

COMPARING THE PERFORMANCE OF DEEP LEARNING ALGORITHMS FOR VEHICLE
DETECTION AND CLASSIFICATION

A Thesis
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By
Elizabeth Arthur

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Civil, Construction, and Environmental Engineering

November 2022

Fargo, North Dakota

North Dakota State University
Graduate School

Title

COMPARING THE PERFORMANCE OF DEEP LEARNING
ALGORITHMS FOR VEHICLE DETECTION AND CLASSIFICATION

By

Elizabeth Arthur

The Supervisory Committee certifies that this *disquisition* complies with North Dakota
State University's regulations and meets the accepted standards for the Degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Dr. Ying Huang

Chair

Dr. Pan Lu

Dr. Simon Ludwig

Approved:

11/17/2022

Date

Dr. Xuefeng Chu

Department Chair

ABSTRACT

The rapid pace of developments in Artificial Intelligence (AI) provides unprecedented opportunities to enhance the performance of Intelligent Transportation Systems. Automating vehicle detection and classification using computer vision methods can complement traditional sensors or serve as a cost-effective and environmentally friendly substitute for conventional sensors. This study investigates the robustness of existing deep learning models for vehicle identification and classification using a heterogenous dataset. The dataset is grouped into six distinct classes based on the Federal Highway Administration (FHWA) vehicle classification scheme. This study uses three different versions of You Only Look Once (YOLO) single-stage object detection models, namely YOLOv7, YOLOv5m, and YOLOv5s. The comparative evaluation will depend on four performance metrics: recall, precision, F1-score and mean average precision (MAP). The results show that for this case study, YOLOv7 outperformed the other models with 84.7% precision, 89.4% recall, 86.1% F1-score and 93% MAP at 0.5, and 82.4% MAP at 0.95.

ACKNOWLEDGMENTS

Firstly, I would like to express my sincere gratitude to my advisor Dr. Ying Huang for the continuous support of my graduate study and related research, for their patience, motivation, and immense knowledge. Her guidance helped me in all the time of research and writing of this thesis.

Besides my advisor, I would like to thank the rest of my thesis committee: Dr. Simone Ludwig, and Dr. Pan Lu, for their insightful comments.

I also would like to acknowledge the financial support from National Science Foundation under award number OIA-2119691 to this study.

My final thanks go to the Department of the Civil, Construction, and Environmental Engineering Department at North Dakota State University for the resources they provided me to get through my graduate education. I am most grateful.

DEDICATION

I dedicate this thesis to my fiancé and lovely family, who have been integral to my education through their constant support and love.

TABLE OF CONTENTS

ABSTRACT.....	iii
ACKNOWLEDGMENTS	iv
DEDICATION.....	v
LIST OF TABLES.....	viii
LIST OF FIGURES	ix
LIST OF APPENDIX FIGURES.....	x
1. INTRODUCTION	1
1.1. Research Background.....	1
1.2. FHWA Vehicle Classification Scheme	1
1.3. Conventional Vehicle Detection and Classification Methods.....	4
1.4. Vehicle Detection and Classification Using Deep Learning.....	6
1.4.1. Dataset Used for Object Detection and Classification.....	6
1.4.2. Object Detection and Classification Models Performances.....	11
1.5. Problem Statement and Significance of This Study.....	15
1.6. Objectives and Organization of Thesis	16
2. DEEP LEARNING MODELS.....	18
2.1. Object Detection and Classification.....	18
2.2. YOLOv5.....	19
2.3. YOLOv7.....	21
2.4. Summary	25
3. METHODOLOGY	26
3.1. Data Collection.....	26
3.2. Vehicle Classes	28
3.3. Vehicle Annotation	30

3.4.	Data Preprocessing and Augmentation	32
3.5.	Model Training.....	33
3.6.	Summary	35
4.	RESULTS AND DISCUSSION.....	36
4.1.	Evaluation Metrics	36
4.2.	Model Training and Validation	38
4.3.	Model Testing Results.....	39
4.4.	Summary	46
5.	CONCLUSIONS AND FUTURE WORK.....	48
	REFERENCES	51
	APPENDIX.....	55

LIST OF TABLES

<u>Table</u>	<u>Page</u>
1. FHWA vehicle classification definitions [5]	3
2. Specification of YOLOv5 model variants [23].....	21
3. Number of frames for each time of day and weather condition	27
4. Percentages of different resolutions.....	28
5. Suggested vehicle classification scheme	29
6. Number of vehicle instances for each class	31
7. The hyperparameter values used in training	34
8. Confusion matrix diagram for binary classification	36
9. Summary of performance metrics of model testing.....	45

LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
1. FHWA 13-category scheme for vehicle classifications [4]	2
2. Generalized YOLOv5 network architecture [23]	20
3. Different variants of the YOLOv5 model [23]	21
4. Extended efficient layer aggregation networks [26]	22
5. Model scaling for concatenation-based models [26]	23
6. Planned re-parameterized model [26]	24
7. Coarse for auxiliary and fine for lead head label assigner [26]	24
8. Vehicles captured in sunny, rainy, snowy, and night conditions.....	27
9. Frequency distribution of the number of frames for each condition	27
10. Vehicle classes from different angles	29
11. Vehicle annotation using CVAT.....	30
12. Frequency distribution of the number of class instances	31
13. Some annotation challenges encountered	32
14. Data augmentation methods applied to training data using roboflow	33
15. Results from YOLOv7 model training	38
16. Results from YOLOv5s model training.....	38
17. Results from YOLOv5m training	39
18. Confusion matrices for object detection models.....	39
19. Example of YOLOv5s misclassifying class 1 as a background	40
20. Example of YOLOv5s predicting a class 3 as class 2	41
21. YOLOv5m misclassifying the background as class 2	42
22. False positives due to the YOLOv7 detecting unlabeled vehicles.....	43

LIST OF APPENDIX FIGURES

<u>Figure</u>	<u>Page</u>
A1.YOLOv7 vehicle detections.....	55
A2.YOLOv5s vehicle detection.....	56
A3.YOLOv5m vehicle detections.....	57

1. INTRODUCTION

1.1. Research Background

Vehicle classification is the process of dividing all detected vehicles into specific predefined classes [1]. In transportation engineering, the accuracy of vehicle classification data is paramount for appropriate future highway design, including determining pavement characteristics, eradicating traffic jams, and enhancing safety. Local and national governments use vehicle classification data to make informed decisions about mobility, infrastructure, and taxation. Vehicle classification poses a complex problem as some vehicle classes have high intra-class contrasts and relatively low inter-class variations [2]. Organizations depending on vehicle classifiers for data collection should be mindful that systems are sometimes impacted by hardware and sensor malfunction and the device's implementation of the classification scheme [3]. Automating vision-based vehicle detection and classification using convolutional neural networks (CNNs) could be a more effective alternative to sensors. Computer vision is not limited to vehicle detection and classification but also vehicle tracking and traffic anomaly detection, which can all be leveraged for transportation engineering.

1.2. FHWA Vehicle Classification Scheme

Figure 1 illustrates the standardized vehicle classification system developed by the Federal Highway Administration (FHWA) in the mid-1980s. This 13-category system resulted from compromises invented to satisfy the needs of many traffic data users and to meet the requirement for the electronic equipment and sensors available at the time (primarily simple road tubes) to distinguish passing vehicles into the predefined classifications. Traffic data users such as pavement designers and the safety community were very interested in the portion of travel occurring in multi-unit vehicles (power units of various types pulling trailers of multiple configurations). At the time, available sensors could measure the presence of vehicles, detect axles, and determine the distance between consecutive axles based on every vehicle's speed as it moves over the sensors [4].


















FHWA Vehicle Classifications			
1. Motorcycles 2 axles, 2 or 3 tires 	2. Passenger Cars 2 axles, can have 1- or 2-axle trailers 	3. Pickups, Panels, Vans 2 axles, 4-tire single units Can have 1 or 2 axle trailers 	4. Buses 2 or 3 axles, full length 
5. Single Unit 2-Axle Trucks 2 axles, 6 tires (dual rear tires), single-unit 	6. Single Unit 3-Axle Trucks 3 axles, single unit 	7. Single Unit 4 or More-Axle Trucks 4 or more axles, single unit 	8. Single Trailer 3- or 4-Axle Trucks 3 or 4 axles, single trailer 
9. Single Trailer 5-Axle Trucks 5 axles, single trailer 	10. Single Trailer 6 or More-Axle Trucks 6 or more axles, single trailer 	  	
11. Multi-Trailer 5 or Less-Axle Trucks 5 or less axles, multiple trailers 	12. Multi-Trailer 6-Axle Trucks 6 axles, multiple trailers  	13. Multi-Trailer 7 or More-Axle Trucks 7 or more axles, multiple trailers 	

Figure 1. FHWA 13-category scheme for vehicle classifications [4]

The FHWA classification definitions shown in Table 1. are presently used for most Federal reporting requirements and serve as the foundation for most state vehicle classification counting efforts [5].

Table 1. FHWA vehicle classification definitions [5]

Class	Class Definition	Class Includes	Number of Axles
1	Motorcycles	Motorcycles	2
2	Passenger Cars	All cars. Cars with one-axle trailers. Cars with two-axle trailers	2, 3, or 4
3	Other Two-Axle Four-Tire Single-Unit Vehicles	Pickups and vans. Pickups and vans with one- and two- axle trailers	2, 3, or 4
4	Buses	Two- and three-axle buses	2 or 3
5	Two-Axle, Six-Tire, Single-Unit Trucks	Two-axle trucks	2
6	Three-Axle Single-Unit Trucks	Three-axle trucks. Three-axle tractors without trailers	3
7	Four or More Axle Single-Unit Trucks	Four-, five-, six- and seven-axle single-unit trucks	4 or more
8	Four or Fewer Axle Single-Trailer Trucks	Two-axle trucks pulling one- and two-axle trailers. Two-axle tractors pulling one- and two-axle trailers. Three-axle tractors pulling one-axle trailers	3 or 4
9	Five-Axle Single-Trailer Trucks	Two-axle tractors pulling three-axle trailers. Three-axle tractors pulling two-axle trailers. Three-axle trucks pulling two-axle trailers	5
10	Six or More Axle Single-Trailer Trucks	Multiple configurations	6 or more
11	Five or Fewer Axle Multi-Trailer Trucks	Multiple configurations	4 or 5
12	Six-Axle Multi-Trailer Trucks	Multiple configurations	6
13	Seven or More Axle Multi-Trailer Trucks	Multiple configurations	7 or more
14	Unused	----	----
15	Unclassified Vehicle	Multiple configurations	2 or more

Researchers have identified some shortcomings of the FHWA classification system. The FHWA definitions are based on vehicle features that can be effortlessly identified visually but cannot be ideally computed based on the number, weight, and spacing of axles. Truck attributes may alter

considerably from state to state because vehicle owners and manufacturers create and optimize vehicles to increase their profits, which relies on each state's truck size and weight laws. The FHWA chart in Figure 1 categorizes pickup trucks as class 2, class 3, and class 5. Let us consider three pickup trucks with the same number of axles and axle spacing. The pickup truck will belong to class 3 if it has a conventional (two-tire) rear axle, whereas it will be classified as class 5 if it has dual tires on the sides of its four-tire rear axle [5]. In this example, the weight in motion (WIM)-based classification that includes axle weights as a classification parameter can accurately differentiate traditional pickups from larger pickup trucks because the truck weight configuration will vary based on the heavier engine. Nonetheless, traditional vehicle classifiers that do not have access to axle weight information cannot differentiate between those two vehicles. Traditional vehicle classifiers may place both conventional pickups and trucks in the same vehicle category, resulting in one of these trucks being correctly classified and the other being misclassified [5].

1.3. Conventional Vehicle Detection and Classification Methods

Conventionally, vehicle detection and classification can be conducted through three approaches, including manual counting, in-pavement sensors, or roadside sensors. The manual traffic count is the most straightforward of classifying vehicles. In this method, trained observers use tally sheets or mechanical counters to record visually observed vehicles belonging to a particular class at specific locations and times. The manual approach can be conducted onsite or offsite, but it is time-consuming and subject to human errors.

To overcome the disadvantages of the manual counting method, in-pavement sensors became available and popular since 1920 including the pneumatic tube, magnetic loop detector, and piezoelectric sensors. The pneumatic tube detectors were introduced, and presently they collect vehicular data for a short period. A pneumatic tube can identify the number of axles and the axle spacing of a moving vehicle. This method is not appropriate for high-volume and high-speed

roadways [6]. A magnetic loop detector is a device utilized for vehicle classification in recent decades. A magnetic loop detector is a vehicle classification technology that detects vehicle length. It also measures vehicle speed using dual-loop sensors. Though magnetic loop detectors are inexpensive and execute automatic classification, they do not do well in high congestion [7]. Piezoelectric sensors are used alone or in combination with Weight-In-Motion (WIM) systems to detect the vehicle weight and the axle configuration. The disadvantage of piezoelectric sensors is their sensitivity to vehicle speed and pavement temperature. Weigh-in-Motion (WIM) technology was first conceptualized over 50 years ago. The purpose of a WIM system is to measure the dynamic axle load of moving vehicles to calculate their static axle weight and gross vehicle weights as they pass over a measurement site [8]. The WIM architecture consists of modeling and estimation components [9]. A network of in-pavement sensors, a data collecting facility, and an algorithm or framework for WIM data extraction make up an efficient WIM system. WIM system is categorized depending on the operating vehicle speed; the low-speed weigh-in-motion (LS-WIM) is for passing speeds up to 25 mph, and the high-speed weigh-in-motion (HS-WIM) is for passing speeds up to 80 mph [8]. The WIM system is also expensive and not feasible for local roads [9].

Meanwhile, roadside sensors such as radar, infrared, and video sensors also have been developed for easier sensor management and maintenance. Radar sensors are well-known for classifying vehicles using dimensions such as length, size, and height [10]. The radar sensors are less sensitive to environmental variation than other methods but are inappropriate for dense traffic congestion. Acoustic sensors use speed-independent acoustic signatures to recognize vehicle classes [11]. Infrared sensors estimate the reflected infrared light by each vehicle and compare the data with the database to locate the best-matched profile. Infrared sensors are susceptible to environmental factors. In addition, in recent decades, cameras have become affordable and effectively for vehicle counting and classification using video detections with machine learning techniques. Like other

roadside sensors, the detection accuracy of the video-based methods is also susceptible to weather and environmental factors.

1.4. Vehicle Detection and Classification Using Deep Learning

1.4.1. Dataset Used for Object Detection and Classification

The progress of image processing, pattern recognition, and vehicle-type classification technology based on deep learning has grown in popularity, with most research focusing mainly on vision-based methods. These approaches usually use image processing techniques to detect and classify vehicles using several steps, such as data preprocessing, feature extraction and selection, and classification. The cameras for data collection can be surveillance video systems, omnidirectional cameras, aerial images, Closed-Circuit Television (CCTV), or regular cameras [12]. Deep Learning algorithms are data-hungry and they depend on the high quantity and quality of data to build robust and accurate models. The dataset is a crucial input in deep learning-based classification systems, enabling the algorithms to extract the features and make predictions based on the learned information. Data collection primarily involves data acquisition, image annotation, and improvement of existing data or models [13]. The dataset for the object detection and classification can be obtained using manual recording, fix cameras, or an unmanned aerial vehicle (UAV).

Using manual recording, Espinosa et al. [14] recorded a video sequence of a two-lane road in an urban area of Medellín, Colombia. The vehicles were grouped into four classes: cars (including sedans, vans, and taxis), motorbikes, buses, and trucks. Any vehicle that does not belong to any indicated group was classified as "unknown." The authors made the video recording during daytime and favorable weather conditions and consists of 1812 RGB frames of size 640 * 480. Regmi et al. [15] generated a unique dataset containing a substantial number of mainframes by positioning a high-resolution (2304*1296) CCTV camera toward a road to capture vehicles at 24fps. The video captures keyframes of severe morning fog traffic, dense peak daytime traffic, and nighttime traffic.

According to Akdag et al. [16], there is no publicly accessible large-scale dataset for vital vehicles. Significant obstacles include the lack of emergency vehicles and the varying coloring norms among nations. They combine photos acquired from multiple sources with the aid of the YOLO vehicle recognition model to create a large-scale crucial vehicle dataset in order to deal with the limitations mentioned above. Fire trucks, police cars, ambulances, military police cars, hazardous trucks, and ordinary vehicles are the different classifications of the created dataset.

On the other hand, oblique photography involves moving the camera along the axis to the vertical. Because of the training set, the YOLO algorithm does not produce results in photographs collected from the nadir. By collecting data from two significant Indonesian toll roads, the Jagorawi Toll and the Kapuk Toll, Arinaldi et al. [17] created their dataset named the Indonesian Toll Road dataset. They filmed the video for the Jagorawi toll data at Ramp 2 Taman Mini Indonesia Indah (TMII) Jasa Marga, Indonesia. Manually utilizing a camera, the dataset was collected from a pedestrian bridge. The authors captured the video at the Kapuk Toll Gate for the Kapuk Toll Road. The dataset is available with a resolution of 4096x2160 and a frame rate of 22.0.

Through fixed roadside cameras, Abdullah Anwer Mardin et al. [18] established two datasets. The first dataset is called general accident images (GAI), and it contains pictures from internet google images, DCD, and accident datasets. The second dataset, Kurdish accident photos (KAI), contains pictures directly acquired from the interior ministry of the regional government of Iraqi Kurdistan (KRG). These images depict accidents between 2015 and 2020 on routes like motorways and crossroads. The raw dataset has 100 videos and 2000 photos of the accidents. To create pictures of the accident, they took screenshots of particular sections of the footage. After obtaining the datasets, data preprocessing and data labeling starts. The GAI dataset is divided into six categories: cars, cars and buses, cars and motorcycles, buses and trucks, and cars and trucks. To keep the balance, each class receives 118 images, totaling 708.

The challenge of detecting vehicles is complicated by lighting changes, objects' backgrounds, time of day, occlusions, blur, motion, and camera quality. According to the Federal Highway Administration, Faruque et al. [19] divide vehicles into bicycles, trucks, vans, buses, and trailers (FHWA). The images are all collected from the New Jersey Department of Transportation (NJDOT) traffic video sequences. Note that these videos are captured from similar camera angles during day time. Due to the camera viewing angles, lighting fluctuations, and weather circumstances, there are many complex problems in classifying the vehicles in the footage. There can be more visual instances, making the vehicle detection issue more complex.

Amund Hansen Vedal et al. [20] searched for sources of annotated car images to build a large dataset rapidly. They wanted to build their dataset rather than a pre-made dataset like CompCars, but they recognized how tiring it would be to use Google Image searches. They created a database of 1000 labeled pictures per class –similar to the intent of the ImageNet dataset. Firstly, they downloaded 7217 car model pictures from PlatesMania, a free database of labeled vehicle pictures, and hand-sorted them into 28 distinct classes. The authors selected some classes based on related studies, internet sources, or when needed to minimize the "Other"- class (like concrete mixer, Military, and Crane). They downloaded 296,000+ images of the vehicle models they had already classified throughout our experiments, automatically mapping them to our classes. The images are taken in different environments, not always encompassing the whole vehicle, and have a reasonable resolution (~ 1200x900 pixels). The watermarks, people, buildings, and trees in some images can serve as noise to their dataset, which the network must learn to recognize as irrelevant. The authors claim to have produced the largest vehicle dataset, containing over 207,000 unique images naming it the "PlatesMania dataset."

Muhammad Atif Butt et al. [21] assert that no standardized public vehicle dataset is available that includes pictures of typical automobiles to address classification tasks. For instance, the real-

time classification systems cannot be implemented using the CompCars and Stanford vehicle datasets since they only contain the classes of modern cars from a few regions. They provide a new vehicle dataset to address this problem, which consists of 10,000 photos divided into six classes based on standard road traffic vehicles, and each class has 1670 images. The dataset is formed through manual labeling using the windows editing tool. Data augmentation is a prevalent technique to reduce overfitting from the network by artificially inflating the dataset through label-preserving transformation strategies to raise the diversity of our dataset. The authors used four distinct types of data augmentation: Gaussian blur, rotation, horizontal flip, and Gaussian noise.

Sumeyye et al. [22] used video with a resolution of 1280x720 from an unmanned aerial vehicle (UAV) and 1080x1920 from a terrestrial source. Each of the two films is around a minute long and has a frame rate of 24 fps. There was no preprocessing done before using the videos. Instead of being created at nadir, the footage taken by the UAV was produced obliquely. Nadir photography, which takes pictures with the camera axis vertically below the subject, is the norm for UAV surveys. In 2018, Uzar M. et al. [23] selected parking lots located at Technical University in Istanbul, Turkey, as data collection sites to garner aerial images using the Unmanned Aerial Vehicle (UAV) system. They obtained Ninety-four images with a size of 5472 * 3648 pixels and a resolution of 72 dpi. Most studies apply data augmentation techniques to the dataset to contribute to the representative ability of the dataset. The authors used three primary data augmentation methods for their research: brightness analysis, shear, and rotation. Brightness analysis is to change the image's brightness to become darker or lighter than the original image. Shear transformation is to fix one axis and stretch the image at a certain angle, known as the shear angle. Image rotation enables the image to rotate along the positive or negative axis. Cars, minibuses, and buses are the three categories for vehicle classification. Visual Object Tagging Tool (VoTT) is the software used for data annotation.

Data annotation involves labeling data (photos and video) to teach supervised deep learning models to comprehend input patterns by interpreting them and generating reliable results. Image annotation is executed using different methods that produce different results and image labels. The annotation type depends on the use case and the application domain. The different annotation types are image classification, object detection, and segmentation. Image classification simply assigns the entire image one label to show the presence of the target object. Object detection combines image classification and localization to find the presence, location, and the number of target objects in an image. Image segmentation is a sophisticated data annotation type that separates the image's objects into segments. There are three types of image segmentation: semantic segmentation, instance segmentation, and panoptic segmentation. Semantic segmentation (class segmentation) differentiates between objects by assigning one label to every object belonging to that specific class. Instance segmentation detects and segments each object instance of a particular class in an image. Panoptic segmentation unifies semantic segmentation and instance segmentation.

The data annotation technique is the shape used to select the object of interest when labeling the data. The bounding box method draws rectangular boxes to define the target object's location within an image. It is the most basic data annotation technique due to its simplicity and versatility. It is beneficial when objects are symmetrical and when the precise shape of an object is of little interest. Bounding boxes can be either two-dimensional (2D) or three-dimensional (3D). The 3D cuboid annotation draws a 2D box around the object and considers the depth factor. Polygonal segmentation is a variation of the bounding box technique where polygons define the target object's location and boundaries more accurately. Polygons are suitable for irregularly shaped objects because they eliminate irrelevant pixels that can confuse the model. The polyline technique plots continuous lines and splines to delineate boundaries within an image. It is suitable for linear target objects. Nevertheless, it only works for some use cases because most objects are not linear and require more

than one pixel width. Landmarking (dot annotation) involves connecting dots to represent the outline or skeleton of an object in the image. These dots help detect and quantify the features of an object. This approach is time-consuming and prone to inaccuracy.

1.4.2. Object Detection and Classification Models Performances

Object detection can be challenging because the objects in images often have different sizes, orientations, and overlapping objects resulting in the occlusion of the object of interest. Vehicle detection methods have been developing for several years in academia and industry. So far, some state-of-the-art object detection methods cannot achieve competitive performance on vehicle detection benchmarks. The main issues for vehicle detection are a considerable variation of light, dense occlusion, and significant variation of object scales. Espinosa et al. [14] compared two deep-learning models for vehicle detection using Alex Net and Faster R-CNN. Several tests were executed to evaluate the quality of detections, failure rates, and time to finish the task. The results gave essential conclusions regarding the architectures and strategies for implementing such a network for video detection. Faster R-CNN evaluations are executed based on the Non-max Suppression (NMS) parameter threshold, used to lessen the redundancy of proposed regions. The best results correspond to an NMS threshold of 0.6 to an F1-score of 0.76. The findings demonstrate that lowering the Intersection over Union (IoU) threshold criteria raises the overall and correct detection rates for each class under study while also raising false alarm rates. Meanwhile, working with the AlexNet classifier with the Gaussian Mixture Model (GMM) background subtraction, the best results are for a history of 500 frames ($F1 = 0.57$). The results indicate that Faster R-CNN surpasses the AlexNet + GMM model in the correct detection rate obtained while producing fewer false detections. For the time spent in the analysis, the Faster R-CNN model was closer to real-time with (40 ms per frame) while AlexNet + GMM took almost 100 ms per frame.

Regmi et al. [15] initially compared the vehicle detection model's accuracy and speed. In both comparisons, YOLOv3 models outperform the Mask RCNN model with a slight accuracy tradeoff for light vehicle recognition. The YOLOv5 model, though, functions better than both YOLOv3 and Mask RCNN in terms of accuracy. According to the experimental findings, the YOLO V5 is the best option for vehicle detection. There is a need to improve the accuracy of nighttime vehicle recognition because all of the models' vehicle detection accuracy is relatively low compared to daytime. The authors believe there is a significant research gap in enhancing the accuracy of motorcycle identification, as evidenced by the deficient performance of the models during motorcycle detection. Faruque et al. [19] used traffic videos from the New Jersey Department of Transportation (NJDOT) for the training data sets. The Faster R-CNN and YOLO deep learning methods perform differently when using different training data sets regarding training time, testing time, vehicle classification accuracy, and generalization performance. The experiments show the feasibility of vehicle classification in videos using deep learning methods and reveal that the YOLO deep learning method is much faster than the Faster R-CNN deep learning method.

Pre-training and fine-tuning are frequently used to leverage the limited number of vehicle images and improve classification performance. Akdag et al. [16] assess the performance of the three models, namely EfficientNet, Vision Transformer (ViT)- base, and ResNet-50, by deploying the following performance metrics: recall, accuracy, and F1-score. The study revealed that the vision transformer (ViT) model's average accuracy was 99.39%, while EfficientNet and ResNet-50 were 98.44% and 98.27%, respectively. Although the convolutional neural network (CNN) models have better results in a few class types, the ViT model achieves the best for all metrics on average. It was observed that the models could quickly pinpoint the ambulance and firetruck classes thanks to their distinctive and prominent salient colors and stripes. The military police cars achieve the lowest F1-score of 94.79% in EfficientNet and 96.29% in ResNet-50. With 97.24% in EfficientNet and 96.66%

in ResNet-50, the class of police cars has the second-lowest accuracy rating. Akdag et al. [16] compared the latency for each classification model. EfficientNet and ResNet-50 took 0.015 seconds and 0.013 seconds, respectively, to classify one image, compared to 0.017 seconds for the ViT model. The ViT model achieves more significant latency, which is acceptable, despite exceeding other models for the crucial vehicle classification.

Sumeyye et al. [22] applied high-resolution aerial or remote sensing images to extract data using the YOLO-v3, YOLO-v3-spp, and YOLO-v3-tiny models. The study's findings showed that the Yolov3-spp approach produced the best results, with an average IoU of 84,88% and a precision value of 72,02%. The IoU value is less than 0.5. Thus, even if the vehicle is correctly identified in the aerial video, it is not considered an accurate object. The approach that performed the best on the COCO is the YOLOv3-spp model, with an on-the-ground accuracy of 84.88%. With 88.56% for terrestrial video and 81.21% for UAV, the average IoU was attained. The YOLOv3-tiny method fails to detect small objects. Since the UAV video was captured from a distance, YOLOv3-tiny could not identify the object. As a result, while comparing the accuracy and true ground value of the video captured by UAV, the YOLOv3-tiny approach produced no results. The accuracy findings for the YOLOv3-spp model are better, with a precision of 72.02%; the accuracy for the UAV was 63.53%, and the accuracy for the model was 80.49%. In terms of model estimation and accuracy factor, both model comparisons produced results, and it was discovered that the model with the higher performance on terrestrial videos was more appropriate. The vehicles in UAV images can be successfully identified with the data set appropriate for the specified investigation. With the data set to be employed, more accurate detections can be made concurrently with the goal of the targeted study. Training and verification processes are crucial in the data set appropriate for the input data in order to raise the accuracy factor. The data set employed is better suited for terrestrial photos, which

is the leading cause of the low accuracy in the aerial footage. However, success rates will boost trust in deep learning techniques with more thorough training.

Ahmad Arinaldi et al. [17] evaluate the classification accuracy outcomes of five-fold cross-validation between the SVM-based models and the Faster RCNN model for the classification of vehicle kinds. They demonstrate that, in terms of cross-validation accuracy, Faster RCNN beats both SVM-based classification models for both the Indonesian Toll Road dataset and the MIT Traffic dataset. Another intriguing finding is that the accuracy ranges in the Indonesian Toll Road dataset differ significantly from those in the MIT Traffic dataset. They postulate that this is caused by the Indonesian Toll Road dataset's more pronounced variability, which includes two separate sites and a range of illumination conditions at different times (day and night). In comparison, there is just one scenario with homogeneous lighting in the MIT Traffic dataset. Abdullah Anwer Mardin et al. [18] assert that convolutional neural networks (CNNs) can achieve accurate classification and detection results. Five deep learning models, GoogleNet, ResNet50, MobileNetV2, AlexNet, and SqueesNet, have been compared for vehicle classification. The chosen networks have provided different classification rates (MobileNetV2 and ResNet50 achieved 4% more than the remaining networks). The research discoveries revealed that using detector networks with deep CNN topologies boosts the accuracy of accident vehicle classification by finding the location and vehicle class in the accident image. Though the YOLOv2's training time was extensively less than Faster RCNN, the study results reveal that pre-trained Resenet50, used as a feature extractor with the YOLOv2 model, achieves relatively vehicle detection results (more than 6%) in both datasets. Regardless, these outcomes fall short of anticipations as the research aims to discover a network or method to describe the accident images and detect the vehicle classes accurately. Thus, deep semantic segmentation is under research to help precisely classify vehicle classes.

Uzar M. et al. [23] implemented the nine YOLO models, namely YOLOv5s-CSP, YOLOv5s-tiny, YOLOv5s-P5, YOLOv5s-P6, YOLOv5s, YOLOv5l, YOLOv5m, YOLOv5n, YOLOv5x for performance analysis of automatic vehicle detection using UAV-based aerial images. The YOLOv5s-tiny model gives the highest F1-score of 0.89. YOLOv5 models give similar results, but YOLOv5s-CSP, YOLOv5s-P5, and YOLOv5s-P6 models show relatively lower F1-scores than the other architectures. The YOLOv5 models have higher MAP values than the YOLOv5s models. The YOLOv5m provides the MAP of 84% as the highest value. For a model to perform real-time vehicle detection, the fps values of the models must be examined. YOLOv5s-tiny and YOLOv5n models with a processing speed of 63 fps are the fastest ones developed for low-performance systems. The slowest models are YOLOv5s-P5, YOLOv5s-P6, and YOLOv5x, developed for high-performance systems. All of the YOLOv5 models provided the highest MAP values. Thus, YOLOv5 MAP results for each class show that cars are the most accurately predicted class. However, minibuses and buses reduce the overall accuracy of the model.

1.5. Problem Statement and Significance of This Study

The above literature review shows that traditional vehicle classification methods, though having many advantages, tend to have some drawbacks. The manual counts require more than one counter to achieve maximum accuracy. Manual counting is also incredibly ineffective and is vulnerable to human error while putting the counters' safety at risk. The permanent traffic site counting devices are relatively expensive and cause damage to the surrounding environment. They also have a limited design life and provide moderate accuracy.

Recently, research on vision-based vehicle detection and classification methods has been rising because of the progress of image processing, pattern recognition, and vehicle-type classification technology based on deep learning. Vision-based vehicle detection and classification face many complex problems due to camera viewing angles, lighting fluctuations, deplorable

weather, dense occlusion, and significant variation of object scales. Also, traffic data can contain many visual instances of objects that are not of interest, which serve as noise that the models must learn to recognize as irrelevant.

As there is a need to leverage deep learning to improve vision-based vehicle classification methods, it is worth exploring the performance of deep learning algorithms for vehicle detection and classification since they can be a cost-effective, environmentally friendly, and safer means of collecting, analyzing, and reporting vehicle classification data. Though not perfect, deep learning models are well known for providing very high performances for detection and classification tasks, and studies to explore these models' performance on vehicular data are of interest. There is a need to assess the performance of state-of-the-art deep learning models for vehicle classification in different scenarios to assure users of their effectiveness, strengths, and weaknesses.

1.6. Objectives and Organization of Thesis

This thesis investigates the performance of different state-of-the-art deep learning models for vehicle detection and classification and compares the model's effectiveness using some performance metrics. Each model will generate varying results; therefore, we must determine which algorithm is most suitable for the use cases and datasets. The research undertaken has the following objectives:

- 1) First, the study collects unique datasets that other studies have not exhaustively investigated.

This requires amassing massive data samples based on varying weather conditions, time of day, resolution, and camera positions. The images were annotated further to serve as ground truth for the various object detection models. The annotations were done using the computer vision annotation tool (CVAT).

- 2) Second, the study conducts a comparative analysis of various object detection models. Three cutting-edge single-stage object detection models were considered to accomplish this goal:

YOLOv5m, YOLOv7, and YOLOv5s. Each model was trained and tested on thousands of

heterogeneous datasets. The purpose of the heterogeneous dataset is to increase the variance of the training dataset. The models were then tested for precision, recall, F1-score, and mean average precision (MAP).

The organization of the rest of this thesis document is as follows. Chapter two describes the state-of-the-art deep learning models being used for the research. The methodology for this study is presented in Chapter three. Chapter four presents a discussion of the experimental results. Finally, Chapter five summarizes the research, the conclusions drawn from the results, and recommendations for future research.

2. DEEP LEARNING MODELS

2.1. Object Detection and Classification

The application of convolutional neural networks (CNNs) and computer vision technologies unlocks limitless possibilities. Deep learning techniques for object detection using convolutional neural networks have become more prevalent compared to feature and edge extraction methods like scale-invariant feature transform (SIFT) and Histogram of oriented gradients (HOG). Object detection models are trained to recognize the presence of instances of specific objects associated with a predefined class by enclosing the target object in a bounding box, identifying their class, and providing the probability of the object belonging to that class. Object detection models can be used in images, videos, or real-time operations.

The original You Only Look Once (YOLO) was introduced in 2016 by Joseph Redmon et al. [17] in a custom framework called Darknet. Using CNN, YOLO can predict all objects in a single forward pass, hence its full name, “You Only Look Once.” YOLO is one of the most influential algorithms for object detection, but before YOLO, the two-stage object detection architecture dominated the object detection field. YOLO was the first object detection network to combine the task of drawing bounding boxes and identifying class labels in one end-to-end differentiable network. Thus, it is referred to as a single-stage object detector. The general YOLO architecture consists of various parts such as the backbone, neck, and head. The process starts by feeding the input (which is the training dataset) to the network, where they are processed in batches in parallel by the GPU. The backbone network performs feature extraction to compute feature maps from the input images. The neck is a subset of the bag of specials that conducts feature aggregation by collecting feature maps from different stages of the backbone before passing them on to the prediction head. There are many versions of the YOLO real-time object detection models, but for this research, we utilize three YOLO models, namely: YOLOv7, YOLOv5s, and YOLOV5m.

2.2. YOLOv5

In June 2020, YOLO Version 5 (YOLOv5) was released by a company called Ultralytics. YOLOv5 is an iteration of the YOLO series and a state-of-the-art single-stage object detection algorithm. YOLOv5 is based on Scaled-YOLOv4, but unlike YOLOv4, YOLOv5 uses the PyTorch framework instead of Darknet. As shown in Figure 2, YOLOv5 employs Cross Stage Partial Darknet (CSPDarknet) as the backbone to extract essential features from the given input image. CSPDarknet53 backbone within YOLOv5 consists of 29 convolutional layers 3×3 , a receptive field size of 725×725 , and 27.6 M parameters. The Spatial Pyramid Pooling (SPP) block attached over YOLO's CSPDarknet53 expands the proportion of receptive fields without influencing its operating speed. CSPNet overcomes the issue of repeated gradient information in large-scale backbones by integrating gradient changes into the feature map, reducing the model's parameters and FLOPS (floating-point operations per second), and ensuring inference, speed, and accuracy while reducing model size.

The architecture's neck consists of layers that blend and integrate representational image features to proceed further with prediction. The feature aggregation is performed through Path Aggregation Network (PANet) by exploiting different backbone levels. Yolov5 used PANet to improve information flow. PANet employs a new feature pyramid network (FPN) structure with an improved bottom-up path to improve the propagation of low-level features. Concurrently, adaptive feature pooling, which connects the feature grid and all feature levels, propagates valuable information in each feature level directly to the following subnetwork. PANet enhances the utilization of accurate localization signals in lower layers, improving object location accuracy.

The head utilizes features from the neck and generates predictions from the anchor boxes for object detection and class prediction. The head generates three different sizes of feature maps (18×18 , 36×36 , 72×72) to achieve multi-scale prediction, enabling the model to handle small, medium, and

large objects. The activation functions that YOLOv5 uses are leaky rectified linear unit (ReLU) and sigmoid activation. YOLOv5 uses stochastic gradient descent (SGD) and ADAM as optimizer options and Binary cross-entropy with logit loss as a loss function. YOLOv5 pushes state-of-the-art features such as weighted-residual connections, cross-stage partial-connections, cross mini-batch, normalization, and self-adversarial training, making it exceptionally efficient.

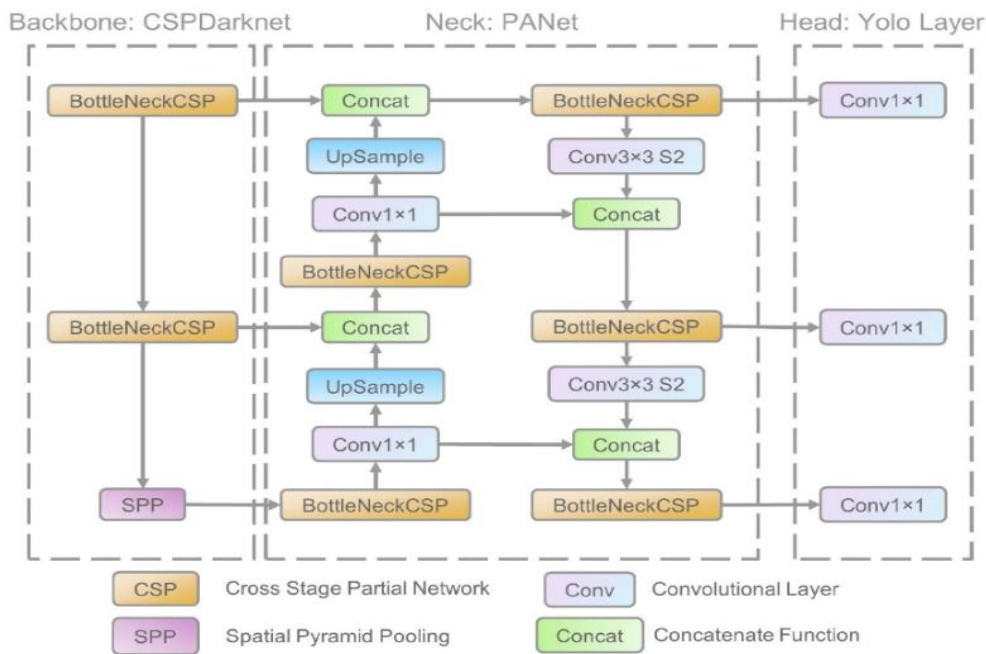


Figure 2. Generalized YOLOv5 network architecture [23]

YOLO models usually release a series of variant models by scaling up and down the model size depending on parameters such as the width (number of channels) and depth (number of layers) for different use cases. Although these variants belong to the same YOLOv5 family, there are still significant changes to the models that can alter their performance on the same dataset. Figure 3 illustrates the different scales of the YOLOv5 models, and Table 2 provides the difference in specification between the various YOLOv5 models.

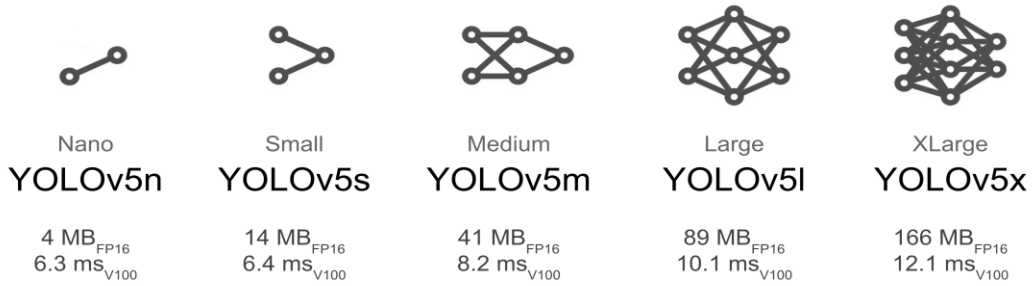


Figure 3. Different variants of the YOLOv5 model [23]

Table 2. Specification of YOLOv5 model variants [23]

Model	Number of Layers	Number of Parameters	FLOPs
YOLOv5n (nano)	270	1,797,927	4.2
YOLOv5s (small)	270	7,027,720	15.9
YOLOv5m (medium)	369	20,879,400	48.1
YOLOv5l (large)	468	46,149,064	108
YOLOv5x (extra-large)	567	86,231,272	204.2

YOLOv5 offers five models including YOLOv5 extra small, medium, large, and extra-large and each delivers different levels of detection accuracy and performance. YOLOv5n is the smallest in the family and is used for mobile deployment. YOLOv5s is a small model with 7.2 million parameters, and YOLOv5m is a medium-sized model with 21.2 million parameters. They are both suitable for cloud deployments. YOLOv5l is a large model of the YOLOv5 family with 46.5 million parameters. It is appropriate for datasets requiring smaller object detection. YOLOv5x is the largest among the five models, with 86.7 million parameters. It is slower to run because models with more parameters need more CUDA memory to train [23].

2.3. YOLOv7

Wang et al. [26] published the latest official YOLO version, YOLOv7, in July 2022. The YOLOv7 aimed to advance object detection by designing a network architecture that would predict more accurately than its peers at comparable inference speeds. The YOLOv7 architecture is based on previous YOLO model architectures, such as YOLOv5s, Scaled YOLOv5s, and YOLO-R.

Model re-parameterization merges numerous computational models at the inference stage to accelerate inference time. It takes motivation from prior studies on network efficiency. As shown in Figure 4, YOLOv7's backbone uses E-ELAN (Extended efficient layer aggregation networks) for model re-parameterization. The E-ELAN architecture of YOLOv7 uses expand, shuffle, and merge cardinality to continuously enhance the network's learning ability without transforming the original gradient path. This approach employs group convolution to extend the channel and cardinality of computational blocks by using the same group parameter and channel multiplier for every computational block in the layer. The block then calculates the feature map, shuffles it into many groups, and combines it. The groups are combined to merge cardinality, ensuring that the number of channels in each group of feature maps is the same as that of the original architecture. Changing the model architecture only in the computational block does not affect the transition layer, and the gradient path remains fixed. The E-ELAN design analyzed the following factors that influence speed and accuracy; Memory access cost, I/O channel ratio, element wise operation, activations, and gradient path.

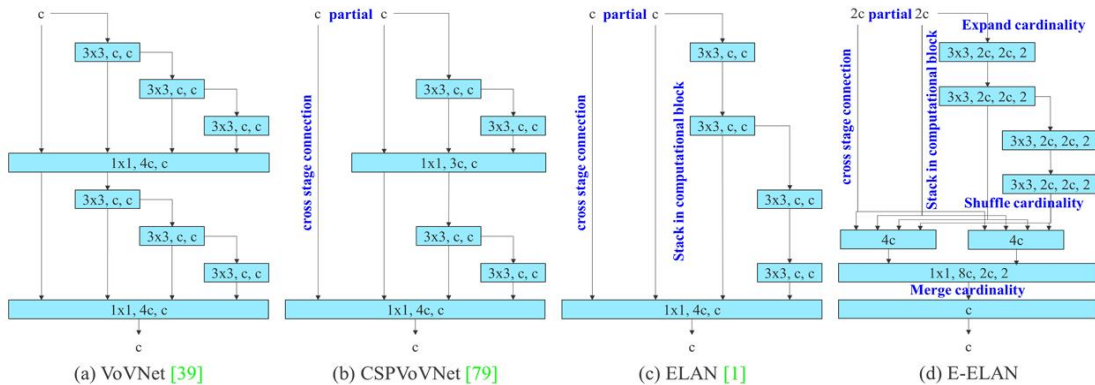


Figure 4. Extended efficient layer aggregation networks [26]

The YOLOv7 model concurrently scales the network depth and width while concatenating layers, as shown in Figure 5. Ablation studies reveal that the compound scaling technique optimizes the model architecture while scaling for different sizes.

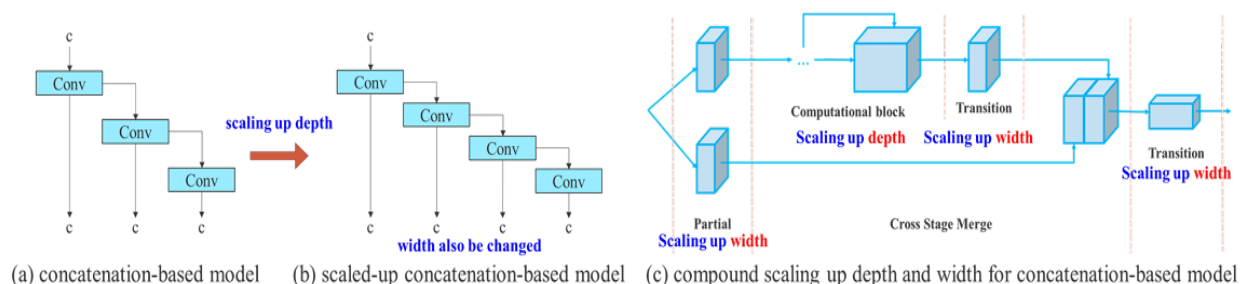


Figure 5. Model scaling for concatenation-based models [26]

Bag of Freebies (BoF) are techniques that increase the model’s performance without increasing training costs. Re-parameterization is a technique that averages a set of model weights to create a more robust model for generalization. Model level and module level ensemble are two types of re-parameterizations used to finalize models. Recent studies show that module-level re-parameterization has acquired traction. This process splits the model training process into multiple modules, and the outputs are ensembled to obtain the final model. YOLOv7 uses gradient flow propagation paths to analyze how to combine re-parameterized convolution with different networks. Figure 6 shows how to place the convolutional blocks with the check-marked options representing that they worked.

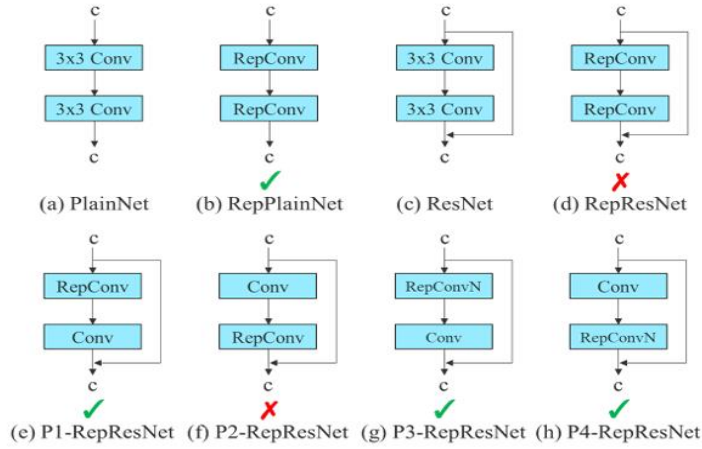


Figure 6. Planned re-parameterized model [26]

The head contains the predicted model outputs. YOLOv7 is inspired by Deep Supervision, a technique commonly used in training deep neural networks. The lead head is in charge of the final output, while the auxiliary head is in charge of assisting with training in the middle layers. In addition, a Label Assigner mechanism, as shown in Figure 7, was introduced to improve deep network training, which considers network prediction results and ground truth before assigning soft labels. In contrast to traditional label assignment, which relies solely on the ground truth to generate complex labels based on given rules, reliable soft labels employ calculation and optimization methods that consider the quality and distribution of prediction output in addition to the ground truth.

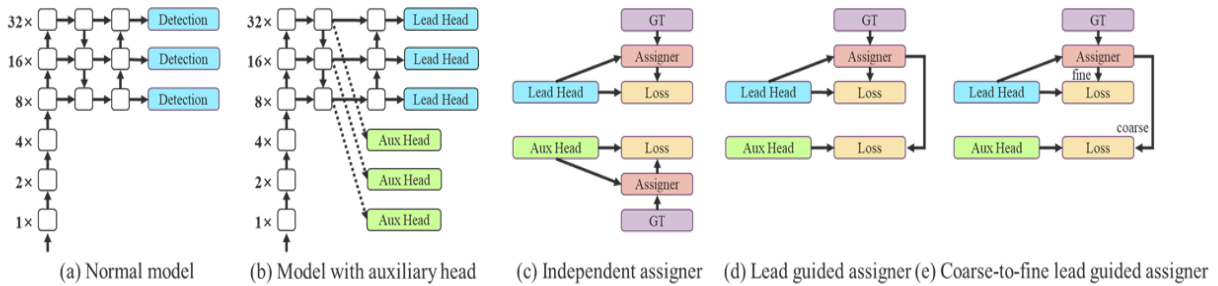


Figure 7. Coarse for auxiliary and fine for lead head label assigner [26]

2.4. Summary

Vehicle detection combines classification and localization to specify the exact locations of vehicles in an image using bounding boxes and classify the vehicles into previously defined classes. YOLO is a leading object detection algorithm and was the first one-stage object detection network that combined the task of drawing bounding boxes and identifying class labels in one end-to-end differentiable network. There are many renditions of the YOLO real-time object detection models, but for this research, we utilize three YOLO models, including YOLOv7, YOLOv5m, and YOLOv5s. These are the latest official YOLO models. YOLOv7 was only released a few months ago, and there has not been extensive research on this model. We use two models from the YOLOv5 family to investigate how they compare to YOLOv7. Although these variants belong to the same YOLOv5 family, there are still significant changes to the models that can alter their performance on the same dataset.

The differences in the YOLOv5 and YOLOv7 architectures contribute to their differences in performance. YOLOv5 employs Cross Stage Partial Darknet (CSPDarknet) as the backbone, while YOLOv7's backbone uses E-ELAN (Extended efficient layer aggregation networks) for model re-parameterization. Model re-parameterization can significantly increase the architecture's performance by enhancing the network's learning ability. Unlike YOLOv5, YOLOv7 introduces a compound scaling method applied to concatenation-based architectures to calculate alterations in the computational block's output channel. The recommended compound scaling method maintains the qualities of the original and optimal model design. Unlike the traditional independent label assigner in YOLOv5, YOLOv7 proposes a multi-headed framework. The lead head is in charge of the final output. The auxiliary head helps with training in the middle layers. The YOLOv7 model learns better by simultaneously getting the training lead head and auxiliary head labels.

3. METHODOLOGY

3.1. Data Collection

Data collection formed the most integral part of this research work since we employ supervised learning to train our data for vehicle detection and classification. The initial concept for data collection was to take a video recording of busy highways at varying times. However, it became evident that relying solely on video recordings may not be as effective because of the overwhelming number of passenger cars and comparatively fewer instances of other vehicle classes. Hence, vehicle images from the internet seemed to alleviate this problem since it would be easier to control what vehicle classes would be incorporated into work. This enables the collection of vehicles from rarely seen classes like the multi-trailer trucks. Vehicles were captured from different angles, including the front, rear, side, and aerial views, to ensure that the model learns to identify the vehicle from any viewpoint. Vehicles of each class come in different brands, shapes, sizes, and colors, so capturing many vehicles belonging to different classes and not just one type was essential. The vehicles were also captured at different times and weather conditions, including sunny, rainy, snowy, and nighttime. For this study, 3,327 images with different weather conditions were collected to form the dataset used for vehicle classification, as shown in Table 3 and Figure 8 for the detailed distribution of the number of images for each weather condition in the formed dataset. The data was gathered from different sources; about 30% of the total data was gathered by recording videos of the I-94 at Otsego, Minnesota. The remaining 70% of the data was gathered using the google search engine and videos from YouTube. These images have different resolutions, which are necessary for image variety, promoting the model's robustness. Table 4 shows the different sources and the different image resolutions.

Sunny



Snowy



Rainy



Night time



Figure 8. Vehicles captured in sunny, rainy, snowy, and night conditions.

Table 3. Number of frames for each time of day and weather condition

Weather Condition	Number Of Frames
Sunny	1,753
Snowy	602
Rainy	455
Night	517

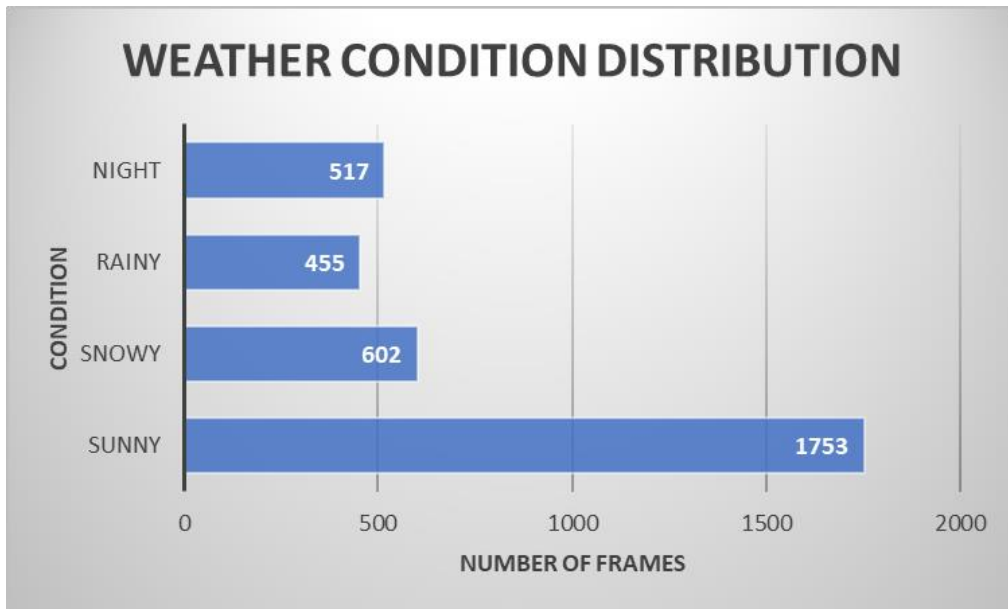


Figure 9. Frequency distribution of the number of frames for each condition

Table 4. Percentages of different resolutions

Source	Percentage	Pixels
Recorded Video at I94 Otsego, Minnesota	0.3	1280x720
Google Search	0.1	259x194
Google Search	0.08	275x183
Google Search	0.04	284x177
Google Search	0.02	700x319
Google Search	0.09	1000x665
YouTube Video	0.05	1920x1000
YouTube Video	0.15	1920x1080

3.2. Vehicle Classes

Although most states based their vehicle classification on the FHWA’s 13-category classification system in Table 1, many individuals, companies, and agencies have revised the original FHWA classification system numerous times to address some of the system’s shortcomings. A key recommendation supporting vehicle detection and classification purposes is to use only three or four generic categories of vehicles provided in Table 1 [4]. Some researchers recommend using an aggregated classification scheme because, in some States, volumes in many of the 13-FHWA vehicle categories are very low [5]. When volumes within a vehicle class are low, the models predict the classes very poorly, resulting in low accuracies. This study suggests six vehicle classes, as shown in Table 5 and Figure 10.

Table 5. Suggested vehicle classification scheme

Class	Vehicle Type
Class 1	Motorcycles – Two or three-wheeled motorized vehicles. Vehicles in this class usually have saddle-type seats controlled by handlebars instead of steering wheels. Class 1 includes motorcycles, motor-powered bicycles, motor scooters, mopeds, and three-wheel motorcycles.
Class 2	Passenger Cars – Passenger Cars – All two axle four-tire vehicles designed mainly to transport passengers with/ without recreational or light trailers such as sedans, coupes, station wagons, pickups, panels, vans, campers, motor homes, ambulances, hearses, carryalls, and minivans.
Class 3	Buses – All two or more axle vehicles built as passenger-carrying buses. This class contains only conventional buses (including school buses), customized buses should be considered to be a truck and should be suitably classified.
Class 4	Single Unit Trucks – All two or more axle trucks on a single frame. Truck tractor units without a trailer is regarded as single-unit trucks.
Class 5	Combination trucks - All four or more axle trucks with two units, the tractor or straight truck power unit and the trailer. This class contains semi-trailer trucks.
Class 6	Multi-trailer trucks - All five or more axle trucks with a tractor and two or more trailer units.

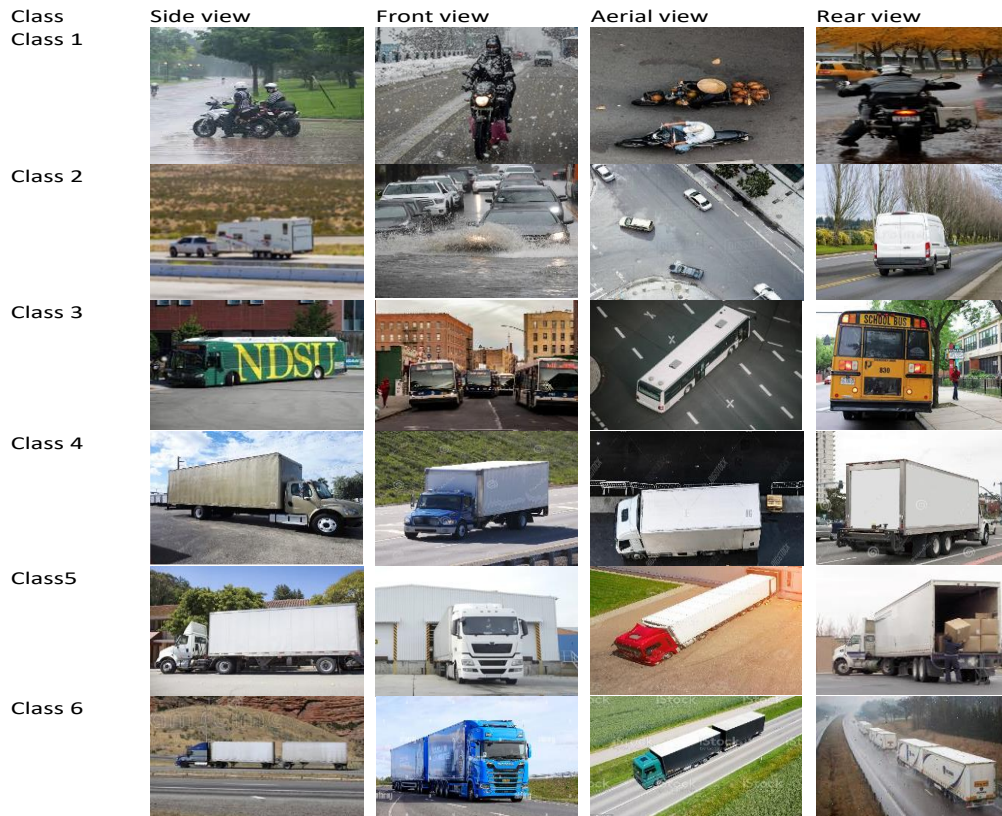


Figure 10. Vehicle classes from different angles

3.3. Vehicle Annotation

Data annotation is the use of corresponding bounding box coordinates (bottom left and top right (x, y)) to enclose an object belonging to a predefined class (label). Vehicle annotation is a time-consuming and exhaustive process that requires consistent, reliable, and accurate labeling to not feed the deep learning model with wrong information since we are training the model to learn from our annotated data, which could affect the model's accuracy. The data annotation type chosen for this research is object detection, and the data annotation technique used is the bounding box. The bounding box technique was preferred to other techniques because it is a relatively faster, less expensive, and user-friendly alternative. This study derived 8,587 annotated instances across 3,327 frames using a software called Computer Vision Annotation Tool (CVAT). The CVAT was the preferred annotation tool as it is beginner-friendly and has advanced annotation functionality. The overall statistics for the annotated vehicle instances are summarized in Table 6. It is very evident that the number of Class 2 (passenger cars) instances dominates the dataset, which is a true reflection of most traffic conditions.

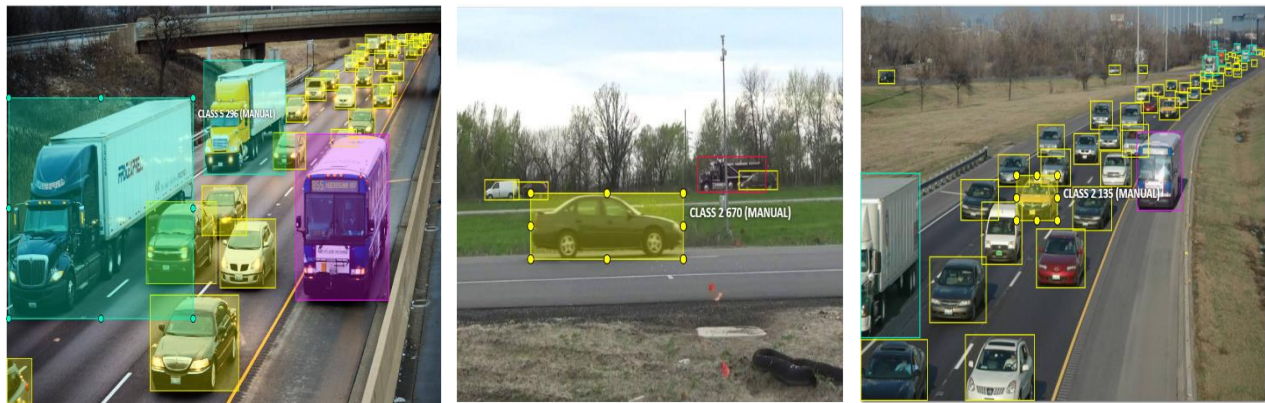


Figure 11. Vehicle annotation using CVAT

Table 6. Number of vehicle instances for each class

Class	Number Of Instances
Class 5	364
Class 1	429
Class 4	883
Class 6	1,039
Class 3	1,551
Class 2	4,321

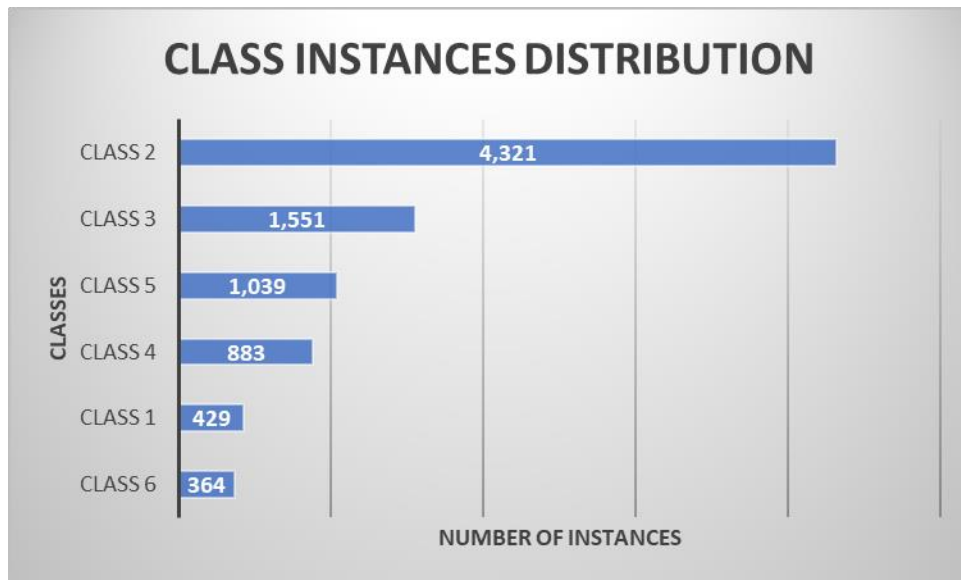


Figure 12. Frequency distribution of the number of class instances

There are some challenges encountered during the data annotation, as illustrated in Figure 13, which may influence the classification accuracy later, including:

- 1) Occlusion: In frames where two or more vehicles are too close and appear to merge, the vehicle that does not fully appear becomes difficult to annotate since an integral feature is omitted.
- 2) Adverse weather conditions and tiny targets: Vehicles in adverse weather conditions and small vehicles not in the camera's focus tend to be quite difficult to label accurately. These

instances were meticulously labeled with potentially compressed bounding boxes from a very distant point of view.

- 3) Special vehicles: A few special vehicles with class ambiguity, such as tractors, dozers, and excavators, appear in some frames. From the literature review, it was observed that some researchers choose to assign a label to unknown vehicles, but for this study, these unique vehicles are entirely ignored.



Figure 13. Some annotation challenges encountered

3.4. Data Preprocessing and Augmentation

Roboflow, a developer tool for building computer vision models, was used for data preprocessing and augmentation. The dataset in this study contains images of various sizes. However, most modern convolutional neural networks usually expect equally sized square-shaped images as input. Thus, all the frames are resized to 417 x 417. Data augmentation is a technique used to train large neural networks by increasing the diversity of the data without collecting new samples. In most of the reviewed studies, data augmentation techniques were employed since small datasets tend to result in overfitting deep learning models. This study applied three data augmentation methods,

noise, grayscale, and flip, to the training dataset, as shown in Figure 14. After this technique, the number of frames for the dataset increased from about 3,327 to 7,985 frames.



Figure 14. Data augmentation methods applied to training data using roboflow

3.5. Model Training

The programming language used for this study was Python, as it provides access to great deep-learning libraries. Google Colaboratory (Colab) is a cloud-based Jupyter notebook that allows the writing and execution of arbitrary python code through the browser while providing access to computing resources, including GPUs. Google Colab offers an NVIDIA K80 / T4 GPU with 52GB RAM for training and testing neural networks. The code for each neural network is cloned from the GitHub accounts of the creators into Google Colab pro. The dataset was split into 70% for training, 15% for validation, and 15% for testing. Some hyperparameters were altered to ensure that the model was trained to generate the best results possible. The hyperparameter settings for training YOLOv5m, YOLOv7, and YOLOv5s have been summarized in Table 7. The PyTorch framework is used for all three models. The batch size is the number of training samples used in one iteration, which can affect the training speed and memory usage. The batch size was set to 16 for all three models because the

models train at an optimum speed while avoiding resource-exhausted errors caused by running out of memory. An optimization algorithm adjusts the attributes of the neural network, such as weights and learning rate. For training the neural network, we used stochastic gradient descent as a training optimizer for YOLOv5s and YOLOv5m and adaptive moment estimation as the optimizer for YOLOv7. The learning rate decides the step size at each iteration while moving toward a minimum loss function. It regulates the rate at which an algorithm revises the parameter estimates. Each model uses an initial learning rate of 0.01. The activation function controls whether or not to activate a neuron by calculating the weighted sum and adding bias. The activation function aims to introduce non-linearity into a neuron’s output to help reduce the overall loss and improve accuracy. All three models use the leaky rectified linear activation function. The epoch size denotes the number of passes the algorithm’s entire training dataset has completed. Each model was set to train on 1000 epochs. However, early stopping was enabled to allow the models to stop training when convergence occurs, revealing that the model is not experiencing any significant improvement in the last 100 epochs. Training YOLOv5m on 801 epochs took 17 hours 38 mins 2s, YOLOv5s took 10 hours 27 min 38 s to train on 699 epochs, and finally, YOLOv7 took 9 hours 14 mins 2s to train on 247 epochs. After training, the best weights were used for the testing process.

Table 7. The hyperparameter values used in training

Hyperparameters	YOLOv5s	YOLOv5m	YOLOv7
Batch Size	16	16	16
Learning Rate	0.01	0.01	0.01
Epoch Size	699	801	247
Activation Function	Leaky Rectified Linear (ReLU)	Leaky Rectified Linear (ReLU)	Leaky Rectified Linear (ReLU)
Optimization Algorithm	Stochastic Gradient Descent (SGD)	Stochastic Gradient Descent (SGD)	Adaptive Moment Estimation (ADAM)

3.6. Summary

In total, 3,327 vehicle images were collected from different angles, brands, shapes, sizes, weather conditions, times of day, and colors, to form a dataset/database for this study, ensuring that the model learns to identify the vehicle in diverse scenarios. An aggregated classification scheme with six vehicle classes based on the FHWA 13 vehicle classification scheme is recommended for this study. This study derived 8,587 annotated instances across 3,327 frames using Computer Vision Annotation Tool (CVAT) software. The dataset was split into 70% for training, 15% for validation, and 15% for testing. Three data augmentation methods, including noise, grayscale, and flip, were applied to the training dataset causing the number of frames to increase from about 3,327 to 7,985 frames. The programming language used for training and testing the models was Python. The models were trained on an NVIDIA K80 / T4 GPU with 52GB RAM using Google Colab pro. Some hyperparameters, such as batch size, learning rate, optimizer, activation function, and epochs, were altered to ensure the model training generated the best results possible. Training YOLOv5m on 801 epochs took 17 hours 38 mins 2s, YOLOv5s 10 hours 27 mins 38s to train on 699 epochs, and YOLOv7 took 9 hours 14 mins 2s to train on 247 epochs.

4. RESULTS AND DISCUSSION

4.1. Evaluation Metrics

To assess the performance of the three models, we deploy four evaluation measures, including precision, recall, F1-score, and mean average precision (MAP). The confusion matrix refers to a $N \times N$ matrix used to assess the performance of a classification model, where N represents the number of target classes. The matrix compares the ground truth to the model's predictions. This gives us a concise overview of how well the models perform and the types of prediction errors they make.

Table 8. Confusion matrix diagram for binary classification

Confusion Matrix		Actual	
		Positive	Negative
Predicted	Positive	TP (True Positive)	FP (False Positive)
	Negative	FN (False Negative)	TN (True Negative)

Notes: The definition of TP, TN, FP, and FN are below.

True Positive (TP): The actual is positive, and the model correctly predicted it as positive.

True Negative (TN): The actual is negative, and the model correctly predicted it as negative.

False Positive (FP): The actual is negative, but the model wrongly predicted it as positive.

False Negative (FN) The actual is positive, but the model wrongly predicted it as negative.

According to the established guidelines, determining the metrics to assess the performance of the three models relies on the dataset and its characteristics [29]. It was addressed earlier that the dataset contains an overwhelming amount of Class 2 vehicles. This is due to numerous factors, such as (1) real-life traffic contains more passenger cars, (2) Class 2 entails a variety of passenger cars such as campers, minibuses, pickups, vans, ambulances, and more, to capture that variety as much as we could, (3) other classes like Class 6 which represents multi-trailer trucks were not as easy to come by like passenger cars. Some literature suggests that using accuracy to evaluate a classification

model might not correctly reflect the classifier model's performance due to class imbalance. Thus, this study uses only precision, recall, MAP, and F1-score as the performance metrics to prevent misleading results [30].

Recall indicates the percentage of ground truth positives predicted as true positives. Mathematically, recall is the ratio of true positives to the sum of true positives and false negatives as:

$$Recall = \frac{True\ Positive(TP)}{True\ Positive(TP)+False\ Negative(FN)} \quad (1)$$

Precision represents the percentage of predicted positives that are true positives. It is denoted by the ratio of true positives to the sum of true positives and false positives as:

$$Precision = \frac{True\ Positive(TP)}{True\ Positive(TP)+False\ Positive(FP)} \quad (2)$$

The F1-score is a harmonic mean of precision and recall that summarizes a model's predictive performance by combining two otherwise competing metrics, which can be represented as:

$$F1 - score = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (3)$$

The F1-score is highest when precision equals recall.

In addition, MAP is estimated by taking all the classes' mean of the average precision (AP). The AP summarizes the precision-recall curve into one value representing the average of all precisions. Equations (4) and (5) depict the AP and MAP formula as follows:

$$AP = \sum_{k=0}^{k=n-1} [Recall(k) - Recall(k + 1)] * Precision(k) \quad (4)$$

$$MAP = \frac{1}{n} \sum_{k=1}^{k=n} AP_k \quad (5)$$

where AP_k is the AP of class k, and n is the number of classes. The MAP at 0.5 and 0.95 thresholds will be provided for this study.

4.2. Model Training and Validation

The validation dataset demonstrates how the trained model behaves on unseen data. Figures 16, 17, and 18 illustrate the results from the training and validation of each of the models. The plots show the training and validation loss against the number of training epochs. It is observed that the training losses are slightly lower than the validation losses. The training and validation losses converged after some iterations revealing that the model was not experiencing any more significant losses and was done learning. Even though YOLOv7 trained for the least number of epochs, it still obtained the highest precision and recall compared to the other models. YOLOv7's and YOLOv5s's precision surpassed 0.9, while YOLOv5m's precision only exceeded 0.8, even though it trained for the longest. The recall for all three models exceeded 0.8, with YOLOv7's model getting closer to 0.9. The MAP estimates how well the models are detecting objects, and YOLOv7 once again obtains better MAPs at both 0.5 and 0.95 thresholds.

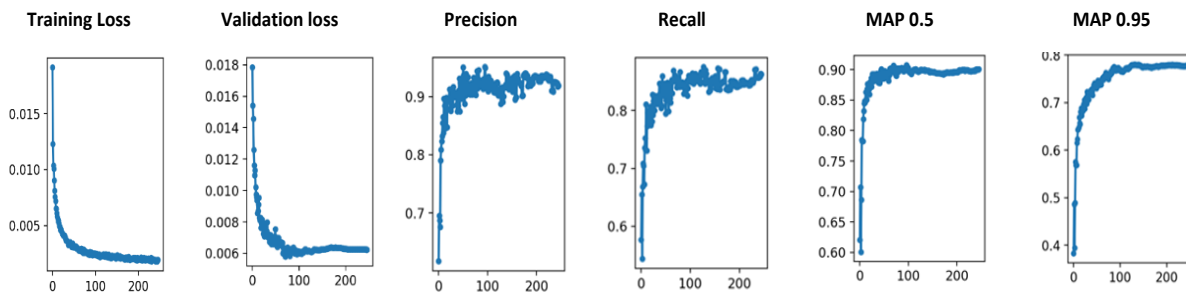


Figure 15. Results from YOLOv7 model training

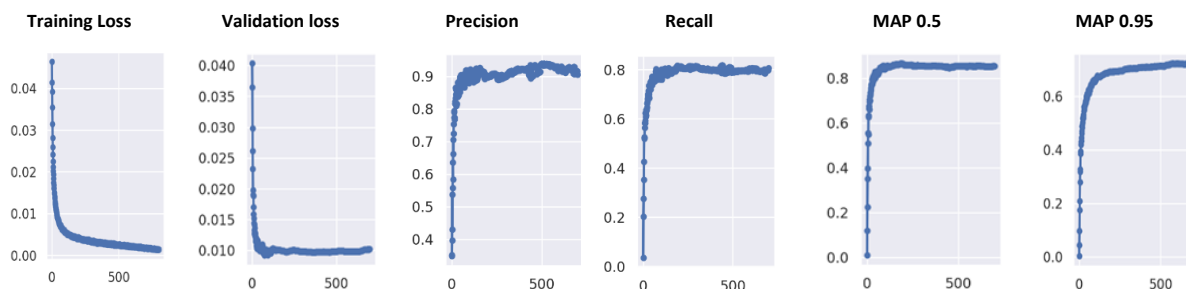


Figure 16. Results from YOLOv5s model training

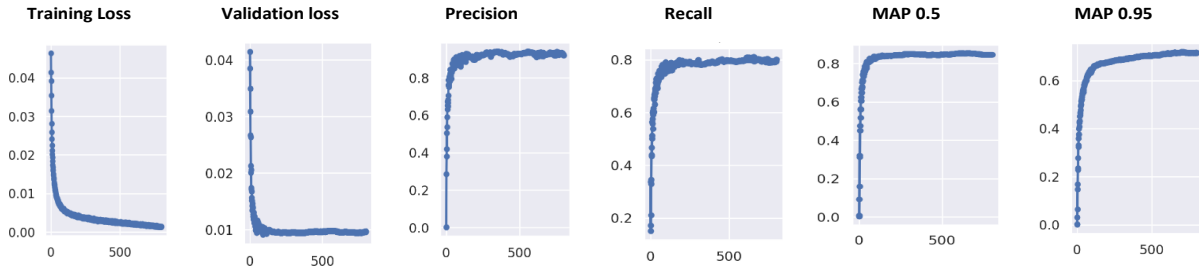


Figure 17. Results from YOLOv5m training

4.3. Model Testing Results

After training and validation, we tested the remaining 15% of the dataset allocated for testing using the best-trained weights. Testing is used to determine the generalization ability of the trained model on new data. Firstly, we generate each model's confusion matrix to understand the model's confusion when predicting vehicles of the various classes. According to the model developers, the object detection model's confusion matrix is unlike the traditional classification confusion matrix because most mistakes will be with the background class rather than other classes. Figure 18 illustrates the confusion matrix for YOLOv7, YOLOv5s, and YOLOv5m.



Figure 18. Confusion matrices for object detection models

YOLOv7 had the highest number of true positives for Class 1, followed by YOLOv5s and then YOLOv5m. YOLOv5s and YOLOv5m did not have very high true positives for Class 1, meaning the models could not detect this class and mistake it for the background. Figure 19 shows that though the annotator labeled the bicycle as Class 1, YOLOv5s could not detect the object of interest, counting it as a background, but YOLOv7 could correctly predict it. We can also observe from Figure 18 that YOLOv7 did not confuse Class 1 with any other class. In contrast, YOLOv5s and YOLOv5m mistake 2% of actual Class 1 for Class 2. Class 1 has the least interclass misclassification compared to the other classes. This means that Class 1 has distinct features making it relatively easier for the models to differentiate it from other classes. After all, it is the only class with two wheels, and it has a much smaller size than the other classes.

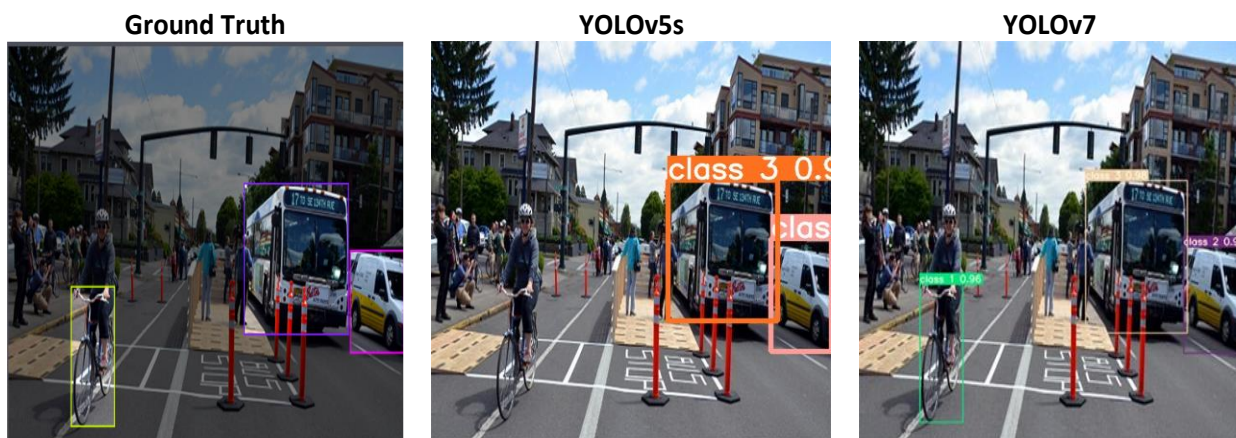


Figure 19. Example of YOLOv5s misclassifying class 1 as a background

The greatest number of Class 2 true positives are seen in YOLOv7, followed by YOLOv5s and YOLOv5m. YOLOv7 mistook 3% of actual Class 4 for Class 2, which is not surprising because some passenger pickups share similar features with smaller single-unit trucks. YOLOv5s wrongly predicted actual classes 1, 3, 4, and 5 as Class 2. YOLOv5m also misclassifies true classes 1, 3 and 4 as Class 2. For YOLOv5s and YOLOv5m, Class 2 still receives the highest number of true positives compared to the other classes. Class 2 has the most intraclass variations, consisting of pickups, vans,

and campers. Class 2 vehicles share some similarities with other classes, which could be a reason for all the interclass misclassifications associated with Class 2. In Figure 20, YOLOv5s wrongly predicts Class 3 as Class 2.



Figure 20. Example of YOLOv5s predicting a class 3 as class 2

Across all three models, Class 2 had the highest misclassification of actual backgrounds as Class 2 vehicles. Object detection models must be able to differentiate between the detection of the background and foreground of an image. Precision reduces when backgrounds are classified as vehicles, thus increasing the false positives. Deep learning models thrive on quality images, but to ensure the model was trained and tested on all kinds of image quality, we included some images with poor quality. In Figure 21, the YOLOv5m model detects the background and classifies it as Class 2. This happened in a snow weather condition image, probably, the model did not have enough snow weather images to learn from, or the image quality was just not high enough.

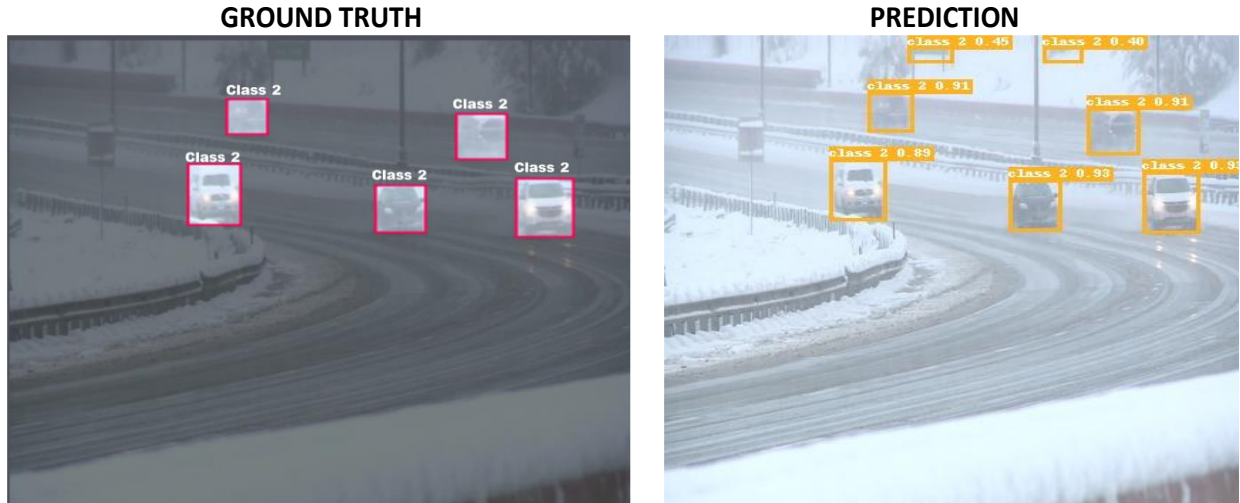


Figure 21. YOLOv5m misclassifying the background as class 2

During the annotation process, the annotator found some vehicles very difficult to label since images had to be zoomed in before being able to draw the bounding box around the vehicle. For some images, the vehicles were so far in the background that we decided not to annotate them. However, the models still detected some background vehicles, as seen in Figure 22. Hence, even though the model correctly predicted that vehicle, since it was not annotated as ground truth, this will contribute to the model not performing well because this will be counted as a false positive.



Figure 22. False positives due to the YOLOv7 detecting unlabeled vehicles

For Class 3, the true positives are highest in YOLOv7, then YOLOv5s, and then YOLOv5m. The three models do not have many confusions of Class 3 with other classes. YOLOv7 misclassifies only 1% of true Class 4 as Class 3. The FHWA vehicle classification scheme states that Class 3 should contain only conventional buses. Customized buses should be considered trucks and should be suitably classified. Because modified buses share the same color (yellow with black strips) as traditional buses, it could be that the YOLOv7 could still not accurately classify the modified buses. YOLOv5s and YOLOv5m predict 2% of actual Class 5 as Class 3. Again Class 3 represents buses that are quite distinct from other Classes; however, the models still had some false positives and false negatives because the models did not correctly detect some buses.

In classes 4, 5, and 6, YOLOv7 has the most significant number of true positives, followed by YOLOv5s and YOLOv5m. All three models confuse Class 4 for Class 5 and Class 6 and vice versa. Interclass misclassifications can occur between Class 4, Class 5, and Class 6 (the classes

designated to different types of trucks). Trucks belonging to Classes 4, 5, and 6 sometimes look similar when the truck is being filmed from the front view, aerial view, rear view, or when there is occlusion. This is not surprising since some frames do not capture the entire truck and may miss essential features like the number of truck axles. A multi-trailer truck can easily be confused with a semi-trailer truck when its multiple trailers are not revealed in that particular frame.

The results from the confusion matrices in Figure 18 are used to compute the precision, recall, and F1-score summarized in Table 9. The average of all the classes' precision, recall, and F1-scores have been provided for each model.

Precision informs us about the accuracy of the model's positive predictions, and precision is preferred when False Positives are more relevant than False Negatives. From the experimental results in Table 9, the YOLOv7 model had the highest average precision compared to YOLOv5s and YOLOv5m. This means that on average YOLOv5m had the highest False Positive during prediction. The highest precision for YOLOv7 is Class 6, but for YOLOv5s and YOLOv5m, it is Class 1. Class 2 had the lowest precision for all three models. In scenarios where False Negatives are more relevant than False Positives, recall is the best metric. Recall estimates the number of actual positive instances predicted correctly. Again, YOLOv7 had the highest average recall, followed by YOLOv5s and YOLOv5m. Class 5 had the highest recall for YOLOv7, but Class 2 had the highest recall for YOLOv5s and YOLOv5m. Class 1 recorded the least recall for all three models. Ideally, a model would have perfect precision and recall, but practically, a tradeoff usually exists between the two.

Table 9. Summary of performance metrics of model testing

YOLOV7					
Class Name	Precision	Recall	F1-Score	MAP @ 0.5	MAP @0.95
CLASS 1	0.953	0.820	0.882	0.848	0.679
CLASS 2	0.571	0.929	0.708	0.944	0.752
CLASS 3	0.878	0.869	0.873	0.882	0.802
CLASS 4	0.906	0.879	0.892	0.947	0.871
CLASS 5	0.795	0.970	0.874	0.967	0.889
CLASS 6	0.978	0.900	0.938	0.989	0.952
Average	0.847	0.894	0.861	0.930	0.824

YOLOV5s					
Class Name	Precision	Recall	F1-Score	MAP @ 0.5	MAP @ 0.95
CLASS 1	0.982	0.550	0.705	0.604	0.437
CLASS 2	0.569	0.870	0.688	0.892	0.679
CLASS 3	0.823	0.657	0.730	0.727	0.508
CLASS 4	0.814	0.782	0.798	0.848	0.549
CLASS 5	0.822	0.830	0.826	0.910	0.704
CLASS 6	0.904	0.859	0.881	0.850	0.521
Average	0.819	0.758	0.771	0.805	0.566

YOLOV5m					
Class Name	Precision	Recall	F1-Score	MAP @ 0.5	MAP @ 0.95
CLASS 1	0.982	0.535	0.692	0.604	0.438
CLASS 2	0.586	0.859	0.697	0.878	0.674
CLASS 3	0.778	0.566	0.655	0.684	0.490
CLASS 4	0.800	0.640	0.711	0.761	0.534
CLASS 5	0.806	0.822	0.814	0.896	0.714
CLASS 6	0.884	0.752	0.813	0.852	0.530
Average	0.806	0.696	0.730	0.779	0.563

Due to this tradeoff, it is vital to comprehend the task we are attempting to solve and any underlying consequences of prioritizing False Positives over False Negatives or vice versa. False Positives and False Negatives are instrumental to this study, and given this competing tradeoff, it is very convenient to have the F1-score as a single performance metric that is neutral to both precision and recall. For YOLOv7 and YOLOv5s, Class 2 has the lowest F1-score, and Class 3 has the lowest F1-score for YOLOv5m. Class 6 has the highest F1-score for YOLOv7 and YOLOv5, and Class 5 has the highest F1-score for YOLOv5m. On average, YOLOv7 scored the highest F1-score of 0.861, then YOLOv5s with 0.771, and YOLOv5m with 0.730. The analysis shows that all three models had

performances comparable to the results in previous research that trained YOLOv7, YOLOv5s, and YOLOv5m (C-Y. Wang et al. [26], Horvat Marko et al. [31] and Uzar et al. [23]).

Most researchers prefer evaluating the object detection models using the mean average precision because it compares the ground-truth bounding box to the predicted bounding box. The MAP is measured at a 0.5 and 0.95 intersection over the union (IOU) threshold. The object detector with a higher MAP score is considered the more accurate object detection model. In this study, the YOLOv7 model has a 0.93 MAP at 0.5 thresholds and 0.824 MAP at 0.95 thresholds, the highest MAP at both 0.5 and 0.95 thresholds among all the models. The YOLOv5s gives the second highest MAP at both thresholds with a score of 0.805 and 0.566 at a threshold of 0.5 and 0.95, respectively. The least performing model is YOLOv5m, with a MAP score of 0.779 at 0.5 and 0.563 at 0.95.

These results may be because YOLOv7's backbone uses E-ELAN (Extended efficient layer aggregation networks) for model re-parameterization, significantly improving the network's learning ability. YOLOv7 also uses a compound scaling method to compute alterations in the computational block's output channel. Finally, YOLOv7 uses a multi-headed framework that improves performance by allowing the shallower auxiliary head to directly learn the information that the lead head has already learned. Hence, the lead head is more focused on learning residual data that has not been learned previously.

4.4. Summary

The performances of the three models are estimated using precision, recall, F1-score, and mean average precision (MAP) at 0.50 and 0.95 thresholds. Accuracy is not used because the study was conducted using imbalanced data. We observed that the training and validation decrease till they converge while the performance measures: precision, recall, and MAP increase till they record high scores showing that the models are performing well. The comparison of the three deep learning models reveals the dominance of the YOLOv7 model in detecting and classifying vehicles scoring

the highest performance results with 84.7% precision, 89.4% recall, 86.1% F1-score, 93% MAP at 0.5 thresholds and 82.4% MAP at 0.95 thresholds on average. YOLOv5s is the second-best result, followed by YOLOv5m. YOLOv5s scored the second highest results with 81.9% precision, 75.8% recall, 77.1% F1-score, 80.5% MAP at 0.5 threshold, and 56.6% MAP at 0.95 threshold. YOLOv5m was the worst performing model with 80.6% precision, 69.6% recall, 73% F1-score, 77.9% MAP at 0.5 threshold, and 56.3% MAP at 0.95 threshold.

5. CONCLUSIONS AND FUTURE WORK

This thesis compares the robustness of three state-of-the-art object detection models (YOLOv7, YOLOv5m, and YOLOv5s) for vehicle classification according to a modified FHWA vehicle classification scheme. We generated 7,985 frames from a heterogeneous dataset containing images of vehicles in different conditions, including sunny, snowy, rainy, and night. Each model was trained on the same dataset allocating 70% for training, 15% for validation, and 15% for testing. After running inference on the trained models, we calculated the precision, recall, MAP, and F1-scores, which were used as metrics to evaluate and identify the best model. The conclusions of this thesis can be drawn as follows:

- 1) YOLOv5m trained 801 epochs in 17 hours 38 minutes 2 seconds, YOLOv5s trained 699 epochs in 10 hours 27 minutes 38 seconds, and YOLOv7 trained 247 epochs in 9 hours 14 minutes 2 seconds. The training and validation loss plots against the epochs show that the training losses are slightly lower than the validation losses for all models. The training and validation losses converged after some iterations indicating that the model finished training. Despite training for the least epochs, YOLOv7 obtained the highest precision and recall compared to the other models.
- 2) Interclass misclassifications can occur between Class 4, 5, and Class 6. The classes designated to different truck types increase the False Positives. Classes 4, 5, and 6 trucks sometimes look similar when filmed from the front view, aerial view, or when there is occlusion. Some frames only show a portion of the truck and may miss vital features like the number of truck axles. A multi-trailer truck can easily be mistaken for a semi-trailer truck when its multiple trailers are excluded from that particular frame.
- 3) Two explanations for the algorithms' difficulties distinguishing foreground and background are that deep learning models thrive on high-quality images, yet the formed heterogeneous dataset contains poor-quality nighttime and inclement weather images. Additionally, because

certain occurrences were left unlabeled by the annotator when the model accurately predicted these vehicles, the detections will be regarded as false positives.

- 4) The YOLOv7 model had the highest average precision compared to YOLOv5s and YOLOv5m. This means that on average YOLOv5m had the highest False Positive during prediction. The highest precision for YOLOv7 is Class 6, but for YOLOv5s and YOLOv5m, it is Class 1. Class 2 had the lowest precision for all three models.
- 5) YOLOv7 had the highest average recall, followed by YOLOv5s and YOLOv5m. Class 2 had the highest recall for YOLOv7, but Class 2 had the highest recall for YOLOv5s and YOLOv5m. Class 1 recorded the least recall for all three models.
- 6) In this study, the YOLOv7 model has a 0.93 MAP at 0.5 thresholds and 0.824 MAP at 0.95 thresholds, the highest MAP at both 0.5 and 0.95 thresholds among all the models. The YOLOv5s gives the second highest MAP at both thresholds with a score of 0.805 and 0.566 at a threshold of 0.5 and 0.95, respectively. The worst-performing model is YOLOv5m, with a MAP score of 0.779 at 0.5 thresholds and 0.563 at 0.95 thresholds.
- 7) The reason why YOLOv7 dominates the other models may be due to the difference in the models' architectures. In the YOLOv7 backbone, E-ELAN (Extended efficient layer aggregation networks) is used as the computational block for model re-parameterization to dramatically improve the YOLOv7's learning ability. YOLOv7 also employs a compound scaling method to calculate alterations in the computational block's output channel. Finally, YOLOv7 uses a multi-headed framework to increase performance by ensuring the shallower auxiliary head can directly learn the data the lead head has already learned. Hence, the lead head is more focused on learning residual data that has not been learned yet.

In future work, researchers can create a new benchmark dataset containing a balanced number of instances for all 13 FHWA vehicle classes and investigate how differently the models perform,

considering accuracy as an evaluation measure. Since this study focuses primarily on urban areas, future work can concentrate on rural areas and examine how individual weather conditions influence the performance of the models. Future studies can also compare the deep learning models' performance to the traditional sensor for vehicle detection and classification.

REFERENCES

- [1] Sun C., Ritchie S.G. Heuristic vehicle classification using inductive signatures on freeways. *Transp. Res. Rec.* 2000;17:130–136. doi: 10.3141/1717-16.
- [2] Ambardekar, A., Nicolescu, M., Bebis, G. et al. Vehicle classification framework: a comparative study. *J Image Video Proc* 2014, 29 (2014). [https://doi.org/10.1186/1687 - 5281-2014-29](https://doi.org/10.1186/1687-5281-2014-29)
- [3] Refai, Hazem & Bitar, Naim & Al Kalaa, Mohamad Omar & Schettler, Jesse. (2014). *The Study of Vehicle Classification Equipment with Solutions to Improve Accuracy in Oklahoma.*
- [4] James L. Randall. (2012). *Manual: Traffic Recorder Instruction Manual*
- [5] Verification, Refinement, and Applicability of Long-Term Pavement Performance Vehicle Classification Rules Chapter 2. *Introduction To Vehicle Classification.* (2014). <https://www.fhwa.dot.gov/publications/research/infrastructure/pavements/ltp/13091/002.cfm>
- [6] Murrugarra R., Wallace W., Wojtowicz J. *Task 30: Data Fusion Methodology; Technical Report 10-06.* Center for Infrastructure, Transportation and the Environment, Department of Civil and Environmental Engineering, Rensselaer Polytechnic Institute; Troy, NY, USA: 2010
- [7] Sun Z., Ban X. Vehicle classification using GPS data. *Transp. Res. Part C Emerg. Technol.* 2013;37:102–117. doi: 10.1016/j.trc.2013.09.015.
- [8] Mu'ath Ahmad Al-Tarawneh. 2016. *In-Pavement Fiber Bragg Grating Sensors for Weight-In-Motion Measurements*
- [9] Roh H.J., Sharma S., Sahu PK *Modeling snow and cold effects for classified highway traffic volumes.* *KSCE J. Civ. Eng.* 2016;20:1514–1525. doi: 10.1007/s12205-015-0236-0.

- [10] Abdullah N.F., Rashid N.E.A., Othman K.A., Khan Z.I., Musirin I. Ground vehicles classification using multi perspective features in FSR micro-sensor network. *J. Telecommun. Electron. Comput. Eng.* 2017;9:49–52.
- [11] Alexandre E., Cuadra L., Salcedo-Sanz S., Pastor-Sánchez A., Casanova-Mateo C. Hybridizing Extreme Learning Machines and Genetic Algorithms to select acoustic features in vehicle classification applications. *Neurocomputing.* 2015;152:58–68. doi: 10.1016/j.neucom. 2014.11.019
- [12] Shokravi H, Shokravi H, Bakhary N, Heidarrezaei M, Rahimian Kolor SS, Petru M. A Review on Vehicle Classification and Potential Use of Smart Vehicle-Assisted Techniques. *Sensors (Basel).* 2020 Jun 8;20(11):3274. doi: 10.3390/s20113274. PMID: 32521806; PMCID: PMC7309154.
- [13] Y. Roh, G. Heo, and S. E. Whang. 2019. A Survey on Data Collection for Machine Learning A Big Data- AI Integration Perspective.
- [14] Espinosa, J.E., Velastin, S.A. y Branch, J.W. 2017. "Vehicle Detection Using Alex Net and Faster R-CNN Deep Learning Models: A Comparative Study. In *Advances in Visual Informatics.*" *Lecture Notes in Computer Science*, 10645, pp. 3-15.
- [15] Biplav Sharma Regmi, Ramesh Thapa, Biplove Pokhrel. 2021. "Comparative study of CCTV based Vehicle Identification and Classification Models during Adverse Conditions in Pokhara."
- [16] Erkut Akdag, Egor Bondarev and Peter H. N. De With. 2022. "Critical Vehicle Detection for Intelligent Transportation Systems."
- [17] Arinaldi, Ahmad & Pradana, Jaka & Gurusinga, Arlan. (2018). Detection and classification of vehicles for traffic video analytics. *Procedia Computer Science.* 144. 259-268. 10.1016/j.procs.2018.10.527.

- [18] Mardin Abdullah Anwer, Shareef M. Shareef, Abbas M. Ali. 2021. "Accident vehicle types classification: a comparative study between different deep learning models."
- [19] Faruque, M. O., Ghahremannezhad, H., & Liu, C. (2019). Vehicle classification in video using deep learning. In the 15th International Conference on Machine Learning and Data Mining (pp. 117-131).
- [20] Amund Hansen Vedal. 2017. "Comparing performance of convolutional neural network models on a novel car classification task."
- [21] Muhammad Atif Butt, Asad Masood Khattak , Sarmad Shafique, Bashir Hayat , Saima Abid, Ki-Il Kim , Muhammad Waqas Ayub, Ahthasham Sajid, and Awais Adnan. 2021. "Convolutional Neural Network Based Vehicle Classification in Adverse Illuminous Conditions for Intelligent Transportation Systems."
- [22] Sumeyye CEPNI, Muhammed Enes ATIK and Zaide DURAN. 2020. "Vehicle Detection Using Different Deep Learning Algorithms from Image Sequence."
- [23] Uzar, M., Öztürk, Ş., Bayrak, O. C., Arda, T., & Öcalan, N. T. 2021. "Performance analysis of YOLO versions for automatic vehicle detection from UAV images." *Advanced Remote Sensing*, 1(1), 16-30
- [24] C. -Y. Wang, A. Bochkovskiy and H. -Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection" 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 13024-13033, doi: 10.1109/CVPR46437.2021.01283.
- [25] Chien-Yao Wang et al. 2022. "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors."
- [26] Sokolova M., Lapalme G. A systematic analysis of performance measures for classification tasks. *Inf. Process. Manag.* 2009;45:427–437.

- [27] Pradeep Kumar et al. 2021. Classification of Imbalanced Data: Review of Methods and Applications.
- [28] Horvat, Marko & Jelečević, Ljudevit & Gledec, Gordan. (2022). A comparative study of YOLOv5 models performance for image localization and classification.

APPENDIX

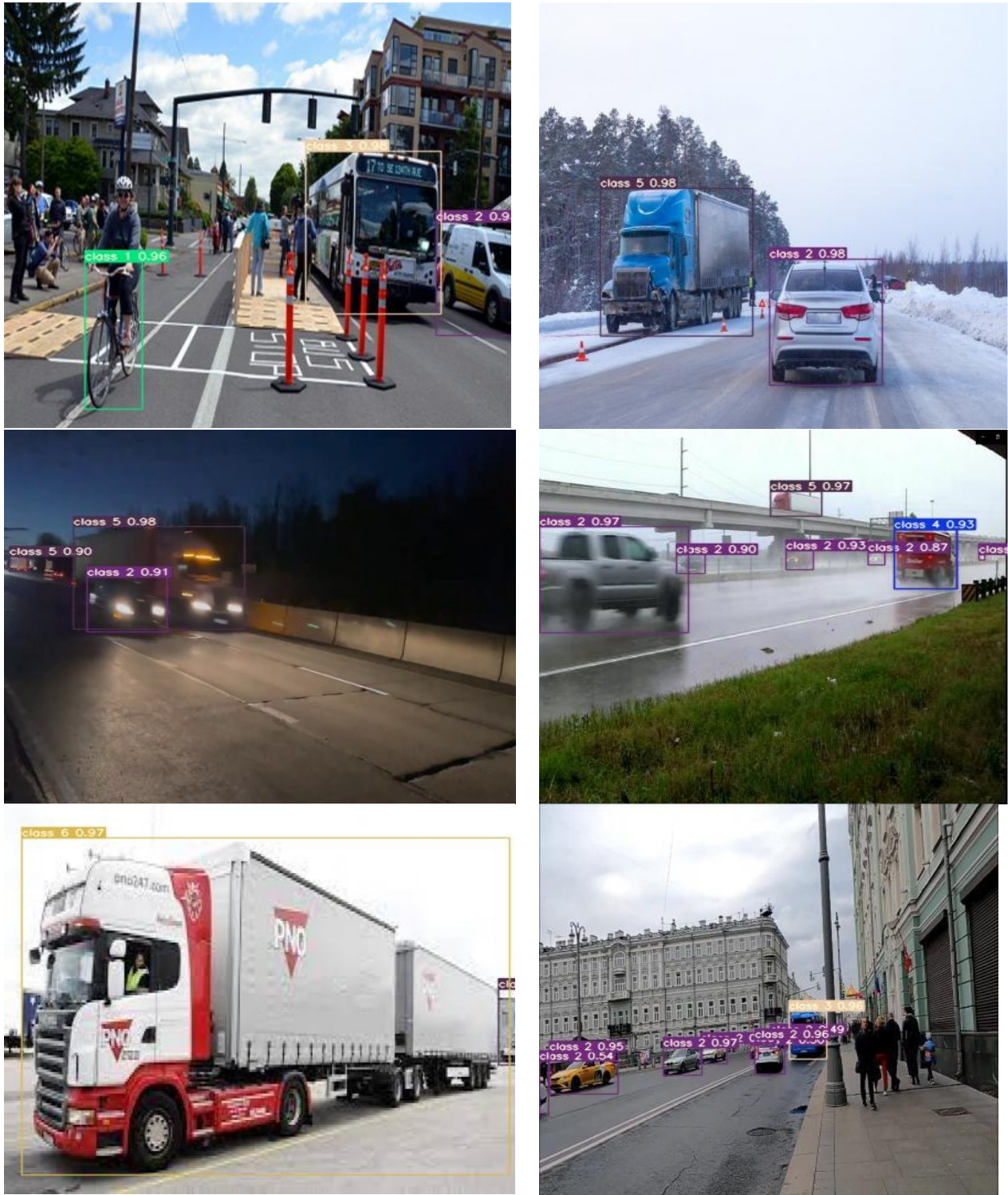


Figure A1. YOLOv7 vehicle detections

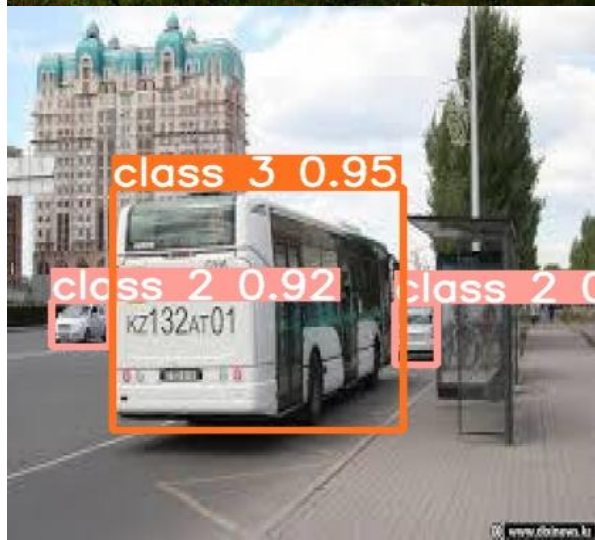


Figure A2. YOLOv5s vehicle detection

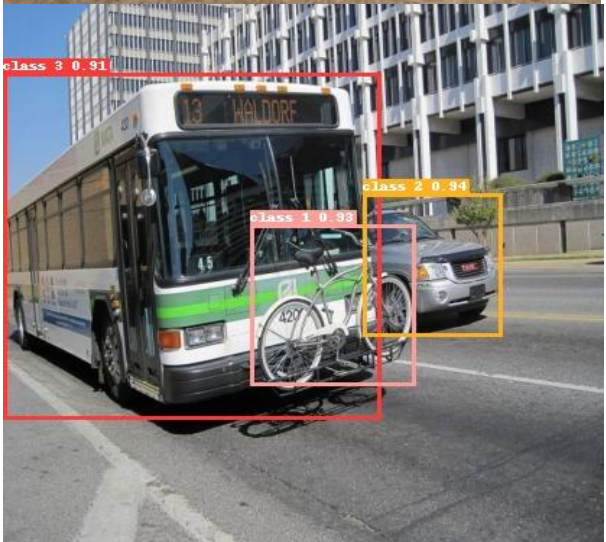
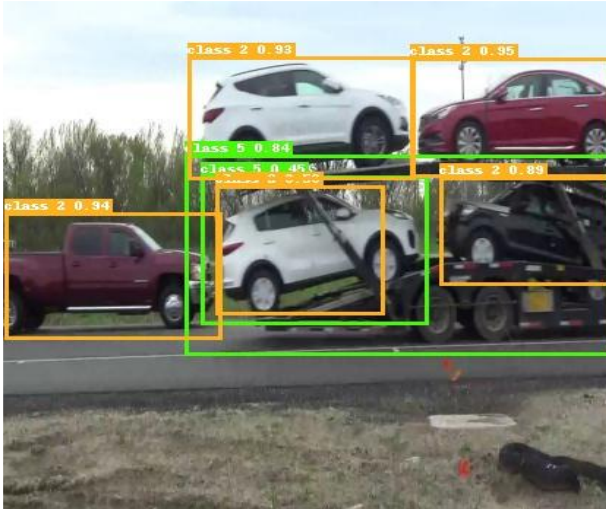


Figure A3. YOLOv5m vehicle detections