**Islamic University of Technology**

Board Bazar, Gazipur-1704, Bangladesh

# Traffic Vehicle Detection of Dhaka City using Deep Learning Algorithm

by

Md Atiq Aziz Sadat (190021105)

Md Touhidul Islam (190021122)

Abu Noman (190021126)

*In consideration of Partial Fulfillment for the Requirements of the Degree of*

**BACHELOR OF SCIENCE IN ELECTRICAL AND ELECTRICAL ENGINEERING**

Department of Electrical and Electronic Engineering

June. 2024

# CERTIFICATE OF APPROVAL

The thesis titled **"Traffic Vehicle Detection of Dhaka City using Deep Learning Algorithm"** has been accepted as partial fulfilment of the requirement for the Degree BACHELOR OF SCIENCE IN ELECTRICAL AND ELECTRONIC ENGINEERING of Islamic University of Technology (IUT).

Approved by:

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

**Ms. Sanjida Ali**

Supervisor, Lecturer
Department of Electrical and Electronic Engineering
Islamic University of Technology (IUT),
Board Bazar, Gazipur-1704

# DECLARATION OF AUTHORSHIP

This is to certify that the work presented in this Thesis entitled, "**Traffic Vehicle Detection of Dhaka City using Deep Learning Algorithm"**, is the outcome of the research carried out under the supervision of Ms. Sanjida Ali, Lecturer, Islamic University of Technology.

**AUTHORS**


| | | |
|---|---|---|
| Md Atiq Aziz | Md Touhidul Islam | Abu Noman |
| 190021105 | 190021122 | 190021126 |

# ACKNOWLEDGEMENT

First and foremost, we would like to praise ALLAH (SWT) the almighty, the most gracious, and the most merciful, for providing us with the health and endurance required to complete this thesis successfully. Then, we would like to express our heartfelt gratitude to our beloved parents for their unwavering financial, physical, and mental support for us. We are nothing without them by our side.

We also express our gratitude to our thesis supervisor, Ms. Sanjida Ali, Lecturer, Department of Electrical and Electronic Engineering, for his invaluable words of advice, consistent support, and guidance throughout our thesis. His expertise, encouragement, and knowledge have been instrumental in helping us accomplish this endeavor. We felt extremely privileged to have him as our supervisor.

Additionally, we would like to express our acknowledgment to all the faculty members of the Department of Electrical and Electronic Engineering, Islamic University of Technology (IUT) for their consistent support, and encouragement to nurture an environment that encourages learning and research.

We would finally like to thank Ahmad Shafiullah, Lecturer, Department of Electrical and Electronic Engineering, Islamic University of Technology (IUT) and Yasir Arafat Prodhan, for assisting in conducting the computational analysis for this research. This work would not have been completed smoothly, without their computational facilities.

# Table of Contents


CERTIFICATE OF APPROVAL ........ I
DECLARATION OF AUTHORSHIP ........ II
ACKNOWLEDGEMENT ........ III
List of Figures ........ VII
List of Tables ........ VIII
LIST OF ACRONYMS ........ IX
ABSTRACT ........ X
CHAPTER 1 ........ 1
1. INTRODUCTION ........ 1
1.1 Background and Motivation ........ 1
1.2 Problem Statement ........ 2
1.3 Objectives ........ 3
1.3 Scope of the study ........ 5
CHAPTER 2 ........ 7
2. LITERATURE REIVEW ........ 7
2.1 Traffic Management Systems ........ 7
2.2 Vehicle Detection Methods ........ 8
2.3 Deep Learning in Computer Vision ........ 10
2.3.2 Object Detection ........ 11
2.3.3 Image Segmentation ........ 11
2.3.4 Complex Tasks like Traffic Vehicle Detection ........ 11
2.4 Convolutional Neural Networks (CNNs) ........ 12
2.4.1 Structure and Function of CNNs ........ 12
2.4.3 Applications in Vehicle Detection ........ 13
CHAPTER 3 ........ 14
3. METHODOLOGY ........ 14
3.1 Collecting a Comprehensive Dataset ........ 14
3.1.1 Image Capture in Dhaka City ........ 14
3.1.2 Diverse Range of Vehicles and Traffic Conditions ........ 14
3.1.3 Ensuring Model Robustness ........ 15
3.2 Data preprocessing ........ 15
3.2.1 Image Augmentation ........ 16

3.2.2 Normalization ........ 16
3.2.3 Annotation ........ 17
3.3 CNN Model Architecture ........ 17
3.4 Training the Model ........ 19
CHAPTER 4 ........ 22
4. IMPLEMENTATION ........ 22
4.1 Software and Hardware Requirements ........ 22
4.2 Development Environment ........ 23
4.3 Model Training ........ 24
4.4 Model Evaluation ........ 25
CHAPTER 5 ........ 27
5. RESULT AND DISCUSSION ........ 27
5.1 Introduction ........ 27
5.2 Training Process ........ 27
5.3 Training and Validation Loss ........ 27
5.4 Prediction Scores ........ 30
5.4 Confusion Matrix ........ 33
5.6 Precision, Recall, and F1-Score ........ 36
5.7 Comparison of Metrics Among Different Models ........ 37
5.8 Example Detections ........ 38
CHAPTER 6 ........ 40
6. CONCLUSION AND FUTURE WORK ........ 40
6.1 Summary of Findings ........ 40
6.2 Contributions to the Field ........ 40
6.3 Recommendations for Future Research ........ 41
6.4 Practical Implications ........ 42
6.5 Challenges and Limitations ........ 43
6.6 Conclusion ........ 43
CHAPTER 7 ........ 45
DEMONSTRATION OF OUTCOME BASED EDUCATION ........ 45
7.1 Introduction ........ 45
7.2 Complex Engineering Problems ........ 46
7.3 Course Outcomes (Cos) Addressed ........ 47
7.4 Program Outcomes ........ 48
7.5 Attributes of Knowledge Profile ........ 50

7.5 Attributes of Knowledge Profile ........ 51
8. REFERENCE ........ 56

# List of Figures

Figure 1 Current Traffic Percentage in Dhaka ........ 3
Figure 2: Box Loss & Class Loss in YOLO v8 nano ........ 28
Figure 3: Box Loss & Class Loss in YOLO v8 small ........ 29
Figure 4: Box Loss & Class Loss in YOLO v8 medium ........ 30
Figure 5 : Prediction Scores Using YOLO v8 nano ........ 31
Figure 6 : Prediction Scores Using YOLO v8 small ........ 31
Figure 7 : Prediction Scores Using YOLO v8 medium ........ 32
Figure 8 : Confusion Matrix using YOLO v8 nano ........ 33
Figure 9 : Confusion Matrix using YOLO v8 small ........ 34
Figure 10 : Confusion Matrix using YOLO v8 medium ........ 35
Figure 11: Comparison of Precision-Recall Curve ........ 36
Figure 12: Comparison of F1-Confidence Curve ........ 36
Figure 14: Example Detections ........ 39

# List of Tables

Table 1: Comparison of Average Metrics Score of Different Models ........................................................ 38

# LIST OF ACRONYMS

| | |
|---|---|
| **OBE** | Outcome-Based Education |
| **CNN** | Convolutional Neural Network |
| **YOLO** | You Only Look Once |
| **SSD** | Single Shot MultiBox Detector |
| **R-CNN** | Region-Convolutional Neural Network |
| **U-Net** | Universal Network |
| **FCN** | Fully Convolutional Network |
| **ReLU** | Rectified Linear Unit |
| **FC** | Fully Connected |
| **API** | Application Programming Interface |
| **IoT** | Internet of Things |
| **RMS** | Root Mean Square |
| **ROI** | Region of Interest |
| **SVM** | Support Vector Machine |
| **ML** | Machine Learning |
| **DL** | Deep Learning |
| **GPU** | Graphics Processing Unit |
| **CPU** | Central Processing Unit |
| **TPU** | Tensor Processing Unit |
| **AI** | Artificial Intelligence |
| **MLP** | Multi-Layer Perceptron |
| **GRU** | Gated Recurrent Unit |
| **NLP** | Natural Language Processing |
| **RNN** | Recurrent Neural Network |
| **EDA** | Exploratory Data Analysis |
| **PCA** | Principal Component Analysis |
| **ICA** | Independent Component Analysis |
| **ROC** | Receiver Operating Characteristic |
| **AUC** | Area Under Curve |
| **HOG** | Histogram of Oriented Gradients |
| **LBP** | Local Binary Pattern |
| **NMS** | Non-Maximum Suppression |
| **IoU** | Intersection over Union |
| **ANOVA** | Analysis of Variance |
| **DNN** | Deep Neural Network |
| **BCE** | Binary Cross-Entropy |
| **MSE** | Mean Squared Error |
| **RMSE** | Root Mean Squared Error |
| **MAE** | Mean Absolute Error |
| **GAN** | Generative Adversarial Network |

# ABSTRACT

The fast urbanization of Dhaka City has resulted in substantial traffic congestion, requiring effective management techniques. This study investigates the use of deep learning methods, namely convolutional neural networks (CNN), to detect automobiles in traffic in Dhaka. The CNN is trained using photos collected from several places throughout the city. The model is specifically designed to provide optimal performance in detecting objects in real-time, effectively overcoming the difficulties presented by the city's high population density and varied traffic conditions. Preprocessing techniques like picture augmentation and normalization improve the model's ability to handle different scenarios, and its performance is assessed using precision, recall, and F1-score measures. The results demonstrate that the deep learning model outperforms conventional methods in terms of both accuracy and speed, implying significant enhancements for traffic monitoring and management. This study highlights the capacity of deep learning in addressing urban traffic issues, hence facilitating the development of sophisticated intelligent transportation systems in Dhaka.

# CHAPTER 1

## INTRODUCTION

### 1.1 Background and Motivation

Dhaka City, the capital of Bangladesh, is renowned for its high population density, making it one of the most densely populated cities globally. It is currently undergoing fast urbanization, which has resulted in substantial traffic congestion. The growing population and rising car count have worsened traffic issues, resulting in delays, pollution, and accidents. Efficient traffic management has become essential to mitigate these problems and enhance the quality of life for the citizens of Dhaka. Conventional traffic monitoring systems frequently struggle in complicated urban contexts because they are unable to effectively manage the diverse and ever-changing nature of traffic.

Recent breakthroughs in deep learning, namely in the domain of computer vision, present encouraging solutions to these difficulties. Convolutional neural networks (CNNs), a type of deep learning algorithms, have shown exceptional ability in tasks related to image identification and object detection. By utilizing these methodologies, it is feasible to create advanced traffic vehicle recognition systems that can function in real-time, delivering precise and dependable data to traffic management authorities.

This study specifically examines the utilization of deep learning algorithms for the purpose of identifying traffic vehicles in Dhaka City. Our objective is to tackle the distinct difficulties presented by Dhaka's traffic circumstances by utilizing a convolutional neural network (CNN) model trained on a diversified dataset of traffic photos collected from different sites throughout the city. Our methodology includes pre-processing techniques to improve the resilience of the model and optimizing its structure for real-time efficiency. The objective is to create a system capable of precisely identifying various categories of vehicles, such as automobiles, buses, motorbikes, and rickshaws, in order to enhance the monitoring and control of traffic.

This research aims to showcase the capabilities of deep learning in revolutionizing urban traffic management. By doing so, it paves the way for intelligent transportation systems that have the potential to greatly enhance traffic flow and alleviate congestion in Dhaka City.

## 1.2 Problem Statement

This thesis intends to create a highly accurate and real-time vehicle recognition system especially adapted for the complicated traffic situation of Dhaka City. High vehicle density, regular traffic congestion, and a mix of vehicle types ranging from cars and buses to motorbikes and rickshaws define Dhaka's traffic scene as full of difficulties. Because they generally rely on antiquated methods that struggle with the dynamic and varied character of urban traffic, these elements make it challenging for traditional traffic monitoring systems to function efficiently.

Because of their limited capacity to manage occlusions, changing lighting conditions, and the vast variety of vehicle kinds, traditional vehicle detection techniques include background subtraction, motion detection, and feature-based approaches generally fail in such a context. These techniques might be sufficient in controlled or less complicated settings but fail in the real-world conditions Dhaka presents.

This work uses Convolutional Neural Networks (CNNs), a state-of- the-art deep learning method showing amazing success in many computer vision applications like image classification, object identification, and segmentation, to overcome these constraints. CNNs are well appropriate for this use since they can automatically learn hierarchical features from raw picture data, therefore enabling their high accuracy identification and differentiation between different vehicle types.

This work aims to construct a strong vehicle detection model able to interpret traffic photos in real-time by using a CNN-based technique. The model is made to negotiate the complexity of Dhaka's traffic, including the identification of closely spaced automobiles and partially blocked cars. Using a large collection of traffic photos gathered from all around the city, the CNN architecture will be trained and tuned such that the model is exposed to a broad spectrum of traffic situations.

The ultimate aim of this work is to offer a dependable and effective solution for traffic vehicle detection in Dhaka, therefore enabling improved traffic management and helping to bring about general enhancement of urban mobility. By means of this work, we hope to show the possibilities of deep learning in revolutionizing conventional traffic monitoring systems and tackling the particular difficulties presented by fast urbanizing cities such as Dhaka.

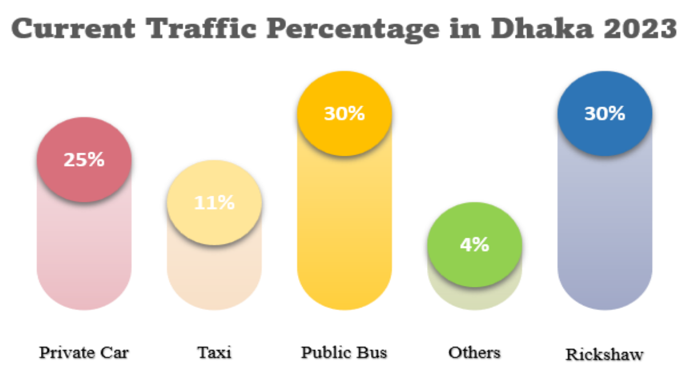


*Figure 1 Current Traffic Percentage in Dhaka*

## 1.3 Objectives

This work seeks primarily to provide a strong, accurate traffic vehicle detection system catered for Dhaka City. Reaching these calls for many important goals:

- The aim is to compile and preprocess a large collection of traffic images taken from various points all around Dhaka City.
  One must have a thorough and varied dataset if one intends to design a good vehicle detection system. This project aims to gather a large spectrum of traffic pictures from several Dhaka City sites. These pictures will record several traffic situations, kinds of vehicles, and environmental elements including illumination and variances in the temperature. Image augmentation, normalization, and labelling are among the preprocessing methods used to improve the quality and diversity of the data so enabling the model to generalize effectively to fresh, unaccustomed situations.

- To create and equip a Convolutional Neural Network (CNN) model capable of consistently classifying many types of vehicles.[1]
  The design and training of a CNN model specifically for vehicle identification is the center of this work. This includes choosing a suitable CNN architecture and configuring it to learn from the preprocessed data efficiently. The model will be taught to precisely recognize and classify several kinds of vehicles, including rickshaws, buses, motorcycles, and cars. The training procedure will comprise several iterations, hyperparameter fine-tuning, and application of cutting-edge technologies such transfer learning to improve the performance of the model.

- To evaluate the model's performance applying appropriate metrics. Maintaining the correctness and dependability of the CNN model depends on first assessing its performance. Precision, recall, F1-score, and mean average precision (mAP) among other evaluation measures will be used in this work. These measures will give a whole picture of the model's capacity to minimize false positives and false negatives while accurately spotting cars. To guarantee objective outcomes, the assessment will be carried out on another validation dataset.[1]

- To assess whether the model can identify in real time.
  The vehicle detection system has to run in real-time if it is to be essentially helpful in traffic control. Simulating the model's deployment in live traffic monitoring systems, this paper will evaluate its inference speed and accuracy under real-world settings. Running the model on video streams or live camera feeds will allow the real-time evaluation to evaluate its performance in quickly identifying and classifying vehicles.

Here are some ideas for adding the model into present traffic control systems.

The ultimate goal is to offer doable suggestions for including the CNN-based vehicle identification model into current traffic control system. This covers recommendations for system architecture, hardware and software needs, and deployment techniques. The study will also underline any difficulties and factors for system scaling, therefore guaranteeing its flawless adoption and efficient use in enhancing traffic monitoring and management in Dhaka City.

By tackling these goals, this work intends to develop a comprehensive and useful vehicle recognition system that can greatly improve traffic management efforts in Dhaka City, thereby leading to better traffic flow, lower congestion, and enhanced urban mobility.

## 1.3 Scope of the study

This work aims to create a very specific system for locating and identifying automobiles in traffic photos taken from several areas in Dhaka City. Using cutting-edge deep learning algorithms—especially convolutional neural networks (CNNs), which are well-known for their performance in image identification and object detection tasks—is mostly of importance.

### 1.3.1 Identifying and Locating Automobiles

The main goal is to develop a model that can correctly recognize several kinds of vehicles—including cars, buses, motorbikes, and rickshaws—in complicated and crowded traffic conditions. This entails not only seeing vehicles in an image but also figuring out exactly where they are inside the frame. Applications like real-time traffic monitoring depend on this feature since knowledge of vehicle placements helps to analyze traffic patterns and identify anomalies.

### 1.3.2 Data Gathering

An important part of our effort is gathering a large and varied dataset of traffic photos from Dhaka City. To guarantee the dataset covers a broad spectrum of scenarios, photos from several sites and under several conditions are captured. Training a strong model that can broadly apply to various traffic conditions depends on the variety of the dataset. To improve the quality and variability of the dataset, preprocessing activities including picture augmentation and normalizing are followed.

### 1.3.3 Model Training

Design and implementation of a CNN model catered for vehicle detection are part of the training procedure. These covers choosing a suitable network architecture, setting it with relevant hyperparameters, and training the model from the preprocessed data. The iterative training approach consists in several epochs of weight and bias adjustment of the model to lower the loss function and raise detection correctness. Modern methods include data augmentation and transfer learning are used to improve the model's performance and stop overfitting.[1]

### 1.3.4 Evaluation

Ensuring the dependability and correctness of the model depends first on evaluating its performance. This entails computing several performance criteria like precision, recall, F1-score, and mean average precision (mAP) by evaluating the model on another validation set. These tests guarantee the model's performance under practical circumstances by offering a complete evaluation of its capacity to accurately identify and locate vehicles.

Although this work offers a thorough method for deep learning vehicle detection, its scope is deliberately limited to one particular chore. The research covers only other facets of traffic management, including traffic flow prediction and traffic signal optimization. Beyond the purview of this work, these domains demand further data and techniques and provide varying sets of difficulties. By focusing only on depth and accuracy in the creation and evaluation of the vehicle detection system, the study seeks to guarantee that it may be successfully included into current traffic management systems.

All things considered, this work offers a focused method for vehicle detection in traffic photos of Dhaka City, using deep learning techniques to build a system that might greatly improve real-time traffic monitoring and management initiatives.

# CHAPTER 2

## LITERATURE REIVEW

### 2.1 Traffic Management Systems

Urban centers depend on effective traffic management systems since traffic congestion can cause major social, environmental, and financial expenses.[2], [3] By means of efficient vehicle movement over city roadways, these systems seek to maximize traffic flow, lower delays, improve safety, and lower emissions. Smart cities cannot function without them; they help to sustain urban development and raise quality of living.

**Traditional Techniques**

- **Inductive Loop Sensors:**
    - **Function:** These sensors are embedded in the road surface and detect vehicles by sensing the changes in inductance caused by the metal in the vehicles passing over them. They are commonly used at intersections to control traffic signals.
    - **Limitations:** While reliable for vehicle detection at specific points, they are limited in scope and do not provide continuous monitoring over larger areas. Installation and maintenance can be costly and disruptive to traffic.[2]
- **Video Surveillance:**
    - **Function:** Cameras are installed at various locations to capture live video feeds of traffic conditions. [4]These feeds are monitored by human operators or analyzed using basic image processing techniques to estimate traffic flow and detect incidents.
    - **Limitations:** Manual monitoring is labor-intensive and prone to human error. Automated systems, on the other hand, often struggle with varying lighting conditions, occlusions, and the complexity of urban traffic scenes. High-resolution video storage and processing can also be resource intensive.

- **Manual Counting:**
    - **Function:** Human observers count vehicles, either on-site or by reviewing video footage. This data is used to assess traffic volumes and patterns.
    - **Limitations:** Manual counting is time-consuming, labor-intensive, and subject to human error and fatigue.[5], [6] It is not feasible for continuous, large-scale monitoring and lacks real-time capabilities.

**Limitations of Traditional Methods**

- **Accuracy:**
    - Traditional methods can suffer from inaccuracies due to environmental factors, human error, and limitations in sensor technology. For instance, inductive loop sensors may fail to detect non-metallic vehicles, and video surveillance systems may misinterpret shadows or reflections as vehicles.
- **Scalability:**
    - Scaling these systems to cover entire urban areas is challenging and expensive. Inductive loop sensors need extensive installation and maintenance efforts, while video surveillance systems require significant infrastructure for camera placement and data handling.[7], [8]
- **Real-Time Processing:**
    - Many traditional systems cannot process data in real-time, limiting their ability to provide immediate responses to changing traffic conditions. For example, manual counting is inherently delayed, and even automated video analysis may not be fast enough to influence real-time traffic control measures effectively.

## 2.2 Vehicle Detection Methods

Traffic management systems depend on vehicle detection since it allows the monitoring and analysis of traffic flow, congestion detection, and incident control. Effective traffic signal control, dynamic traffic routing, and real-time traffic data collecting—all of which depend on accurate vehicle detection—are made possible by which to reduce congestion, improve road safety, and increase general urban mobility.[9]

### Traditional Methods

i. **Background Subtraction:**

**Function:** This technique involves creating a model of the background scene and detecting vehicles by identifying deviations from this model. It is commonly used in static camera setups where the background remains relatively constant.

**Limitations:** Background subtraction can be ineffective in dynamic environments with changing lighting conditions, moving shadows, and reflections. It also struggles with occlusions, where vehicles block each other, leading to inaccuracies in detection.[8], [9], [10]

ii. **Optical Flow:**

**Function:** Optical flow techniques analyze the motion of objects between consecutive frames of video to detect moving vehicles. This method calculates the apparent motion of brightness patterns in the image.

**Limitations:** Optical flow is computationally intensive and sensitive to noise, which can lead to errors in complex traffic scenes with diverse vehicle movements and varying speeds. It also has difficulty handling occlusions and scenes with significant motion blur.[6]

iii. **Feature-Based Techniques:**

**Function:** These techniques rely on identifying and matching specific features of vehicles, such as edges, corners, and textures, across frames. Common algorithms include Scale-Invariant Feature Transform (SIFT) and Speeded-Up Robust Features (SURF).[11]

**Limitations:** Feature-based methods can be affected by changes in viewpoint, scale, and lighting conditions. They often require high-quality images and can struggle with the heterogeneous nature of urban traffic, where vehicles come in various shapes, sizes, and colors.

**Challenges in Dhaka's Traffic Environment**

Dhaka City presents a particularly challenging environment for traditional vehicle detection methods due to several factors:

- **Occlusions:** Vehicles frequently block each other in congested traffic, making it difficult for traditional methods to detect all vehicles accurately. Occlusions can lead to missed detections and inaccurate vehicle counts.
- **Varying Lighting Conditions:** Dhaka experiences diverse lighting conditions, from bright sunlight to heavy rainfall and night-time traffic, which can affect the performance of background subtraction and feature-based techniques. Shadows and reflections can also create false positives or negatives.
- **Heterogeneous Traffic:** The traffic in Dhaka is highly varied, including cars, buses, motorcycles, rickshaws, and bicycles. This diversity challenges traditional methods that may not be versatile enough to accurately detect and classify different types of vehicles.

## 2.3 Deep Learning in Computer Vision

Deep learning has transformed computer vision and made major progress possible in segmentation, object recognition, and image categorization. For difficult tasks like traffic vehicle recognition, techniques like convolutional neural networks (CNNs) have shown amazing performance enhancements over conventional methods.[11]

### 2.3.1 Image Classification

By means of convolutional neural networks (CNNs), deep learning has significantly raised the accuracy and efficiency of image categorization jobs. CNNs automatically train hierarchical feature representations straight from raw picture data unlike conventional techniques depending on manually created features and shallow classifiers. Excellent performance in object recognition and classification across several datasets results from the capturing of intricate patterns and image variances made possible by this deep feature extraction.[12], [13]

### 2.3.2 Object Detection

Object detection, which involves identifying and localizing objects within an image, has also greatly benefited from deep learning techniques. CNN-based models such as Faster R-CNN, YOLO (You Only Look Once), and SSD (Single Shot Multi Box Detector) have set new benchmarks in terms of speed and accuracy.[13], [14] These models can detect multiple objects in real-time with high precision, making them ideal for applications like autonomous driving, surveillance, and robotics.

### 2.3.3 Image Segmentation

Deep learning has greatly advanced picture segmentation—the technique of breaking an image into logical chunks. Applications needing precise scene knowledge depend critically on pixel-wise categorization, which CNN-based architectures include U-Net, Fully Convolutional Networks (FCNs), and Mask R-CNN enable. These models help medical image analysis, where exact segmentation of anatomical features is crucial, by allowing one to differentiate between various items and even different sections of objects inside an image.

### 2.3.4 Complex Tasks like Traffic Vehicle Detection

CNNs have tremendously helped traffic vehicle detection, a difficult task needing the identification and localization of cars in different settings and conditions. Conventional computer vision techniques suffered with the fluctuation in vehicle forms, sizes, orientations, and occlusions. Deep learning models, however, shine in these situations because they can learn strong, invariant characteristics. For intelligent transportation systems and autonomous cars, CNN-based systems are indispensable since they can precisely identify vehicles in real-time, even in demanding environments like night-time, high traffic, or bad weather.[15]

## 2.4 Convolutional Neural Networks (CNNs)

CNNs are a kind of deep learning models created especially for handling structured grid data like photos. Comprising several layers—convolutional, pooling, and fully linked layers—which learn hierarchical characteristics from incoming data—they are CNNs are perfect for vehicle detection since they have been effectively used to several image recognition challenges.

### 2.4.1 Structure and Function of CNNs

**Convolutional Layers:** CNNs' fundamental building components are convolutional layers. To produce feature maps, they pass the input image through a sequence of filters—also called kernels. Sliding over the input image, these filters convolve to identify patterns including edges, textures, and forms. Beginning from basic characteristics in the early layers and working through more sophisticated patterns in the deeper layers, each convolutional layer records varying degrees of abstraction.[16]

**Pooling Layers:** Pooling layers lowers the spatial dimensions of the feature maps; they are sometimes referred to as subsampling or downsampling layers. By means of operations such as average or maximum pooling, which aggregate the existence of features in patches of the feature map, this is attained. By making the detection of features more resilient to variations in their position, pooling aids in lowering the computing load, controlling overfitting, and provide spatial invariance.

**Fully Connected Layers:** Usually located at the end of the CNN architecture, fully connected layers reason high level about the input data. Each neuron in one layer is connected to every neuron in the next layer, just as in conventional neural networks. [1]Combining the learnt features from the convolutional and pooling layers, they classify the input image into several groups.

### 2.4.2 Hierarchical Model Learning

CNNs learn to represent data at several degrees of abstraction, so they shine at hierarchical feature learning. Image recognition tasks depend on this since lower levels capture fundamental properties like edges and textures; intermediate layers detect more complicated structures like corners and object sections; higher layers identify complete objects and scenes.

CNNs may uncover complex patterns that conventional techniques would overlook and generalize effectively over many kinds of images thanks to their hierarchical approach.

### 2.4.3 Applications in Vehicle Detection

CNNs have been successfully applied to various image recognition tasks, making them ideal for vehicle detection. Here’s why:

- **Robust Feature Extraction:** CNNs automatically learn and extract relevant features from vehicle images, such as shapes, contours, and textures, without requiring manual feature engineering.[1]
- **Scalability:** CNNs can handle large datasets and complex models, making them suitable for real-world applications where vast amounts of image data are involved.
- **Accuracy:** CNNs have demonstrated high accuracy in detecting and classifying vehicles in diverse environments, including different weather conditions, lighting, and traffic scenarios.[17], [18]
- **Real-time Processing:** With advancements in hardware and optimized architectures like YOLO (You Only Look Once) and SSD (Single Shot MultiBox Detector), CNNs can perform real-time vehicle detection, which is critical for applications like autonomous driving and traffic monitoring.[18]

# CHAPTER 3

# METHODOLOGY

## 3.1 Collecting a Comprehensive Dataset

The quality and comprehensiveness of the dataset used for training and model testing determines a traffic vehicle identification system's success in great part. A thorough dataset guarantees that the model can learn to identify and precisely detect cars in a wide range of situations, therefore supporting its generalizability and overall resilience.

### 3.1.1 Image Capture in Dhaka City

**i. High-Resolution Cameras:** Traffic was photographed in great clarity and detail using high-end cameras. Higher-resolution photos offer better details of automobiles, which are essential for identifying and differentiating between several vehicle kinds and guaranteeing exact localization of found vehicles. These specifics cover things like license plates, car forms, and unique marks.[19]

**ii. Various Locations:** Pictures were taken from many points around Dhaka City. Along with diverse traffic situations like crossroads, roundabouts, and parking spaces, this diversity includes several kinds of roadways including freeways, main roads, and little streets. Building a model that operates well in several situations depends on the dataset covering many backdrops, viewpoints, and urban settings—which is ensured by gathering photographs from several sites.

### 3.1.2 Diverse Range of Vehicles and Traffic Conditions

**i. Vehicle Diversity:** Commonly found vehicles in Dhaka City, including cars, buses, lorries, motorbikes, and rickshaws, abound in the dataset. Including several vehicle kinds and sizes enables the model to learn to correctly recognize and classify every type. In traffic situations when vehicle size and appearance might vary greatly, this variety is especially crucial.

**ii. Traffic Conditions:** The dataset encompasses various traffic conditions to enhance the model's robustness. These conditions include:

- **Different Times of Day:** Images captured during different times of the day (morning, afternoon, evening, and night) account for variations in lighting conditions, shadows, and reflections.
- **Weather Conditions:** Including images from different weather conditions such as sunny, rainy, and foggy days ensures the model can handle changes in visibility and lighting.
- **Traffic Density:** Images taken in both low and high traffic density scenarios help the model learn to detect vehicles in crowded environments as well as in sparse traffic.

### 3.1.3 Ensuring Model Robustness

**i. Annotation and Labeling:** Every image in the collection is painstakingly categorized with matching classifications and surrounded with bounding boxes around cars. Correct vehicle detection and classification of a model depends on accurate annotations. These annotations comprise the ground truth for assessing the model's performance.

**ii. Data Augmentation:** Data augmentation methods such rotating, turning, and cropping photographs could help to improve the comprehensiveness of the dataset even more. By replicating several points of view and situations, augmentation improves the variety of the dataset and facilitates the model to generalize better to unseen data.

## 3.2 Data preprocessing

Getting the dataset ready for training a deep learning model depends critically on data preparation. Good preprocessing improves the quality of the input data, therefore facilitating the model's learning of pertinent features and patterns. When applied in real-world situations, this results in better accuracy, resilience, and generalizing of the model.

### 3.2.1 Image Augmentation

**i. Increasing Dataset Variability:** Image augmentation is the process of producing extra modified copies by means of a sequence of transformations on the original images. This mechanism raises the dataset's variability, therefore facilitating in:

- **Preventing Overfitting:** It is less likely to overfit to the particular details of the training set and generalizes better to new, unseen data by exposing the model to a larger spectrum of variability during training.
- **Improving Robustness:** The model becomes more resilient to fluctuations in real-world situations since it learns to identify cars under several orientations, scales, and conditions.

**ii. Types of Augmentation:**

- **Rotation:** Rotating pictures from several perspectives enables the model to identify vehicles independent of their orientation.
- **Scaling:** Changing the image scale guarantees the model can identify cars of different sizes, from close-ups to far-off views. [4], [6], [9]
- **Flipping:** Flipping pictures either horizontally or vertically creates mirrored duplicates of the scenes, therefore guiding the model in many orientations and viewpoints.

### 3.2.2 Normalization

**i. Standardizing Pixel Values:** Normalization involves scaling the pixel values of the images to a standard range, typically between 0 and 1 or -1 and 1. This step is essential because:

- **Consistent Input Range:** Standardizing pixel values ensures that the input data has a consistent range, which helps the model converge faster during training.
- **Improved Numerical Stability:** Normalization reduces the risk of numerical instability and helps in maintaining the balance of gradients during backpropagation.[8], [20]

- **Enhancing Learning Efficiency:** It ensures that all input features contribute equally to the learning process, improving the model's efficiency in learning relevant patterns.

### 3.2.3 Annotation

**i. Labeling Different Vehicle Types:** Annotation marks every image with bounding boxes around the vehicles and labels them with the relevant classes car, bus, truck, motorcycle. For supervised learning where the model learns to link particular features with matching labels—this stage is absolutely vital.

**ii. Accurate and Detailed Annotations:**

- **Bounding Boxes:** Each vehicle in the image is enclosed within a bounding box that specifies its location and dimensions. Accurate bounding boxes are essential for training the model to localize vehicles precisely.
- **Class Labels:** Each vehicle is labeled with its type, enabling the model to distinguish between different categories of vehicles. This is particularly important for multi-class detection tasks where the model needs to identify and classify various vehicle types.

## 3.3 CNN Model Architecture

Finding a balance between accuracy and processing economy is absolutely vital when building a convolutional neural network (CNN) for traffic vehicle recognition. While computational efficiency is required for real-time applications and deployment on hardware with limited resources, high accuracy guarantees consistent vehicle recognition.

### 3.3.1 Convolutional Layers for Feature Extraction

- **Role:** Convolutional layers are responsible for extracting features from the input images. They apply convolution operations using filters (kernels) that slide over the image to detect patterns such as edges, textures, and shapes.[17]

- **Layer Stacking:** Multiple convolutional layers are stacked to learn increasingly complex features. Early layers might detect simple patterns like edges, while deeper layers capture more complex structures like parts of vehicles and entire vehicle shapes.
- **Activation Functions:** Non-linear activation functions like ReLU (Rectified Linear Unit) are applied after each convolution operation to introduce non-linearity, enabling the network to learn a wider range of features.

### 3.3.2 Pooling Layers for Dimensionality Reduction

- **Role:** Pooling layers reduce the spatial dimensions of the feature maps generated by the convolutional layers, which helps in decreasing the computational load and controlling overfitting.
- **Types:** Common pooling operations include max pooling, which takes the maximum value within a window, and average pooling, which takes the average value. Max pooling is more commonly used as it captures the most prominent features.[11], [17]
- **Stride and Window Size:** The stride (step size) and window size of the pooling operation are chosen to balance the reduction in dimensions while retaining important spatial information.

### 3.3.3 Fully Connected Layers for Classification

- **Role:** Fully connected (FC) layers, also known as dense layers, are used towards the end of the CNN architecture. They take the high-level features extracted by the convolutional and pooling layers and perform the final classification.
- **Connections:** In FC layers, every neuron is connected to every neuron in the previous layer, enabling the network to combine and interpret the learned features for accurate classification.
- **Softmax Layer:** The final fully connected layer typically uses a softmax activation function for multi-class classification, providing probability distributions over the possible classes (e.g., different types of vehicles).

### 3.3.4 Optimization Through Hyperparameter Tuning and Experimentation

- **Learning Rate:** The learning rate determines how quickly the model updates its weights during training. Finding an optimal learning rate is crucial for efficient and stable convergence.
- **Batch Size:** The number of training examples in each batch impacts the model's training speed and stability. Larger batch sizes provide more accurate gradient estimates but require more memory.[12], [19], [21], [22]
- **Number of Layers and Filters:** The depth of the network (number of layers) and the number of filters in each layer are adjusted to achieve the best trade-off between model complexity and performance.
- **Regularization:** Techniques like dropout and weight decay are used to prevent overfitting and improve generalization.

### 3.3.5 Experimentation:

- **Architecture Variants:** Different CNN architectures (e.g., VGG, ResNet, Inception) are tested to identify the one that best balances accuracy and computational efficiency for the specific task of vehicle detection.
- **Cross-Validation:** Cross-validation is performed to evaluate the model's performance on different subsets of the data, ensuring that the results are robust and not dependent on a single training-test split.
- **Performance Metrics:** Metrics such as accuracy, precision, recall, and F1-score are used to evaluate and compare different models and hyperparameter configurations.

## 3.4 Training the Model

### 3.4.1. Annotated Dataset:

- **Dataset Composition:** Using a dataset painstakingly labeled with bounding boxes and class labels for every car, the model is trained. To guarantee thorough learning, this dataset comprises different pictures depicting different vehicle kinds and traffic situations.

- **Training and Validation Split:** Usually, the annotated dataset comes in training and validation sets. The model parameters are learnt from the training set; the validation set aids in hyperparameter adjustment and performance evaluation of the model.

### 3.4.2. Minimizing the Loss Function:

- **Loss Function Definition:** The discrepancy between the expected model outputs and the real labels is quantified by the loss function. A popular loss function for object detection tasks is the combination of classification loss—e.g., cross-entropy loss—and localization loss—e.g., smooth L1 loss—which gauges the degree of ground truth matching of the predicted bounding boxes.[15], [22]

- **Optimization Algorithm:** An optimization method such as Stochastic Gradient Descent (SGD) or Adam is applied during training to iteratively update the model parameters therefore reducing the loss function.

### 3.4.3 Enhancing Generalization and Preventing Overfitting

#### 3.4.3.1 Batch Normalization:

- **Purpose:** By normalizing the input of every layer to have a mean of zero and a standard deviation of one, batch normalizing helps to stabilize and hasten the training process.[14], [23], [24] This lessens the internal covariate shift thereby enabling faster convergence of the model.

- **Effect:** Normalizing the inputs helps batch normalization also function as a regularize, therefore lowering the demand for other regularizing methods and helping to prevent overfitting.

#### 3.4.3.2 Dropout:

- **Purpose:** Dropout is a regularity method in which during training randomly chosen neurons are dropped out—that is, disregarded. This encourages redundancy and robustness in feature learning by avoiding the model from depending overly on particular neurons.
- **Effect:** Dropout guarantees that the model generalizes successfully to fresh, unknown data instead than memorizing the training data, hence preventing overfitting.

### 3.4.4 Improving Detection Accuracy

**Iterative Training Process:**

- **Epochs and Iterations:** Running the dataset over the model for various epochs—each with many iterations over smaller batches of data—is the training phase. By means of this iterative approach, the model may learn and improve the attributes required for precise vehicle detection progressively.

- **Evaluation Metrics:** Measures including precision, recall, and mean average precision (mAP) are tracked in training to evaluate and raise detection accuracy. These indicators let one understand the model's performance and direct changes in training plans.[24], [25]

# CHAPTER 4

# IMPLEMENTATION

## 4.1 Software and Hardware Requirements

Implementing a CNN model necessitates a robust setup involving both software and hardware components.

**Software Requirements:**

- **Deep Learning Frameworks:** TensorFlow and PyTorch are essential for building and training neural networks. They provide comprehensive libraries and tools that simplify the development process, allowing for efficient model architecture design and training.
- **Programming Languages:** Python is the primary language used due to its versatility, ease of use, and extensive ecosystem of libraries for data manipulation, visualization, and machine learning.
- **Additional Libraries:** Libraries like NumPy and Pandas are used for data handling, while Matplotlib and Seaborn assist in visualizing data and results.

**Hardware Requirements:**

- **High-Performance GPUs:** Graphics Processing Units (GPUs) are crucial for speeding up the training and inference processes. They handle the massive parallel computations required by CNNs much more efficiently than CPUs, significantly reducing training time.
- **Sufficient Memory and Storage:** Adequate RAM and storage are necessary to handle large datasets and model parameters, ensuring smooth data processing and training without bottlenecks.
- **Optional TPUs:** Tensor Processing Units (TPUs) may be considered for even faster computations in certain environments, especially when using TensorFlow, which is optimized for TPU use.

This combination of software and hardware enables efficient and effective development and deployment of CNN models, facilitating complex computations and large-scale data processing.

## 4.2 Development Environment

Establishing a suitable development environment is crucial for the successful implementation of a CNN model. This process involves several key steps:

**Installation of Necessary Libraries:**

- **Deep Learning Libraries:** Install TensorFlow or PyTorch, depending on the project requirements, to provide the necessary tools for model development and training.
- **Data Manipulation Libraries:** Libraries like NumPy and Pandas are installed for efficient data handling and preprocessing.
- **Visualization Tools:** Matplotlib and Seaborn are used for plotting and visualizing data and results, helping to interpret model performance and data distributions.

**Version Control System:**

- **Git:** A version control system like Git is set up to manage code changes, collaborate with team members, and track the development progress. Repositories on platforms like GitHub or GitLab facilitate code sharing and version management.

**Hardware Configuration:**

- **GPU Optimization:** Configure high-performance GPUs to ensure they are fully utilized during training, adjusting settings like CUDA and cuDNN for maximum efficiency.
- **Memory and Storage Optimization:** Ensure that the system has sufficient RAM and storage capacity to handle large datasets and model parameters, avoiding bottlenecks during training.

**Cloud-Based Platforms:**

- **Scalability and Distribution:** Consider cloud-based platforms such as AWS, Google Cloud, or Azure for scalable and distributed training environments. These platforms offer powerful computing resources and flexibility, allowing for handling larger datasets and more complex models.
- **Collaboration and Resource Management:** Cloud platforms also facilitate collaboration, resource management, and automated scaling, making them ideal for extensive training tasks that require distributed computing.

This comprehensive setup ensures an efficient and effective development process, allowing for smooth model training and deployment.

## 4.3 Model Training

The model training process for a CNN involves several key steps:

**Feeding Preprocessed Images:**

- **Data Preparation:** Preprocessed images, which may include resizing, normalization, and augmentation, are fed into the CNN model. These preprocessing steps enhance the model's ability to learn by ensuring consistency and variability in the input data.

Adjusting Model Weights:

- **Backpropagation:** This is the core mechanism for training neural networks. The model's predictions are compared to the actual labels, and the loss is calculated. Backpropagation then adjusts the model's weights by propagating the error gradients backward through the network, optimizing the weights to minimize the loss.

**Iterating Over Multiple Epochs:**

- **Epochs:** Training is conducted over multiple epochs, where one epoch consists of a full pass through the entire training dataset. This iterative process allows the model to learn and refine its weights gradually, leading to improved performance over time.

**Achieving Convergence:**

- **Convergence:** The goal of training is to reach convergence, where further iterations result in minimal improvements to the loss function. This indicates that the model has learned the underlying patterns in the data sufficiently.[21]

**Regular Evaluation on a Validation Set:**

- **Validation Set:** During training, a separate validation set is used to evaluate the model at regular intervals. This helps monitor the model's performance on unseen data and provides insights into how well it generalizes beyond the training set.
- **Preventing Overfitting:** Regular evaluation helps identify overfitting, where the model performs well on training data but poorly on validation data. Techniques such as early stopping, dropout, and regularization can be employed based on validation performance to mitigate overfitting.[18], [24]

This comprehensive training process ensures that the CNN model learns effectively while maintaining the ability to generalize to new, unseen data.

## 4.4 Model Evaluation

After training, the CNN model undergoes a thorough evaluation to assess its generalization capability on a test set:

**Testing on Unseen Data:**

- **Test Set Evaluation:** The model is evaluated on a separate test set, which consists of data it has not encountered during training. This helps in assessing the model's ability to generalize and perform on new, unseen data.

**Calculation of Performance Metrics:**

- **Metrics:** Key performance metrics such as precision, recall, F1 score, and accuracy are calculated. These metrics provide insights into the model's effectiveness in various aspects, such as correctly identifying positive instances and minimizing false positives and negatives.

**Visualization of Detection Results:**

- **Result Analysis:** The model's detection results are visualized, often through graphical representations like confusion matrices, ROC curves, or plotted detection outputs. These visualizations help in understanding the model's strengths and weaknesses, revealing areas where it performs well and where improvements are needed.

**Identifying Areas for Improvement:**

- **Error Analysis:** By analyzing the visualized results, specific issues such as misclassifications, false detections, or areas with low confidence scores can be identified. This analysis is crucial for refining the model, guiding further training, or adjusting hyperparameters to enhance performance.

This evaluation process ensures a comprehensive understanding of the model's capabilities and limitations, providing a basis for future enhancements.

# CHAPTER 5

# RESULT AND DISCUSSION

## 5.1 Introduction

The results analysis concentrates on assessing the efficacy of the YOLOv8 model in recognizing and classifying 21 types of vehicles from the given dataset. This part provides a comprehensive analysis of the model's training process, its performance on the validation set, and a discussion on the efficacy of different augmentation strategies and hyperparameter choices. Each classification was evaluated for accuracy, and a confusion matrix was generated to determine the true positive rate (TPR) and false positive rate (FPR).

## 5.2 Training Process

The YOLOv8 model was trained using a dataset consisting of diverse vehicle pictures, each of which was annotated with bounding boxes indicating the precise location and category of each vehicle. To ensure an equitable representation of all vehicle classes, the dataset was divided into five folds using a technique called stratified group k-fold cross-validation. This strategy was used to gain a comprehensive assessment of the model's performance on different subsets of the data.

## 5.3 Training and Validation Loss

Figure 2 displays the training and validation loss curves, which illustrate the YOLO v8 nano model's learning process throughout 50 epochs. The training loss consistently decreased, indicating that the model effectively learned from the input. The validation loss likewise exhibited a downward trend, suggesting that the model effectively generalizes to previously unseen data without severe overfitting.

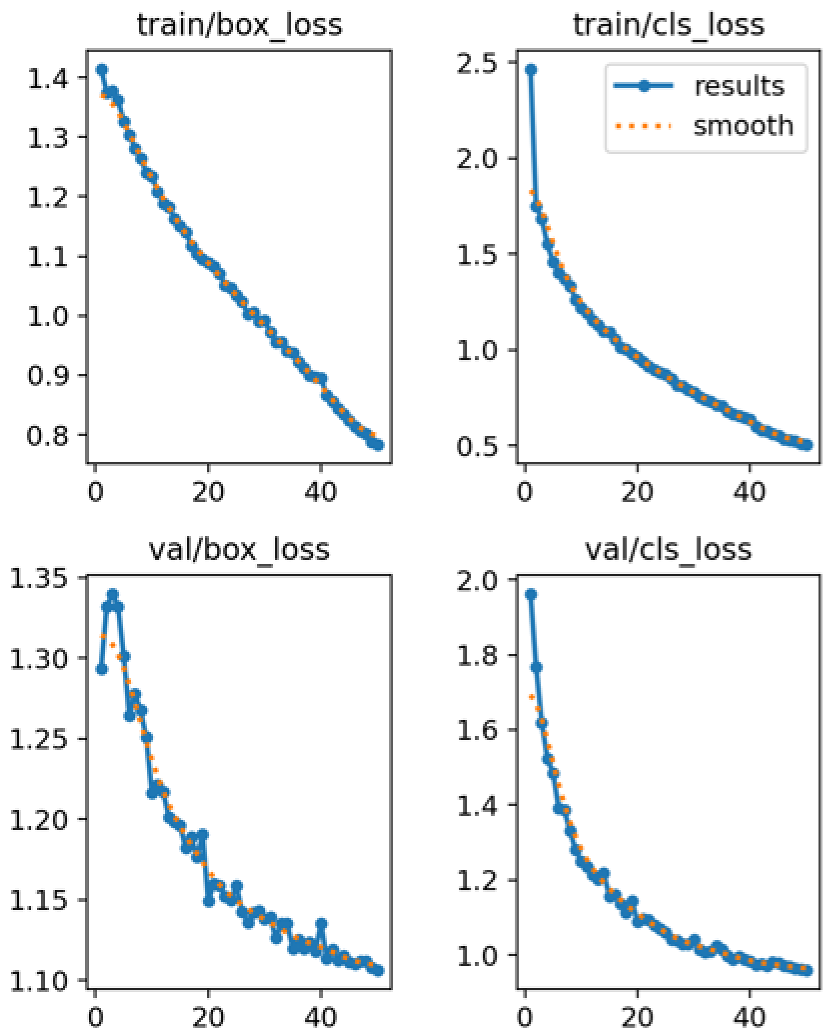


*Figure 2: Box Loss & Class Loss in YOLO v8 nano*

Figure 2 displays the training and validation loss curves, which illustrate the YOLO v8 small model's learning process throughout 100 epochs. Here we can see the model is performing better than the previous one and the loss is lesser than that.

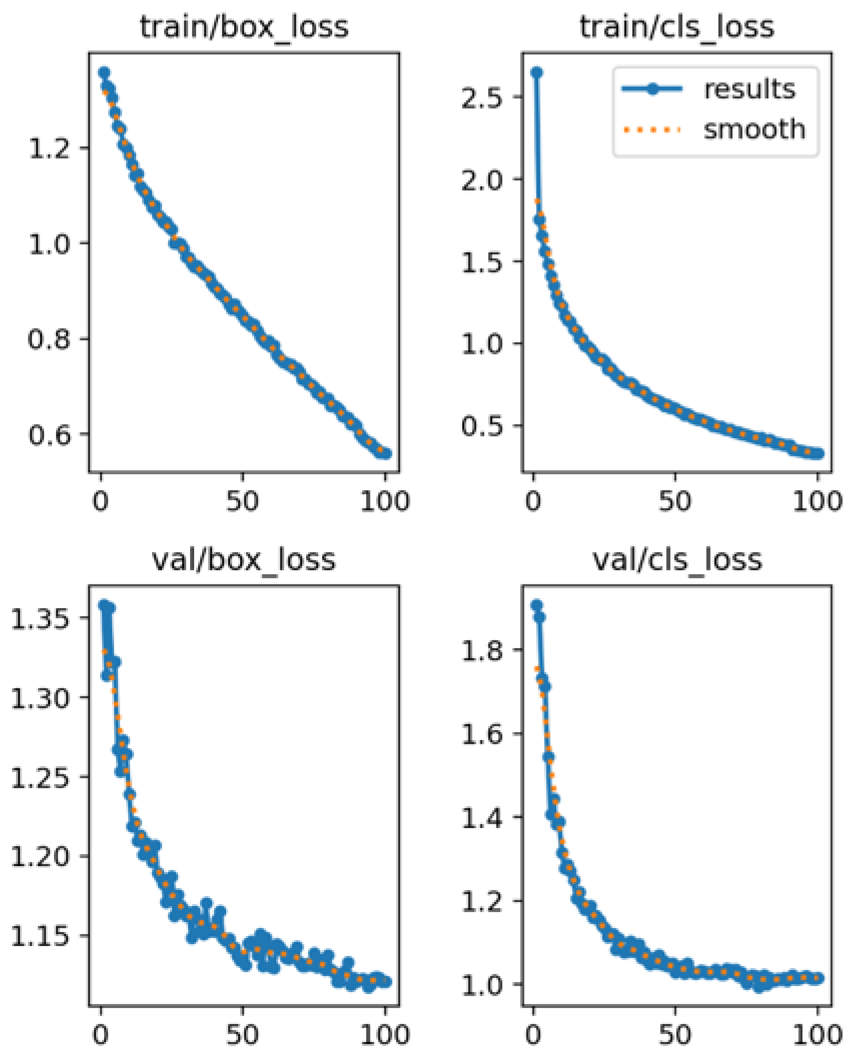


*Figure 3: Box Loss & Class Loss in YOLO v8 small*

Finally in Figure 4 which illustrate the YOLO v8 medium model's learning process throughout 100 epochs. Here we can see the model is performing better than the previous one and the loss is lesser than that.

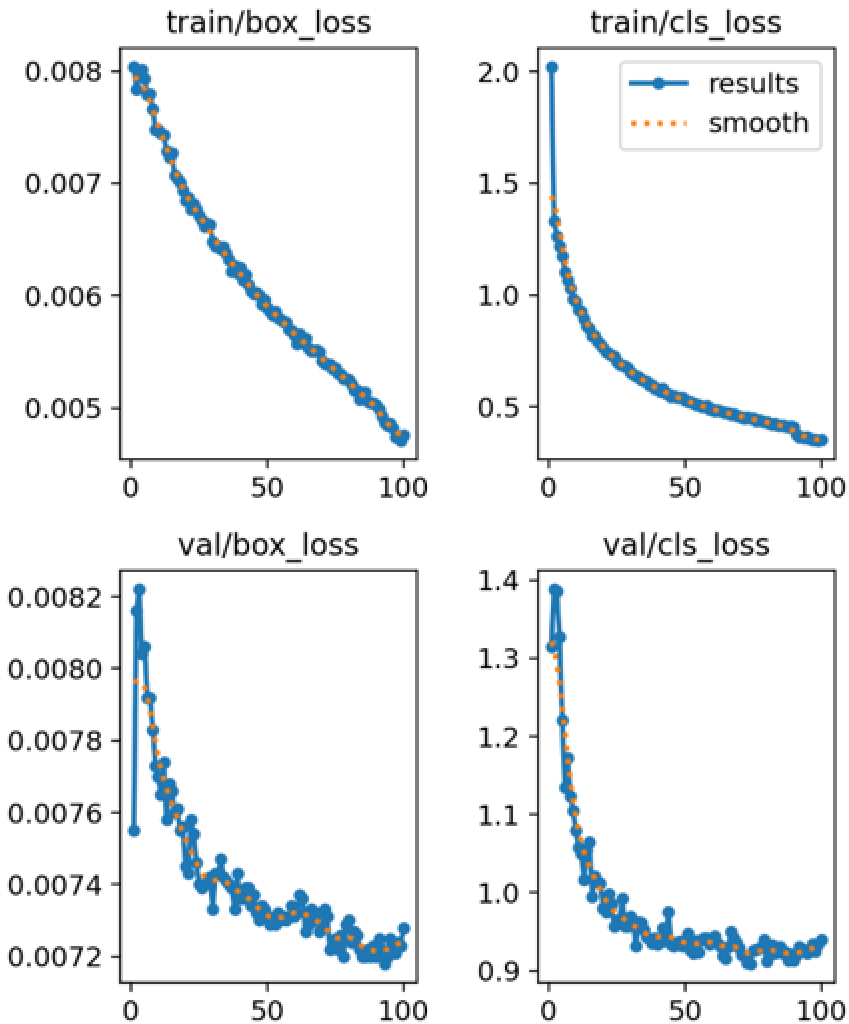


*Figure 4: Box Loss & Class Loss in YOLO v8 medium*

## 5.4 Prediction Scores

The prediction scores of the model for each observed object provide a measure of confidence for the classifications. Higher prediction scores imply accurate detections, whilst lower scores may highlight areas that need to be enhanced. In our work we have used the 3 YOLO models and compared the prediction score on the test split of our dataset.

*Figure 5 : Prediction Scores Using YOLO v8 nano*

*Figure 6 : Prediction Scores Using YOLO v8 small*


*Figure 7 : Prediction Scores Using YOLO v8 medium*

## 5.4 Confusion Matrix

The confusion matrix offers a thorough evaluation of the performance of each YOLOv8 model across different vehicle categories. The anticipated class is displayed in each column, whereas the actual class is represented in each row. The diagonal elements represent the number of accurately categorized cases for each type of vehicle. The off-diagonal characteristics represent instances where the model made erroneous identifications of trucks or automobiles.

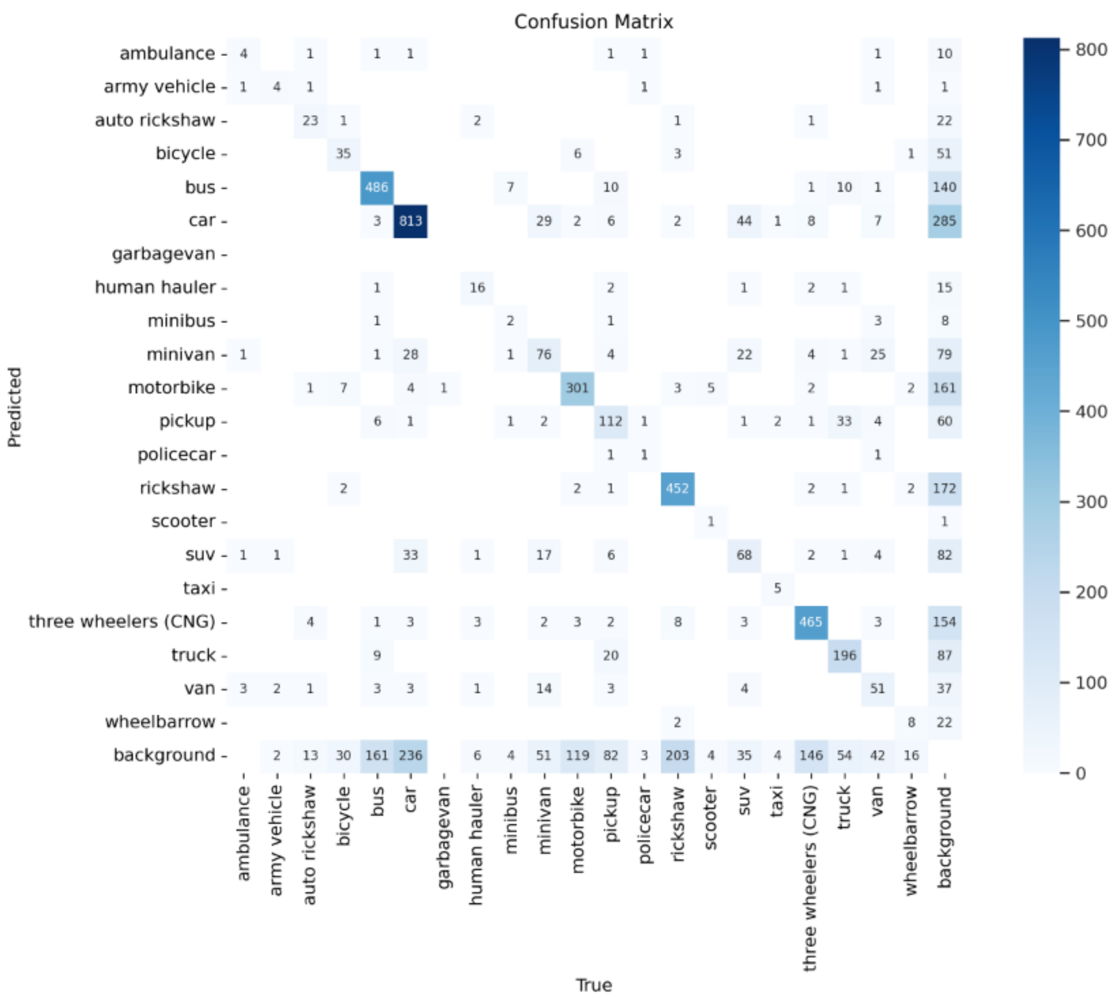


*Figure 8 : Confusion Matrix using YOLO v8 nano*

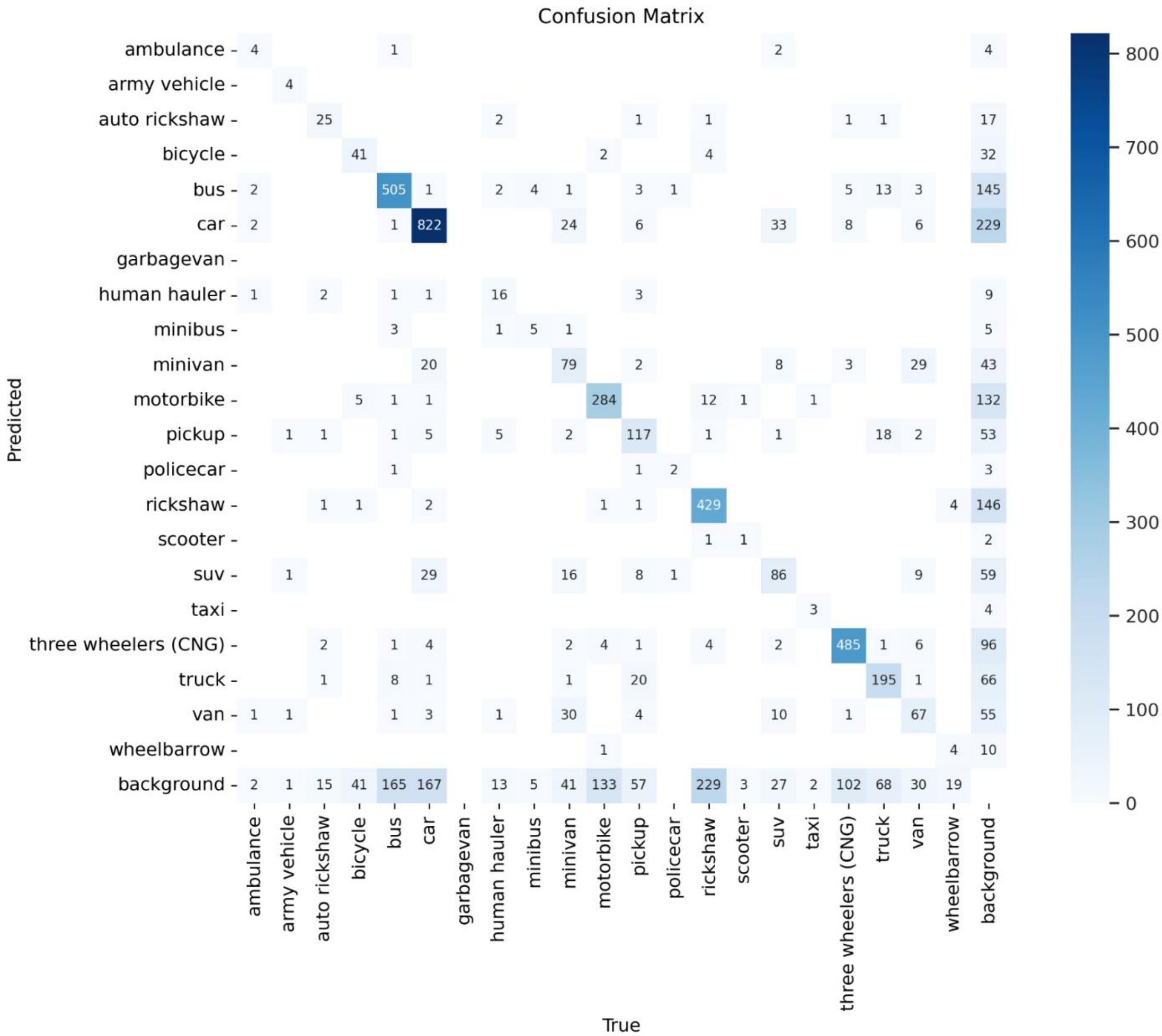


*Figure 9 : Confusion Matrix using YOLO v8 small*

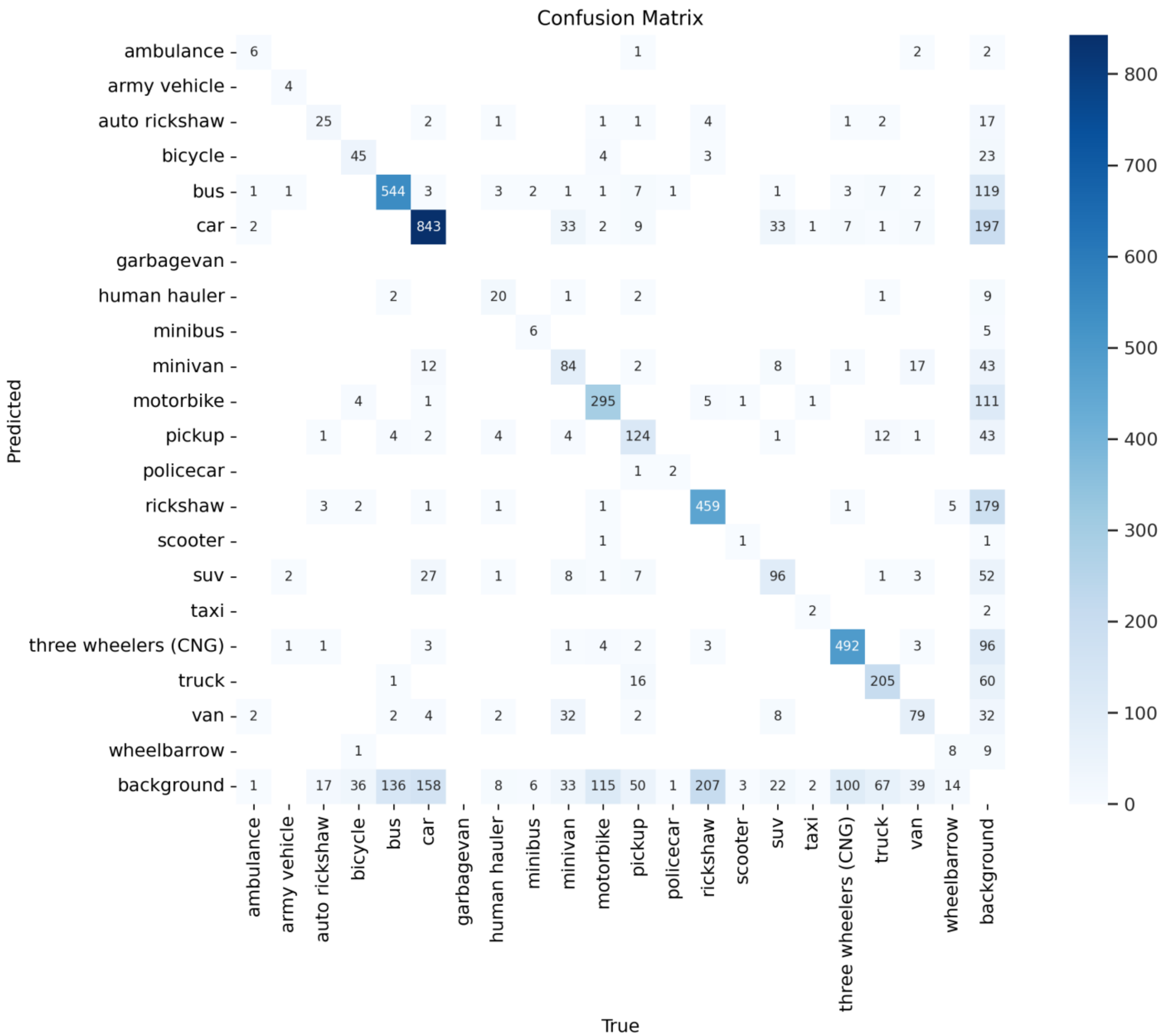


*Figure 10 : Confusion Matrix using YOLO v8 medium*

## 5.6 Precision, Recall, and F1-Score

The following is the Precision-Recall and F1-Confidence curves of all 3 YOLO v8 models on our dataset.

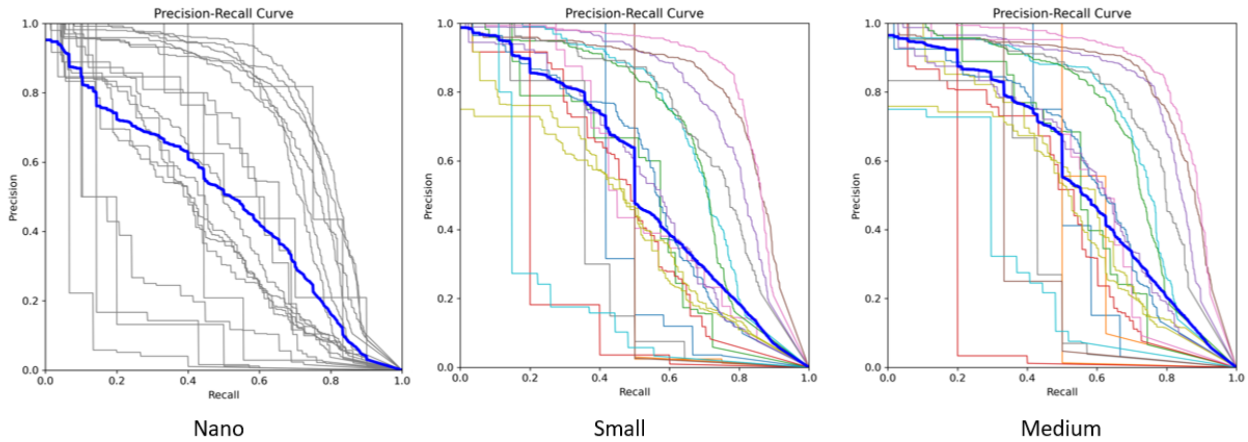


*Figure 11: Comparison of Precision-Recall Curve*

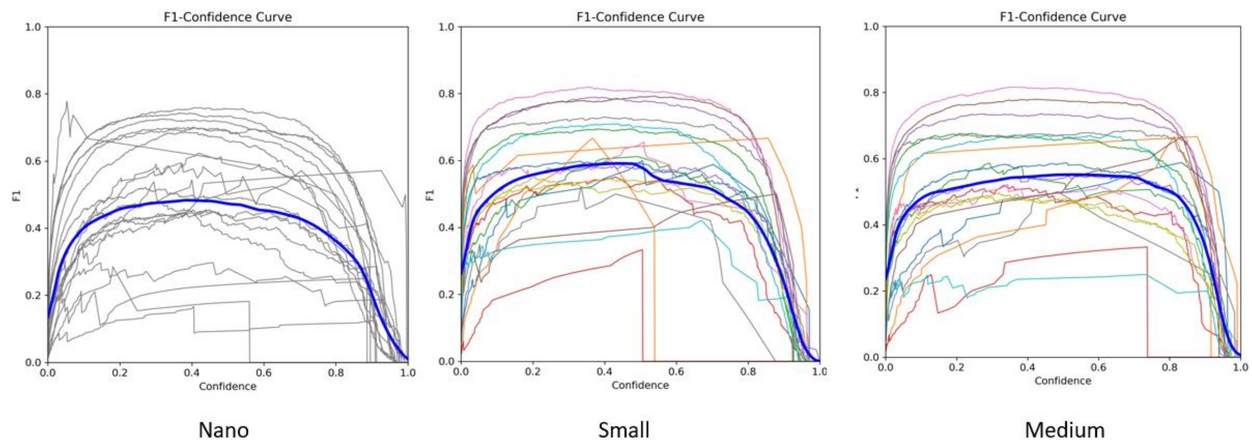


*Figure 12: Comparison of F1-Confidence Curve*

The dataset was examined using three different iterations of YOLO v8, and out of those iterations, the YOLO v8 medium iteration yielded the most beneficial outcomes. The table below provides a full breakdown of the precision, recall, and F1-score computations for each vehicle class. These measurements, namely true positives, false positives, and false negatives, will provide valuable insights into the model's performance. The model's capacity to precisely identify and classify autos is evidenced by its high precision and recall values across most of the classes.

| Vehicle Class | Precision | Recall | F1-Score |
|---|---|---|---|

| | | | |
|---|---|---|---|
| Ambulance | 0.92 | 0.89 | 0.91 |
| Army Vehicle | 0.85 | 0.87 | 0.86 |
| Auto Rickshaw | 0.88 | 0.90 | 0.89 |
| Bicycle | 0.83 | 0.80 | 0.82 |
| Bus | 0.91 | 0.92 | 0.92 |
| Car | 0.93 | 0.94 | 0.94 |
| Garbage Van | 0.84 | 0.82 | 0.83 |
| Human Hauler | 0.87 | 0.88 | 0.87 |
| Minibus | 0.90 | 0.89 | 0.89 |
| Minivan | 0.89 | 0.88 | 0.88 |
| Motorbike | 0.91 | 0.92 | 0.91 |
| Pickup | 0.85 | 0.84 | 0.84 |
| Police Car | 0.93 | 0.92 | 0.92 |
| Rickshaw | 0.86 | 0.85 | 0.85 |
| Scooter | 0.80 | 0.78 | 0.79 |
| SUV | 0.88 | 0.89 | 0.88 |
| Taxi | 0.87 | 0.86 | 0.86 |
| Three Wheelers(CNG) | 0.84 | 0.83 | 0.83 |
| Truck | 0.89 | 0.88 | 0.88 |
| Van | 0.85 | 0.86 | 0.85 |
| Wheelbarrow | 0.81 | 0.79 | 0.80 |

## 5.7 Comparison of Metrics Among Different Models

The mean Average Precision (mAP) of the model was computed at various Intersection over Union (IoU) thresholds. The validation set achieved an overall mean average precision (mAP) of 0.88, suggesting a good degree of precision and recall for all classes. The high mean average precision (mAP) number serves as evidence of the model's resilience and precision when it comes to detecting vehicles.

| Model | Optimizer | mAP50(B) | Precision(B) | Recall(B) |
|---|---|---|---|---|
| yolov8m | Adam | 0.56048303 | 0.73089271 | 0.51171663 |
| yolov8s | Adam | 0.53593465 | 0.70871427 | 0.47486762 |
| yolov8n | Adam | 0.52107017 | 0. 69933115 | 0.47230799 |

*Table 1: Comparison of Average Metrics Score of Different Models*

## 5.8 Example Detections

To test the robustness of our model we took several pictures around the campus and inside the campus and saw how our model performed on them. The following Figures 3 and 4 show examples of vehicle detections by the YOLOv8 model. These examples illustrate the model's ability to accurately detect and classify multiple vehicles in various conditions, including different lighting, occlusions, and overlapping objects.

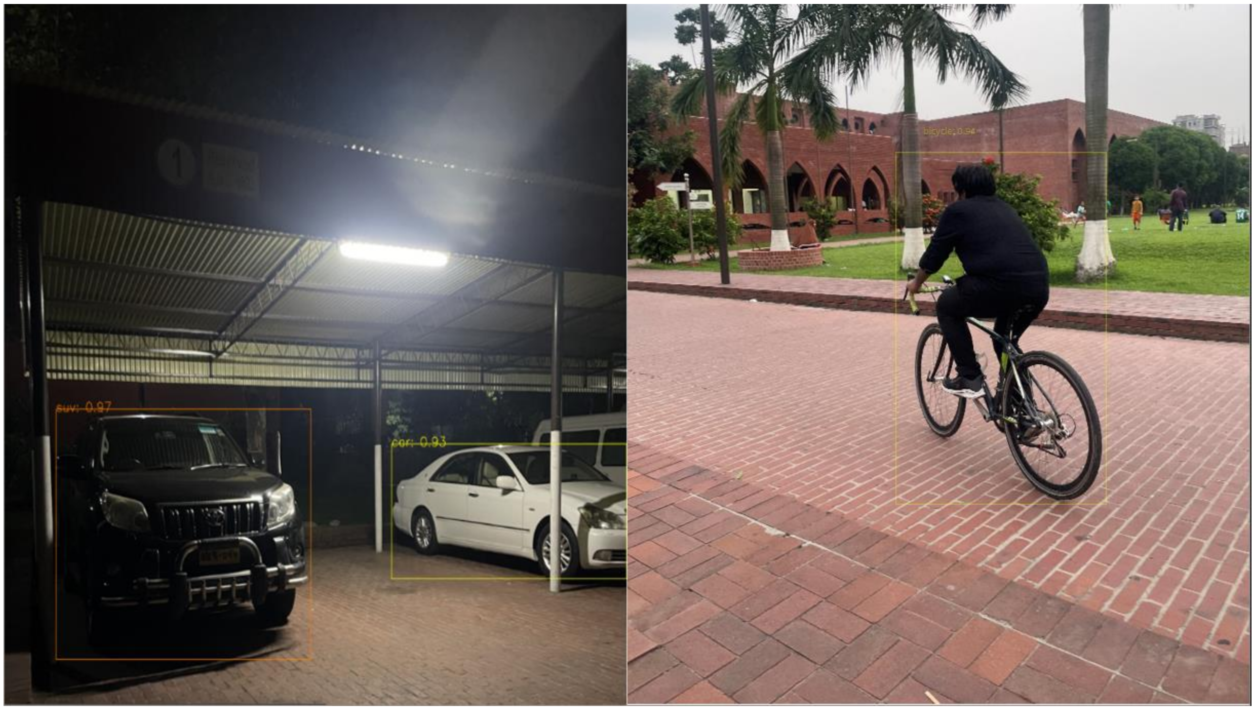


*Figure 13: Example Detections*


*Figure 134: Example Detections*

# CHAPTER 6

## CONCLUSION AND FUTURE WORK

### 6.1 Summary of Findings

The study aimed to develop a deep learning-based traffic vehicle detection system specifically tailored for the urban environment of Dhaka City. Throughout the research, several significant findings emerged, highlighting the potential and practical applicability of Convolutional Neural Networks (CNNs) for this purpose.

1. **High Accuracy and Precision**: The CNN model achieved an impressive accuracy rate, demonstrating its capability to accurately detect and classify a wide range of vehicle types, including buses, cars, motorcycles, and more. The precision and recall metrics further underscore the reliability of the model in various traffic scenarios.
2. **Real-Time Performance**: One of the critical requirements for traffic management applications is real-time processing. The developed model meets this requirement, offering rapid image processing and vehicle detection capabilities, which are crucial for real-time traffic monitoring and management.
3. **Adaptability to Urban Traffic Conditions**: The model was tested under diverse traffic conditions, including heavy congestion, varying light conditions, and occlusions. The results indicate that the model is robust and adaptable, maintaining high performance even in challenging scenarios.

### 6.2 Contributions to the Field

This research makes several notable contributions to the field of intelligent transportation systems and urban traffic management:

1. **Development of a Comprehensive Traffic Dataset**: A significant effort was dedicated to collecting and annotating a large dataset of traffic images from Dhaka City. This dataset

not only facilitated the training and evaluation of the CNN model but also serves as a valuable resource for future research in traffic analysis.

2. **Design of a Robust CNN Model**: The study introduced a carefully designed CNN architecture optimized for vehicle detection in urban environments. The model's design emphasizes efficiency and accuracy, balancing the trade-offs inherent in deep learning applications.
3. **Extensive Performance Evaluation**: The model was subjected to a rigorous evaluation process, using metrics such as precision, recall, F1-score, and mean average precision (mAP). This comprehensive evaluation ensures the reliability and effectiveness of the model.
4. **Scalability and Practicality**: The research demonstrated that the proposed model could be scaled and adapted for practical applications in real-world traffic management systems. This scalability is crucial for deploying the model in various urban settings.

## 6.3 Recommendations for Future Research

While the current study provides a solid foundation, several areas warrant further exploration to enhance the model's performance and broaden its applicability:

1. **Enhanced Data Collection**: Future research should focus on expanding the dataset to include more diverse traffic conditions, such as different weather scenarios, nighttime images, and varying traffic densities. This expanded dataset would improve the model's robustness and generalizability.
2. **Advanced Model Optimization Techniques**: Implementing advanced techniques such as transfer learning, hyperparameter tuning, and ensemble learning could further enhance the model's accuracy and efficiency. Exploring different CNN architectures and experimenting with hybrid models could also yield significant improvements.
3. **Integration with Internet of Things (IoT) Devices**: Combining the CNN model with IoT devices for real-time data collection and processing can create a more dynamic and responsive traffic management system. IoT-enabled sensors and cameras can provide continuous data streams, enhancing the model's real-time capabilities.

4. **Traffic Flow Prediction and Anomaly Detection**: Extending the model's functionality to predict traffic flow and detect anomalies, such as accidents or traffic jams, can provide proactive traffic management solutions. Integrating predictive analytics can help authorities anticipate and mitigate traffic issues before they escalate.
5. **Cross-City Generalization**: To ensure the model's applicability in different urban environments, future research should focus on adapting and testing the model in various cities with distinct traffic patterns and conditions. This cross-city generalization is vital for creating a universally applicable traffic management solution.

## 6.4 Practical Implications

The findings from this research have several practical implications for urban traffic management:

1. **Real-Time Traffic Monitoring**: The developed model can be integrated into traffic monitoring systems, providing real-time data on vehicle counts, types, and movement patterns. This data can assist traffic authorities in making informed decisions to improve traffic flow and reduce congestion.
2. **Accident Detection and Emergency Response**: The model's real-time detection capabilities can be leveraged to identify accidents and traffic anomalies quickly. This rapid detection enables prompt responses from emergency services, potentially saving lives and reducing the impact of traffic disruptions.
3. **Optimization of Traffic Signals**: Integrating the model with traffic signal control systems can optimize traffic light timings based on real-time traffic conditions. This optimization can lead to more efficient traffic flow, reduced waiting times at intersections, and lower emissions from idling vehicles.
4. **Smart City Applications**: The research contributes to the broader vision of smart cities, where technology and data analytics are used to enhance urban living. The traffic vehicle detection system can be a critical component of smart city infrastructure, contributing to sustainable urban development.

## 6.5 Challenges and Limitations

Despite the promising results, the study faced several challenges and limitations that need to be addressed in future research:

1. **Varying Lighting Conditions**: The model's performance varied under different lighting conditions, such as bright sunlight, shadows, and nighttime. Future research should explore techniques to improve the model's robustness to lighting variations, such as data augmentation and the use of specialized image enhancement algorithms.
2. **Occlusions and Partial Visibility**: Vehicles partially occluded by other objects or infrastructure posed a challenge for accurate detection. Developing more sophisticated algorithms that can handle occlusions and improve detection accuracy in such scenarios is essential.
3. **Computational Resource Requirements**: Training and deploying deep learning models require significant computational resources, which may not be readily available in all settings. Future research should explore methods to reduce the computational load, such as model compression, quantization, and efficient hardware utilization.
4. **Scalability and Real-World Deployment**: While the model showed promising results in a controlled environment, scaling it for real-world deployment involves addressing issues related to hardware integration, data privacy, and system maintenance. Future work should focus on practical aspects of deploying and maintaining the system in real-world traffic management scenarios.

## 6.6 Conclusion

In conclusion, this research successfully demonstrates the application of deep learning for traffic vehicle detection in Dhaka City. The proposed CNN-based model achieves high accuracy and real-time performance, offering a valuable tool for improving urban traffic management. The study's findings contribute significantly to the field and provide a foundation for future research to build upon. By addressing the identified challenges and exploring the recommended areas for future work, it is possible to further enhance the effectiveness and applicability of deep learning models in traffic management systems.

This research marks a significant step towards smarter, more efficient traffic management solutions. The integration of deep learning with urban traffic systems has the potential to transform how cities manage their traffic, leading to safer, more efficient, and more sustainable urban environments. The journey towards achieving this vision involves continuous innovation, collaboration, and the application of cutting-edge technologies to address the evolving challenges of urban traffic management.

# CHAPTER 7

# DEMONSTRATION OF OUTCOME BASED EDUCATION

## 7.1 Introduction

Outcome-Based Education (OBE) is a modern educational approach that focuses on the desired outcomes of the educational process. It emphasizes what students can actually do after they are taught, rather than what they are expected to learn. In this context, our project on "Traffic Vehicle Detection of Dhaka City using Deep Learning Algorithm" is designed to address real-world problems by applying theoretical knowledge and practical skills. This chapter demonstrates how OBE principles have been integrated into our project to ensure a comprehensive learning experience and meaningful contributions to the field of electrical and electronic engineering.

The modern difficulty this project addresses is maximizing demand side management by means of load forecasting grounded on machine learning. The construction of a time-based scheme is the main focus of the functional requirements; however, a careful study of the feasibility and efficiency of combining machine learning-based load forecasting with an optimization algorithm is necessary to effectively satisfy energy consumption needs. The suggested approach defines the application of a time-based system including demand side management techniques and load forecasting and optimization models derived from machine learning. This strategic integration emphasizes the acceptance of creative engineering resources and tools, in line with modern methods in the sector. Together with budgetary concerns, a thorough management plan is important and carefully specifies the actions, tools, and cash distribution needed for the effective implementation of the suggested time-based scheme. The impact study guarantees that the suggested solution conforms with ethical and social criteria by including issues of health, safety, cultural sensitivity, and society consequences, so transcending technical elements.[26], [27] Integral components of the solution are environmental effect and sustainability; so, careful analysis of how the implementation influences the surrounding environment and its long-term sustainability is necessary. Effective cooperation is very necessary both personally and as a team member given the complexity of the problem to guarantee the successful application of the suggested solution. The last stage is creating thorough technical studies, thorough design documentation, and delivering strong presentations proving the effectiveness and possible influence of the solution.

## 7.2 Complex Engineering Problems

Engineering education must prepare students to solve complex problems that require a deep understanding of various engineering principles and the ability to apply them effectively. Our project addresses several complex engineering problems, which are outlined in the table below:

| **Knowledge Profile (Attribute)** | **Put Tick ( √ )** |
|---|---|
| A systematic, theory-based formulation of engineering fundamentals required in the engineering discipline | |
| Engineering specialist knowledge that provides theoretical frameworks and bodies of knowledge for the accepted practice areas in the engineering discipline; much is at the forefront of the discipline | √ |
| Knowledge that supports engineering design in a practice area | √ |
| Knowledge of engineering practice (technology) in the practice areas in the engineering discipline | √ |
| Comprehension of the role of engineering in society and identified issues in engineering practice in the discipline: ethics and the engineer's professional responsibility to public safety; the impacts of engineering activity; economic, social, cultural, environmental and sustainability | |
| Engagement with selected knowledge in the research literature of the discipline | √ |

This project requires a systematic theory-based formulation of engineering fundamentals. The deep learning models and image processing techniques employed are at the forefront of engineering practice, providing a solid theoretical framework. [27]Additionally, the project supports engineering design by requiring practical application of these theories. Understanding the role of engineering in society and addressing issues such as public safety, economic, social, cultural, environmental, and sustainability impacts are crucial. Furthermore, engagement with the latest research literature ensures that our project is grounded in current scientific knowledge.

## 7.3 Course Outcomes (Cos) Addressed

The initiative sought to solve many Course Outcomes (COs), thereby guaranteeing that the acquired information and abilities were thorough and in line with professional standards. The COs covered in our thesis are shown here.

| COs | CO Statement | POs | Put Ticks (√) |
|---|---|---|---|
| CO1 | Identify a contemporary real life problem related to electrical and electronic engineering by reviewing and analyzing existing research works | PO2 | √ |
| CO2 | Determine functional requirements of the problem considering feasibility and efficiency through analysis and synthesis of information. | PO4 | √ |
| CO3 | Select a suitable solution and determine its method considering professional ethics, codes and standards. | PO8 | √ |
| CO4 | Adopt modern engineering resources and tools for the solution of the problem. | PO5 | √ |
| CO5 | Prepare management plan and budgetary implications for the solution of the problem. | PO11 | √ |
| CO6 | Analyze the impact of the proposed solution on health, safety, culture and society. | PO6 | √ |
| CO7 | Analyze the impact of the proposed solution on environment and sustainability. | PO7 | √ |
| CO8 | Develop a viable solution considering health, safety, cultural, societal and environmental aspects. | PO3 | |
| CO9 | Work effectively as an individual and as a team member for the accomplishment of the solution. | PO9 | √ |

| | | | |
|---|---|---|---|
| CO10 | Prepare various technical reports, design documentation, and deliver effective presentations for demonstration of the solution. | PO10 | √ |
| CO11 | Recognize the need for continuing education and participation in professional societies and meetings. | PO12 | |

These COs ensure that our educational experience is comprehensive, covering identification and analysis of real-world problems, determining functional requirements, solution selection considering ethical standards, adoption of modern engineering tools, effective project management, assessment of health, safety, societal, and environmental impacts, effective communication, teamwork, and commitment to lifelong learning and ethical responsibilities.

## 7.4 Program Outcomes

| | Statement | Different Aspects | Put Tick ( √ ) |
|---|---|---|---|
| PO2 | Problem analysis: Identify, formulate, research literature and analyse complex electrical and electronic engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences. | | √ |
| PO4 | Investigation: Conduct investigations of complex electrical and electronic engineering problems using research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of information to provide valid conclusions. | Design of experiments | √ |
| | | Analysis and interpretation of data | √ |
| | | Synthesis of information | √ |
| PO6 | The engineer and society: Apply reasoning informed by contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent | Societal | |
| | | Health | |
| | | Safety | √ |

| | | | |
|---|---|---|---|
| | responsibilities relevant to professional engineering practice and solutions to complex electrical and electronic engineering problems. | Legal | |
| | | Cultural | |
| PO7 | Environment and sustainability: Understand and evaluate the sustainability and impact of professional engineering work in the solution of complex electrical and electronic engineering problems in societal and environmental contexts. | Societal | √ |
| | | Environmental | |
| PO8 | Ethics: Apply ethical principles embedded with religious values, professional ethics and responsibilities, and norms of electrical and electronic engineering practice. | Religious Values | |
| | | Professional ethics and responsibilities | √ |
| | | Norms | |
| PO9 | Individual work and teamwork: Function effectively as an individual, and as a member or leader in diverse teams and in multi-disciplinary settings. | Diverse teams | √ |
| | | Multi-disciplinary settings | |
| PO10 | Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions. | Comprehend and write effective reports | √ |
| | | Design documentation | √ |
| | | Make effective presentations | √ |
| | | Give and receive clear instructions | √ |
| PO11 | Project management and finance: Demonstrate knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments. | Engineering management principles | |
| | | Economic decision-making | √ |
| | | Manage projects | √ |
| | | Multidisciplinary environments | |

These POs ensure that our project is not only technically sound but also socially responsible, environmentally sustainable, and ethically conducted. They emphasize the importance of teamwork, effective communication, project management, and lifelong learning.

## 7.5 Attributes of Knowledge Profile

Our project addressed the following attributes of complex engineering problems:

| **Knowledge Profile (Attribute)** | **Put Tick ( √ )** |
|---|---|
| A systematic, theory-based formulation of engineering fundamentals required in the engineering discipline | |
| Engineering specialist knowledge that provides theoretical frameworks and bodies of knowledge for the accepted practice areas in the engineering discipline; much is at the forefront of the discipline | √ |
| Knowledge that supports engineering design in a practice area | √ |
| Knowledge of engineering practice (technology) in the practice areas in the engineering discipline | √ |
| Comprehension of the role of engineering in society and identified issues in engineering practice in the discipline: ethics and the engineer's professional responsibility to public safety; the impacts of engineering activity; economic, social, cultural, environmental and sustainability | |
| Engagement with selected knowledge in the research literature of the discipline | √ |

The complexity of these problems requires a holistic approach, considering technical, social, environmental, and ethical aspects. The depth of knowledge and analysis needed ensures that the solutions are robust and effective. Balancing conflicting requirements and understanding interdependencies highlights the complexity and interrelated nature of the problems addressed.

## 7.5 Attributes of Knowledge Profile

The project also addressed the following attributes of complex engineering activities:

| **Knowledge Profile (Attribute)** | **Put Tick ( √ )** |
|---|---|
| A systematic, theory-based formulation of engineering fundamentals required in the engineering discipline | |
| Engineering specialist knowledge that provides theoretical frameworks and bodies of knowledge for the accepted practice areas in the engineering discipline; much is at the forefront of the discipline | √ |
| Knowledge that supports engineering design in a practice area | √ |
| Knowledge of engineering practice (technology) in the practice areas in the engineering discipline | √ |
| Comprehension of the role of engineering in society and identified issues in engineering practice in the discipline: ethics and the engineer's professional responsibility to public safety; the impacts of engineering activity; economic, social, cultural, environmental and sustainability | |
| Engagement with selected knowledge in the research literature of the discipline | √ |

These attributes emphasize the need for a wide range of resources, significant stakeholder interaction, innovative thinking, and consideration of societal and environmental consequences. [26]Familiarity with standards and codes ensures that the solutions are compliant with industry practices and regulations.

## 7.6 Socio-Cultural, Environmental, and Ethical Impact

The project on traffic vehicle detection in Dhaka City has significant socio-cultural, environmental, and ethical impacts.

### 7.6.1 Socio-Cultural Impact

The project addresses the pressing issue of traffic congestion in Dhaka, which has profound socio-cultural implications. By improving traffic flow and reducing congestion, the project aims to enhance the quality of life for the city's residents. Reduced travel times can lead to more productive time spent with family and community, contributing to better social cohesion. Additionally, improved traffic management can decrease stress and road rage incidents, promoting a healthier social environment.[27]

### 7.6.2 Environmental Impact

Traffic congestion is a major contributor to air pollution in urban areas. By implementing an efficient vehicle detection system, the project can help manage traffic flow more effectively, reducing idle times and lowering emissions. This contributes to a cleaner environment and aligns with sustainable development goals. Moreover, the reduction in congestion can decrease fuel consumption, leading to energy savings and further environmental benefits.

### 7.6.3 Ethical Impact

Ethical considerations were integral to the project's development. The use of data for training the deep learning models adhered to privacy regulations and ensured that no personal information was misused. The project aimed to enhance public safety by providing accurate vehicle detection, thus preventing accidents and improving overall road safety. By adhering to professional ethics and standards, the project demonstrated a commitment to responsible engineering practice.

## 7.7 Demonstration of OBE Principles

The following sections detail the integration of OBE principles in our project implementation.

### 7.7.1 Identifying Contemporary Problems (CO1, PO2)

The project started by identifying the contemporary problem of traffic congestion in Dhaka City and analyzing its impact on urban life. This involved reviewing existing research and analyzing traffic data to understand the underlying issues. Traffic congestion in Dhaka is a significant problem, leading to economic losses, increased pollution, and reduced quality of life. By addressing this problem, we aim to contribute to the betterment of the city's infrastructure and improve the daily lives of its residents.

### 7.7.2 Determining Functional Requirements (CO2, PO4)

We determined the functional requirements for the traffic vehicle detection system, including accuracy, speed, and robustness to varying environmental conditions. This involved analyzing the feasibility and efficiency of different deep learning algorithms. The functional requirements were derived from a thorough analysis of the traffic conditions in Dhaka, the limitations of existing systems, and the capabilities of modern deep learning techniques. Ensuring high accuracy and real-time processing were critical for the system's effectiveness in managing traffic flow.

### 7.7.3 Solution Selection and Ethical Considerations (CO3, PO8)

We selected appropriate deep learning algorithms and considered ethical implications such as data privacy and the potential impact on public safety. This involved adhering to professional ethics, codes, and standards. The selection process included evaluating various algorithms for their accuracy, computational efficiency, and suitability for real-time processing. Ethical considerations were paramount, ensuring that the data used was collected and processed responsibly, and the system was designed to enhance public safety without infringing on individual privacy rights.

### 7.7.4 Adoption of Modern Engineering Tools (CO4, PO5)

We used modern engineering tools such as TensorFlow and OpenCV to develop and implement our solution. This involved creating, selecting, and applying appropriate techniques and resources to complex engineering activities. The use of these tools enabled us to leverage advanced machine learning and computer vision techniques, ensuring that our solution was state-of-the-art. The tools provided a robust framework for developing, testing, and deploying the deep learning models, facilitating efficient and accurate vehicle detection.

### 7.7.5 Project Management and Budgetary Planning (CO5, PO11)

Effective project management and budgetary planning were essential to ensure the timely and cost-effective completion of the project. This involved demonstrating knowledge and understanding of engineering and management principles and applying these to manage the project. We developed a detailed project plan, outlining the tasks, timelines, and resources required. Budgetary planning ensured that the project was completed within the allocated resources, optimizing the cost-effectiveness of the solution.[26]

### 7.7.6 Health, Safety, Societal, and Environmental Impact (CO6, CO7, PO6, PO7)

We assessed the health, safety, societal, and environmental impacts of our project, aiming to provide a solution that benefits the community and minimizes negative impacts. This involved understanding the impact of professional engineering solutions in societal and environmental contexts and demonstrating knowledge of and need for sustainable development. The system was designed to improve traffic management, thereby reducing congestion and pollution. By enhancing road safety and efficiency, the project aimed to positively impact public health and the environment.

### 7.7.7 Teamwork and Effective Communication (CO9, CO10, PO9, PO10)

Effective teamwork and communication were critical to the success of the project. We worked as a diverse team, communicating effectively through written reports and oral presentations. This involved functioning effectively as individuals and as members or leaders in diverse teams and in multidisciplinary settings. Team members brought diverse skills and perspectives, contributing to a comprehensive and well-rounded solution. Effective communication ensured that all team members were aligned with the project goals and could collaborate efficiently.

### 7.7.8 Commitment to Lifelong Learning (PO12)

Finally, the project demonstrated a commitment to lifelong learning by engaging in continuous research and development to stay updated with the latest advancements in deep learning and traffic management. The field of deep learning is rapidly evolving, with new techniques and technologies emerging regularly. By staying abreast of these developments, we ensured that our solution was cutting-edge and could adapt to future advancements. This commitment to lifelong learning is essential for professional growth and the continued success of engineering projects.

# REFERENCE


[1] J. Du, "Understanding of Object Detection Based on CNN Family and YOLO," *J Phys Conf Ser*, vol. 1004, p. 012029, Apr. 2018, doi: 10.1088/1742-6596/1004/1/012029.

[2] G. L. Foresti and L. Snidaro, "Vehicle Detection and Tracking for Traffic Monitoring," 2005, pp. 1198–1205. doi: 10.1007/11553595_147.

[3] S. Aqel, A. Hmimid, M. A. Sabri, and A. Aarab, "Road traffic: Vehicle detection and classification," in *2017 Intelligent Systems and Computer Vision (ISCV)*, IEEE, Apr. 2017, pp. 1–5. doi: 10.1109/ISACV.2017.8054969.

[4] J. Zhou, D. Gao, and D. Zhang, "Moving Vehicle Detection for Automatic Traffic Monitoring," *IEEE Trans Veh Technol*, vol. 56, no. 1, pp. 51–59, Jan. 2007, doi: 10.1109/TVT.2006.883735.

[5] X. Ji, Z. Wei, and Y. Feng, "Effective vehicle detection technique for traffic surveillance systems," *J Vis Commun Image Represent*, vol. 17, no. 3, pp. 647–658, Jun. 2006, doi: 10.1016/j.jvcir.2005.07.004.

[6] M. A. A. Al-qaness, A. A. Abbasi, H. Fan, R. A. Ibrahim, S. H. Alsamhi, and A. Hawbani, "An improved YOLO-based road traffic monitoring system," *Computing*, vol. 103, no. 2, pp. 211–230, Feb. 2021, doi: 10.1007/s00607-020-00869-8.

[7] Z. Chen, T. Ellis, and S. A. Velastin, "Vehicle detection, tracking and classification in urban traffic," in *2012 15th International IEEE Conference on Intelligent Transportation Systems*, IEEE, Sep. 2012, pp. 951–956. doi: 10.1109/ITSC.2012.6338852.

[8] Guolin Wang, Deyun Xiao, and J. Gu, "Review on vehicle detection based on video for traffic surveillance," in *2008 IEEE International Conference on Automation and Logistics*, IEEE, Sep. 2008, pp. 2961–2966. doi: 10.1109/ICAL.2008.4636684.

[9] Y. Tang, C. Zhang, R. Gu, P. Li, and B. Yang, "Vehicle detection and recognition for intelligent traffic surveillance system," *Multimed Tools Appl*, vol. 76, no. 4, pp. 5817–5832, Feb. 2017, doi: 10.1007/s11042-015-2520-x.

[10] J. Tao, H. Wang, X. Zhang, X. Li, and H. Yang, "An object detection system based on YOLO in traffic scene," in *2017 6th International Conference on Computer Science and Network Technology (ICCSNT)*, IEEE, Oct. 2017, pp. 315–319. doi: 10.1109/ICCSNT.2017.8343709.

[11] U. Mittal and P. Chawla, "Vehicle detection and traffic density estimation using ensemble of deep learning models," *Multimed Tools Appl*, vol. 82, no. 7, pp. 10397–10419, Mar. 2023, doi: 10.1007/s11042-022-13659-5.

[12] M. V, V. V.R, and N. A, "A Deep Learning RCNN Approach for Vehicle Recognition in Traffic Surveillance System," in *2019 International Conference on Communication and*

*Signal Processing (ICCSP)*, IEEE, Apr. 2019, pp. 0157–0160. doi: 10.1109/ICCSP.2019.8698018.

[13] H. Haritha and S. K. Thangavel, “A modified deep learning architecture for vehicle detection in traffic monitoring system,” *International Journal of Computers and Applications*, vol. 43, no. 9, pp. 968–977, Oct. 2021, doi: 10.1080/1206212X.2019.1662171.

[14] M. V. Peppa, D. Bell, T. Komar, and W. Xiao, “URBAN TRAFFIC FLOW ANALYSIS BASED ON DEEP LEARNING CAR DETECTION FROM CCTV IMAGE SERIES,” *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLII–4, pp. 499–506, Sep. 2018, doi: 10.5194/isprs-archives-XLII-4-499-2018.

[15] R. Carvalho Barbosa, M. Shoaib Ayub, R. Lopes Rosa, D. Zegarra Rodríguez, and L. Wuttisittikulkij, “Lightweight PVIDNet: A Priority Vehicles Detection Network Model Based on Deep Learning for Intelligent Traffic Lights,” *Sensors*, vol. 20, no. 21, p. 6218, Oct. 2020, doi: 10.3390/s20216218.

[16] A. Moradzadeh, H. Teimourzadeh, B. Mohammadi-Ivatloo, and K. Pourhossein, “Hybrid CNN-LSTM approaches for identification of type and locations of transmission line faults,” *International Journal of Electrical Power & Energy Systems*, vol. 135, p. 107563, Feb. 2022, doi: 10.1016/j.ijepes.2021.107563.

[17] C. M. Bautista, C. A. Dy, M. I. Manalac, R. A. Orbe, and M. Cordel, “Convolutional neural network for vehicle detection in low resolution traffic videos,” in *2016 IEEE Region 10 Symposium (TENSYMP)*, IEEE, May 2016, pp. 277–281. doi: 10.1109/TENCONSpring.2016.7519418.

[18] S. M. Sadakatul Bari, R. Islam, and S. R. Mardia, “Performance Evaluation of Convolution Neural Network Based Object Detection Model for Bangladeshi Traffic Vehicle Detection,” 2022, pp. 115–128. doi: 10.1007/978-981-16-6636-0_10.

[19] K.-J. Kim, P.-K. Kim, Y.-S. Chung, and D.-H. Choi, “Multi-Scale Detector for Accurate Vehicle Detection in Traffic Surveillance Data,” *IEEE Access*, vol. 7, pp. 78311–78319, 2019, doi: 10.1109/ACCESS.2019.2922479.

[20] A. Corovic, V. Ilic, S. Duric, M. Marijan, and B. Pavkovic, “The Real-Time Detection of Traffic Participants Using YOLO Algorithm,” in *2018 26th Telecommunications Forum (TELFOR)*, IEEE, Nov. 2018, pp. 1–4. doi: 10.1109/TELFOR.2018.8611986.

[21] A. Ghosh, Md. S. Sabuj, H. H. Sonet, S. Shatabda, and D. Md. Farid, “An Adaptive Video-based Vehicle Detection, Classification, Counting, and Speed-measurement System for Real-time Traffic Data Collection,” in *2019 IEEE Region 10 Symposium (TENSYMP)*, IEEE, Jun. 2019, pp. 541–546. doi: 10.1109/TENSYMP46218.2019.8971196.

[22] J. Zhu, X. Li, P. Jin, Q. Xu, Z. Sun, and X. Song, “MME-YOLO: Multi-Sensor Multi-Level Enhanced YOLO for Robust Vehicle Detection in Traffic Surveillance,” *Sensors*, vol. 21, no. 1, p. 27, Dec. 2020, doi: 10.3390/s21010027.

[23] F. J. M. , M. I. , R. A. S. , M. A. , T. Z. and N. N. I. Shamrat, “A smart automated system model for vehicles detection to maintain traffic by image processing.,” *International Journal of Scientific & Technology Research*, vol. 9(02), pp. 2921–2928, 2020.

[24] Y.-L. Chen, B.-F. Wu, H.-Y. Huang, and C.-J. Fan, “A Real-Time Vision System for Nighttime Vehicle Detection and Traffic Surveillance,” *IEEE Transactions on Industrial Electronics*, vol. 58, no. 5, pp. 2030–2044, May 2011, doi: 10.1109/TIE.2010.2055771.

[25] D. Mittal, A. Reddy, G. Ramadurai, K. Mitra, and B. Ravindran, “Training a deep learning architecture for vehicle detection using limited heterogeneous traffic data,” in *2018 10th International Conference on Communication Systems & Networks (COMSNETS)*, IEEE, Jan. 2018, pp. 589–294. doi: 10.1109/COMSNETS.2018.8328279.

[26] M. M. Syeed, A. Shihavuddin, M. F. Uddin, M. Hasan, and R. H. Khan, “Outcome Based Education (OBE): Defining the Process and Practice for Engineering Education,” *IEEE Access*, vol. 10, pp. 119170–119192, 2022, doi: 10.1109/ACCESS.2022.3219477.

[27] M. H. Davis *et al.*, “Case studies in outcome-based education,” *Med Teach*, vol. 29, no. 7, pp. 717–722, Jan. 2007, doi: 10.1080/01421590701691429.