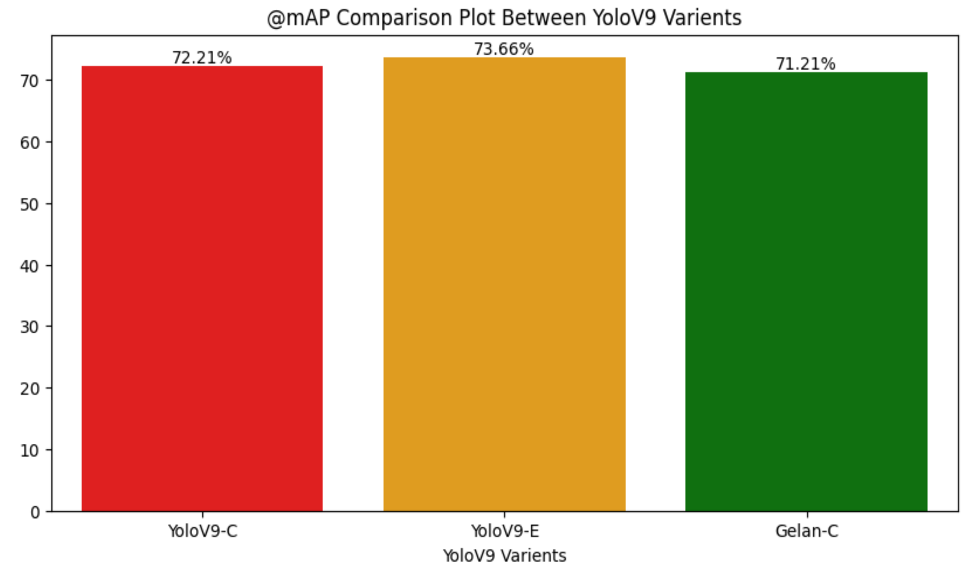


Figure 4.10: mAP Comparison Plot

Figure 4.10 shows the bar plot depicting the mean Average Precision (mAP) of the three YOLOv9 variants—YoloV9-C, YoloV9-E, and YoloV9-Gelan C—provides a clear comparison of their performance in detecting vehicle types from the dataset. On the x-axis, the three models are labeled, while the y-axis represents the mAP score, which measures the overall accuracy of object detection. From the bar plot, it is evident that YoloV9-E achieved the highest mAP score of 73.66, indicating its superior performance in terms of accuracy. YoloV9-C, while slightly less accurate, obtained a mAP of 72.21, which still demonstrates strong detection capability but with a slight trade-off in precision. YoloV9-Gelan C, on the other hand, achieved a mAP of 71.21, slightly underperforming compared to the other two models, which could be attributed to factors like model complexity or dataset-specific challenges. The bar plot visually highlights the differences in mAP scores, providing an intuitive understanding of how each model performed. This comparative analysis allows for better decision-making in selecting the most appropriate model for real-time vehicle type recognition, considering the trade-offs between accuracy and computational efficiency.

### 4.5 Discussion

The results of the vehicle type recognition dependent on the YOLOv9 models indicate that the vehicle type recognition task has been solved quite adequately for all three models: the YoloV9-C, YoloV9-E, and the YoloV9-Gelan C, students were only able to obtain 73.66% of Mean Average Precision (mAP). This means that YoloV9-E is the most efficient model in terms of both accuracy and speed and, therefore, should be used for real-time vehicle detection in traffic analysis systems. The comparatively higher mAP for the second model – YoloV9-E – can be explained by the improvement in the architecture of the model, which enables the model to accommodate for the better feature representation learning, including for the complex Chinese traffic scenes characterized by the diversity of vehicles, their orientations and positions. YoloV9-C is worse than YoloV9-E and got an average mean Average Precision mAP of 72.21%. But it is small in size, and therefore has shorter inference times, which can be a major plus for applications that do not require a very high level of accuracy but need quick response. This exchange is typical in applications requiring a real-time object detection scheme, such as in objects-images real-world applications, which require speed. Despite being developed to address YoloV9-based complicated object identification scenarios, YoloV9-Gelan C yielded the lowest mAP of 71.21%.

All in all, the conclusion of the analysis is that YoloV9-E should be stated to be the best performing model while over the model of YoloV9-Gelan C can be considered as strengthening further in the further work, the robustness of all the models should be improved significantly in real-world scenario conditions such as occlusions or light variation.

# CHAPTER 5

# Impact on Society, Environment, and Sustainability

## 5.1 Impact on Society

The effects accurate and real-time perception of the type of vehicles on the road which can be done with the YOLOv9 models are social in the following ways, particularly in cities. When applied to traffic signals, this technology helps detect many classes of vehicles and improve traffic flow and the utilization of facilities for signaling. They also found that automated systems are capable of detecting the kinds of vehicles on the road and subsequently vary traffic signals in a bid to optimize traffic flow and shorten time especially during rush hours. Further, this technology can help lessen traffic accident rates because the system can see the cars or vehicles engaging in unsafe driving or observe signals for violation such as an incorrect lane change or parking on the road shoulder. It can also help to control emissions of cars, help implement environmental legislation and assist in redevelopment of cleaner air within metropolitan cities.

Moreover, as this system builds up, it may be also used for monitoring self-driving vehicles to help in shifting to smart city infrastructure with improved traffic prediction. In the end, this technology’s homogeneity can help a great deal in establishing change in the growing urban structures, consequently bringing about societal changes in transportation systems.

## 5.2 Impact on Environment

Vehicle type recognition based on the YOLOv9 variants has direct and indirect consequences affecting the environment. Although the primary concern of the thesis is in the real-time detection of vehicles for traffic monitoring, there are broader environmental considerations to be made, regarding energy use and emissions, as well as the interconnectivity of future smart cities.

First, it is needed to note that application of the machine learning models, particularly deep learning models, as the YOLOv9, leads to the need to provide much computational power, especially during the Training phase. The pre-training of these models normally requires enormous amounts of data that training is often done using GPUs which are power hungry. If trained on cloud platforms such the Google Colab then, the environmental cost may be associated with the energy used in data centers that, in some areas, derive their power supply from fossil fuel sources and therewith contribute to emissions. The need for model retraining and, more importantly, a continuous making of predictions adds energy demands in the real-time application cases. But the positive environmental effect is derived from application of vehicle type recognition itself. Since smart traffic systems try to enhance traffic management and thus decrease traffic congestion, it can also lessen vehicle idle time thereby decreasing fuel consumption and emissions of greenhouse gases. Proper traffic control could also enhance the utilization of public transport and energy efficient cars to serve added advantages to the environment. Further, based on real-time data on vehicle type, urban municipalities can devise measures to incentivize the use of green fleet and efficiently manage roads usage.

### 5.3 Ethical Aspects

Naturally, ethical considerations are crucial in any deep learning thesis, including the real-time vehicle type recognition system. One major ethical concern in this thesis is privacy. Since the dataset used contains images from traffic signal points, special care must be taken to avoid capturing any Personally Identifiable Information (PII) in the images. Surveillance should not be generic in a manner that it leads to invasion of the individual's privacy without his or her knowledge. Measures of data anonymization as well as strict adherence to the purpose of the system, say, traffic monitoring, will ensure that privacy is maintained at its best. Another ethical concern is the dilemma weather the model has equal performances across all type of vehicles. Given that the data set was collected at Bangladeshi traffic signal points, the system may seem to recognize only those vehicles most often encountered in Bangladesh. To overcome this bias it is important to have a larger dataset that includes different types of vehicles from different locations. Further, a specific concern for this model is to guarantee that no type of vehicle would be excluded from the trained model to a greater extend than the actual ratio of such vehicles in real-world examples, thus, guaranteeing fairness.

Also, the consequences of the deploying of the system in real-time applications should be also discussed. Although vehicle recognition can improve the traffic control, it needs to guarantee that the recognized vehicle is not likely to be wrong, leading the system to take wrong actions which can include traffic tickets, or wrong classifications. The ethical use of the model can be maintained by explaining to anyone interested how the model works, a development of fair policies regarding the use of the model and ensuring that there are measures in place that punish those who use the model inappropriately or make mistakes.

# CHAPTER 6

# Overview of the Study, Conclusion, and Future Work

## 6.1 Overview of the Study

The research is concerned with the problem of real-time vehicle type identification by utilizing object detection models, namely, YOLOv9 variants, trained on the growing dataset of images captured at traffic signals in Bangladesh. The dataset consists of 281 images of 12 vehicle classes: bicycles, bikes, cars, buses, trucks and others. This work seeks to create a vehicle recognition system for real-time operation using a comprehensive approach that includes data acquisition, labeling, cleanup, expansion, model identification, training, and performance assessment. All annotation was done on Roboflow, and some preprocessing that includes resizing, auto-orientation, and image contrast adjustments were also done to improve the quality of the images in the dataset. Horizontal flipping and adjustment of image brightness were also applied to create additional images of the same, doubling the size of the data set, which helped to enhance the generalization capability of the model. Cross validation was implemented by splitting the data into 90% training set, 5% validation set and 5% strictly the test set. Therefore, i proceeded with model training using three YOLOv9 models, including YoloV9-C, YoloV9-E, and YoloV9-Gelan C. Each of the models was trained for 100 iterations having a batch size of 32 using Google Colab's free GPU. The assessment for all the models was done using the mean Average Precision (mAP) and i got that the YoloV9-E model was the most accurate with 73.66%.

Finally, the results obtained in the course of this study reveal the prospects of using YOLOv9 models for real-time recognition of vehicle type as well as provide essential information for advancing intelligent traffic surveillance systems and the notion of smart cities. Sequel work could be aimed at enhancing results obtained by the model and considering certain critical situations that may arise in real-world conditions.

## 6.2 Conclusions

To conclude, the vehicle type recognition based of YOLOv9 variants indicated that each of the model perform well in real-life Bangladeshi traffic signal dataset. When comparing between three models, YoloV9-E gives the highest results with 73.66 % in Average Precision (mAP), and accurately detect 12 classes of vehicles. When the model was tested using the careful optimization for faster prediction or with YoloV9-C, the mAP was slightly lower, 72.21; the model trained for more complex tasks using YoloV9-Gelan C gave the total mAP of 71.21. These results illustrate the performance comparison of model complexity, response time, and detection competency. Data augmentation strategies such as flipping and brightness proved very effective at improving model generalisation and my models ability to perform well under different conditions in reality. Their deployment on Streamlit further confirmed the models' feasibility for real-time vehicle recognition.

As the results demonstrate the medium performance of the models there is a remaining potential for better performance through the acquisition of more data, adjusting hyperparameters, and considering further specific effects like occlusion or overlaid cars. All in all, this work provides a solid baseline of recognizing vehicle types as a crucial step towards integration into a real-time traffic monitoring system.

## 6.3 Limitations

Despite the promising results, there are several limitations to the vehicle type recognition system using YOLOv9 variants. One major limitation is the relatively small size of the dataset, which consisted of only 402 images. While data augmentation helped increase the dataset size, it may not have been sufficient to capture the full diversity of real-world traffic conditions, such as varying weather conditions, vehicle occlusions, and diverse traffic environments. This could affect the model's generalization ability when deployed in new, unseen scenarios. Another limitation is the model's performance under challenging conditions, such as low visibility, poor lighting, or unusual vehicle orientations. These factors could negatively impact the accuracy of vehicle type detection, especially in real-time applications. Additionally, although the YOLOv9 models performed well in terms of mAP, they may still struggle with detecting smaller vehicles or distinguishing similar-looking vehicles (e.g., between cars and vans) in crowded scenes.

### 6.4 Future Work

Future work in the vehicle type recognition task can focus on several key areas to further improve model performance and applicability. First, expanding the dataset with a larger and more diverse collection of images, including different weather conditions, lighting scenarios, and vehicle orientations, could help improve the generalization ability of the models. Additionally, exploring advanced data augmentation techniques, such as rotation, scaling, and occlusion handling, could further enhance model robustness. Another avenue for future work is fine-tuning the models for specific use cases, such as edge device deployment, by optimizing them for lower computational resources without sacrificing accuracy. Moreover, integrating the model with real-time traffic monitoring systems and evaluating its performance in live environments would allow for practical testing, enabling further refinement based on real-world challenges.

# REFERENCES


[1] B. Saha et al., “Bangladeshi Native Vehicle Detection in Wild,” *arXiv (Cornell University)*, May 2024.

[2] U. Saha, Imam Uddin Ahamed, and M. Iffat Hossain, “YOLOv8 for Bangla License Plate Recognition: Advancing Real-Time Object Detection in Localized Contexts,” pp. 413–418, Jul. 2024.

[3] M. N. Baig, R. Hajong, M. M. Patwary, M. S. Rahman, and H. A. Chowdhury, “BadODD: Bangladeshi Autonomous Driving Object Detection Dataset,” *arXiv (Cornell University)*, Jan. 2024.

[4] A. Saadeldin, M. M. Rashid, A. A. Shafie, and T. F. Hasan, “Real-time vehicle counting using custom YOLOv8n and DeepSORT for resource-limited edge devices,” *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 22, no. 1, pp. 104–112, Feb. 2024.

[5] I. Sarker et al., “Bangladeshi Vehicle Classification using Transfer Learning with YOLOv7,” Dec. 2023.

[6] Israt Jahan Khan, M. Fahim, Mahady Hasan Sabbir, Durba Morshaline Nejhum, Abu, and R. Rahman, “Vehicle Number Plate Detection and Encryption in Digital Images Using YOLOv8 and Chaotic-Based Encryption Scheme,” May 2024.

[7] Abu, Farhana Sultana Eshita, and Tanjina Helaly, “An Automated System for Wrong-Way Vehicle Detection using YOLO and DeepSORT,” Dec. 2023.

[8] S. U. A. Shovo, Md. G. R. Abir, Md. M. Kabir, and M. F. Mridha, “Advancing low-light object detection with you only look once models: An empirical study and performance evaluation,” *Cognitive Computation and Systems*, Sep. 2024.

[9] Md. Humayun Kabir, Mohammad Nadib Hasan, N. Ahmad, and H. Jaki, “Transfer Learning-Based Anomaly Detection System for Autonomous Vehicle,” Nov. 2023.

[10] N. Bhavana, M. M. Kodabagi, B. M. Kumar, P. Ajay, N. Muthukumaran, and A. Ahilan, "POT-YOLO: Real-Time Road Potholes Detection Using Edge Segmentation-Based Yolo V8 Network," *IEEE Sensors Journal*, vol. 24, no. 15, pp. 24802–24809, Aug. 2024.

[11] Y. Du, X. Liu, Y. Yi, and K. Wei, "Optimizing Road Safety: Advancements in Lightweight YOLOv8 Models and GhostC2f Design for Real-Time Distracted Driving Detection," *Sensors*, vol. 23, no. 21, pp. 8844–8844, Oct. 2023.

[12] Shao Xian Tan, Jia You Ong, M. Goh, and C. Tee, "Boosting Vehicle Classification with Augmentation Techniques across Multiple YOLO Versions," *JOIV International Journal on Informatics Visualization*, vol. 8, no. 1, pp. 45–45, Mar. 2024.

[13] M. F. S. Titu, M. A. Pavel, G. K. O. Michael, H. Babar, U. Aman, and R. Khan, "Real-Time Fire Detection: Integrating Lightweight Deep Learning Models on Drones with Edge Computing," *Drones*, vol. 8, no. 9, p. 483, Sep. 2024.

[14] G. Mujtaba and A. Jalal, "Drone-Based Traffic Surveillance using Semantic Segmentation and DeepSort," *ResearchGate*, 2024.

[15] G. Mujtaba, M. O. Yusuf, and A. Jalal, "Vehicle Detection and Classification in Aerial Images with YOLOv8 and CNN-BiLSTM," *ResearchGate*, 2024.

[16] Z. Song, Y. Wang, S. Xu, P. Wang, and L. Liu, "Lightweight Vehicle Detection Based on Mamba_ViT," *Sensors*, vol. 24, no. 22, pp. 7138–7154, Nov. 2024.

[17] Z. Peng, X. Qi, S. Wu, J. Shang, and L. Yang, "Multi-Scale Traffic Camera Image Detection Network Based on Improved YOLOv8," *Springer Pacific Rim International Conference on Artificial Intelligence*, 2024.

[18] R. Juliansyah, M. Rachman, and M. Al Amin, "Development of a Real-Time Traffic Density Detection Website Using YOLOv8-Based Digital Image Processing with OpenCV," *Journal of Information Science*, 2024.

[19] C. Zhou, Y. Liu, and Y. Yang, "A Privacy-Preserving License Plate Encryption Scheme Based on an Improved YOLOv8," *Signal Processing*, vol. 2025, no. 62, pp. 31–47, Jan. 2025.

[20] A. Akoushideh and S. Shahbahrami, "Counting Vehicle Types Using Deep Learning Algorithm in Video Surveillance Systems," *Multimedia Tools and Applications*, vol. 83, no. 4, pp. 6023–6045, Dec. 2024.

[21] D. Anh and B. Thai, "Aerial Vehicle Detection at Night: A Synthesized Dataset and Performance Evaluation of YOLOv8," *IEEE 13th International Conference on Intelligent Transportation Systems*, pp. 1–8, Oct. 2024.

[22] M. Patil and P. Munoli, "Hybrid Deep Learning and OCR-Based Approach to Real-Time Automatic Number Plate Identification," *SSRN Electronic Journal*, 2024.

[23] J. Lee and S. Cho, "Toward Enhanced Tracking of Tiny Airborne Objects Using YOLOv8 and Particle Filter," *AIAA SCITECH Forum*, pp. 1–15, Jan. 2025.

[24] N. Sundarakrishnan and S. Varier, "Vehicle Turn Pattern Counting and Short-Term Forecasting Using Deep Learning for Urban Traffic Management," *IEEE Transactions on Intelligent Transportation Systems*, vol. 26, no. 2, pp. 4012–4025, Jan. 2025.

# Plagiarism Report

REAL TIME BANGLADESHI VEHICLE TYPE RECOGNITION USING YOLOv9 VARIANTS

ORIGINALITY REPORT

| 15% | 11% | 7% | 9% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | Submitted to Daffodil International University<br>Student Paper | 3% |
| 2 | dspace.daffodilvarsity.edu.bd:8080<br>Internet Source | 3% |
| 3 | www.mdpi.com<br>Internet Source | 1% |
| 4 | Submitted to Postgraduate Institute of Medicine<br>Student Paper | <1% |
| 5 | Dinesh Goyal, Bhanu Pratap, Sandeep Gupta, Saurabh Raj, Rekha Rani Agrawal, Indra Kishor. "Recent Advances in Sciences, Engineering, Information Technology & Management - Proceedings of the 6th International Conference “Convergence2024” Recent Advances in Sciences, Engineering, Information Technology & Management, April 24–25, 2024, Jaipur, India", CRC Press, 2025<br>Publication | <1% |