IMPROVED VEHICLE DETECTION: UNVEILING THE POTENTIAL OF MODIFIED YOLOv5

A thesis submitted to the Department of Electronics and Communication Engineering in partial fulfillment of the requirements for the degree of Master of Science in Electronics and Communication Engineering

Submitted by

Md. Milon Rana

Student ID: 1602141

Session: 2021

Supervised by

Professor Md. Mahabub Hossain, Ph.D.

Department of ECE, HSTU

Department of Electronics and Communication Engineering

Hajee Mohammad Danesh Science and Technology University (HSTU)

Dinajpur-5200, Bangladesh.

2023

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

FACULTY OF POST GRADUATE STUDIES

HAJEE MOHAMMAD DANESH SCIENCE AND TECHNOLOGY UNIVERSITY, DINAJPUR-5200, BANGLADESH

CERTIFICATE

This is to certify that the work entitled as **"IMPROVED VEHICLE DETECTION: UNVEILING THE POTENTIAL OF MODIFIED YOLOv5"** written by Md. Milon Rana has been carried out under our supervision. To the best of our knowledge, this work is an original one and was not submitted anywhere for a reward or a degree.

Supervisor

……………………..

Dr. Md. Mahabub Hossain

Professor

Department of Electronics and Communication Engineering.

Hajee Mohammad Danesh Science and Technology University, Dinajpur-5200, Bangladesh.

Co-Supervisor

………………………..

Md. Selim Hossain

Lecturer

Department of Electronics and Communication Engineering.

Hajee Mohammad Danesh Science and Technology University, Dinajpur-5200, Bangladesh.

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
FACULTY OF POST GRADUATE STUDIES

HAJEE MOHAMMAD DANESH SCIENCE AND TECHNOLOGY UNIVERSITY,
DINAJPUR-5200, BANGLADESH

DECLARATION

The work entitled **"IMPROVED VEHICLE DETECTION: UNVEILING THE POTENTIAL OF MODIFIED YOLOV5"** that has been carried out in the Department of Electronics and Communication Engineering, at Hajee Mohammad Danesh Science and Technology University is original and conforms the regulation of this university. We understand the university policy on plagiarism and declare that neither this thesis nor any part of this work has been used or submitted elsewhere for any kind of degree or award.

…..…………………………
Md. Milon Rana

Student ID: 1602141

Session: 2021-2022

The thesis titled **"IMPROVED VEHICLE DETECTION: UNVEILING THE POTENTIAL OF MODIFIED YOLOv5"** submitted by Md. Milon Rana, Student ID: 1602141 and Session 'January-June' 2023, has been accepted as satisfactory in partial fulfillment of the requirement for the degree of Master of Science in Electronics and Communication Engineering.

BOARD OF THESIS EXAMINERS

Professor Dr. Md. Mahabub Hossain Department of ECE, HSTU, Dinajpur.	Chairman
Professor Md. Mehedi Islam Department of ECE, HSTU, Dinajpur.	Examiner (Internal)
Md. Abubakar Siddik Assistant Professor Department of ECE, HSTU, Dinajpur.	Examiner (Internal)
Professor Dr. Sajjad Waheed Department of ICT, MBSTU, Santosh, Tangail-1902.	Examiner (External)
Professor Dr. Md. Maniruzzaman Department of ECE, Khulna University, Khulna.	Examiner (External)

Acknowledgment

At first, I would like to express my gratitude to my understanding supervisor, Professor Dr. Md. Mahabub Hossain, for his unwavering support, guidance, and direction in starting and completing this research within the given timeframe. Throughout the research process, his office door was always open, providing me with a platform to share any positive experiment results. He offered continuous assistance and guidance, steering me in the right direction. Without his invaluable help, I could not have completed this thesis.

I also want to thank Lecturer, Md. Selim Hossain for his extensive knowledge in information theory and his valuable input on my future research and personal growth. Working with him has been a great privilege, and his guidance and suggestions have been immensely beneficial to me. I am grateful for his help and consider myself fortunate to have had the opportunity to work with such an esteemed professional.

At Last, but not least, I would like to express my deep appreciation to my parents and unaffected family for their unwavering support and encouragement throughout my entire life, especially during my years of study. Their constant support has been instrumental in my achievements, and I owe this accomplishment to them. Without their love and guidance, this would not have been possible.

List of Contents

List of Figures

LIST OF ABBREVIATIONS

CNN:	Convolutional Neural Networks
KNN:	K Nearest Neighbors
LR:	Linear Regression
ML:	Machine Learning
MLP:	Multi-Layer Perceptron
RELU:	Rectified Linear Unit
RMSE:	Root Mean Squared Error
MSE:	Mean Squared Error
MAE:	Mean Absolute Error
AI:	Artificial Intelligence
SVR:	Support Vector Regression
SVM:	Support Vector Machine
OCR:	Optical Character Recognition
FTS:	Fuzzy Time Series
IoU	Intersection over Union
AP	Average Precision
YOLO	You Look Only Once
YOLOv5s	You Look Only Once Small
YOLOv5n	You Look Only Once nano
YOLOv5l	You Look Only Once large
YOLOv5m	You Look Only Once medium

Abstract

The detection of vehicles is a crucial task in various applications. In recent years, the quantity of vehicles on the road has been rapidly increasing, resulting in the challenge of efficient traffic management. To address this, the study introduces a model of enhancing the accuracy of vehicle detection using a proposed improved version of the popular You Only Look Once (YOLO) model, known as YOLOv5. The accuracy of vehicle detection using both the original versions of YOLOv5 and our proposed YOLOv5 algorithm has been evaluated. The evaluation is based on key accuracy metrics such as precision, recall, and mean Average Precision (mAP) at an Intersection over Union (IoU). The study's experimental results show that the original YOLOv5 model achieved a mean Average Precision (mAP) of 61.4% and the proposed model achieved an mAP of 67.4%, outperforming the original by 6%. The performance of the proposed model was improved based on the architectural modifications, which involved adding an extra layer to the backbone. The results reveal the potential of our proposed YOLOv5 for real-world applications such as autonomous driving and traffic monitoring and may involve further fine-tuning, robotics and security system and exploring broader object detection domains.

CHAPTER – 1

INTRODUCTION

The provided response emphasizes the commencement of the topic, providing an inclusive introduction that covers the aim, importance of the research, and an overview of the study.

1.1 History

1.2 Problems

1.3 Goal of the Work

1.4 The Work's Significance

1.5 The Work's Limitations

1.6 The Objectives

1.7 Overview of the Work.

INTRODUCTION 1

The section explores the rapid progress of machine learning in object detection, focusing on powerful convolutional neural networks. These networks excel in identifying objects and detecting vehicles. The summary highlights the significance of the models, the encountered challenges, and the study's core objectives.

1.1 History

The number of vehicles has increased sharply, leading to congestion on highways, national highways and roads. To control the huge amount of traffic systems are essential for collecting and managing information, making traffic more serious, authentic and safe. For monitoring, planning and controlling traffic flows, moving vehicles are more important to detect, seek and calculate[1].With the help of induction loop detectors, infrared detectors, using radar detectors or real-time video-based detection methods[2],Detection vehicle can be done. The field of vehicle detection and classification offers a range of techniques, including SVM, CNN[3] Decision Tree, and RNN. With computer vision being a key industry focus, this area experiences constant evolution. In this thesis, the focus lies on investigating different YOLOv5 algorithms to determine their suitability and identify the most effective model for vehicle detection and classification. Video-based detection systems are more sensitive to weather, light and shadow conditions than other detection techniques. However, slow and undisturbed speed of other techniques video-based observation systems provide many advantages like easy installation, and convenient modification, so researchers are more interested in video-based detection systems [4,5,6,7]. Using convolutional neural networks (CNNs) , object detection algorithms have been used recently with satisfactory results for vehicle detection. In addition, some methods like two-stage recursive CNN, single-stage SSD detector, faster R-CNN and other algorithms which is YOLO ,were used for media detection [8,9,10].

However, in practice, the application of the above methods does not yield satisfactory results due to the requirements of large hardware volume and computation time. Several type of parameters and actions require in this Conventional CNNs. An example, VGG-16 requires approximately 138 million operations and 14.9 B operations for image processing [11,12,13]. However, devices like smartphones with limited memory and processing speed cannot be used for a large network. Therefore, it is very important to reduce the computation time, wiring requirements as well as the number of network parameters. Recently, YOLO v5 single-stage detector has been used with higher efficiency for low computational requirement and processing fast. [14].For this, classification of vehicle is important because as per their class for the analysis are counted and segregated by the vehicles. Here , six categories of vehicles are used for detection. Vehicles that can be detected here are given in Table 1.

Table 1. Classes and types of vehicle.

Vehicles Type	Vehicle Class
Car	I
Bus	II
Ambulance	III
Bicycle	IV
Truck	V
Motorcycle	VI

Having embarked on the quest to find the most suitable algorithm for vehicle detection, we set our sights on the YOLO family instead of Fast RCNN. The YOLO algorithms, renowned for their status as single-stage object detectors, offered a distinct advantage - their remarkable speed, surpassing that of Faster R-CNN. As our primary focus centered on detecting vehicles, whose size rendered accuracy paramount, YOLO's swiftness without compromising precision made it a compelling choice [15].

In our pursuit of the ultimate YOLO version, a comparison unfolded between YOLOv3, YOLOv4, and YOLOv5 using our meticulously curated dataset. This rigorous evaluation encompassed consistent input and output parameters, with accuracy and loss serving as the essential metrics. The outcome was crystal clear: YOLOv4 and YOLOv5 showcased superior performance in terms of accuracy compared to YOLOv3. Consequently, our research journey embarked on the vehicle identification and classification path, leveraging the potent capabilities of the YOLOv5 algorithm. Below figure 1 shows a general system of vehicle detection using YOLOv5. In this research, our proposed YOLOv5 and YOLOv5s algorithms have been implemented for more efficiently vehicle detection and improved multimedia detection, responsiveness requirements, and real-time query.

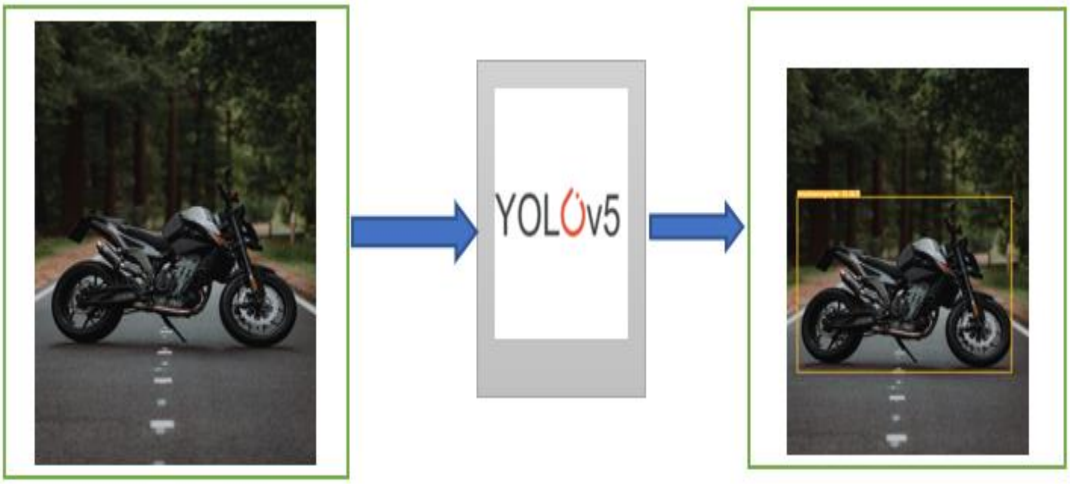

Figure 1. A general system of vehicle detection using YOLOv5.

1.2 Problems

In the realm of research, object detection has emerged as a focal point, with an intense focus on achieving remarkable levels of accuracy. Leveraging the power of machine learning, various techniques have been proposed to tackle this challenge. The primary goal of this thesis is, making its selection based on accuracy and performance as key criteria in our proposed methods of YOLOv5. An ideal fit for machine learning is the task of detecting and classifying objects in images and videos giving the dataset as the intricate and multifaceted features. On traffic and detection vehicle, the system's significance extends with a particular emphasis across a diverse array of applications.

This investigation involved comprehensive training and meticulous analysis on a detailed dataset, extensively discussed in Chapter 4. The study aimed to gain a deeper understanding of the subject through a determined pursuit of knowledge and discovery.

1.3 Goal of the Work

The purpose of work is evaluated the performance of proposed YOLOv5 and original YOLOv5 algorithms, Though there are several classes of YOLOv5 like YOLOv5s (small), YOLOv5n (nano), YOLOv5l (large), YOLOv5m (medium), and YOLOv5x (the largest of the five), in detecting vehicles in images and videos. Here ,We modify the original YOLOv5s models .The evaluation will be done using the same dataset and preprocessing methods for all models, and the algorithm with the highest accuracy and performance will be recommended. The thesis will implement six classifiers that can predict the class of an image, such as Bicycle, Car, Bus, Truck, Motorcycle, and Ambulance. In addition detecting vehicles purpose, the vehicle detector will be proficient in precisely determining the bounding box coordinates encapsulating each identified vehicle.

1.4 The Work's Significance

With the advent of the industrial revolution, the number of vehicles on roads has witnessed a steady surge, posing new challenges in traffic surveillance. In bustling cities, people often find themselves trapped in traffic congestion, underscoring the critical need for a digitized traffic system that operates 24/7 and enhances overall efficiency worldwide. Such a computerized traffic system hinges on the deployment of a highly accurate vehicle detection system, given its far-reaching impact on the economy, citizens' daily lives, and various industries.

As we strive towards a more advanced future, it remains imperative to continually contribute to the evolution of this field. The pursuit of algorithms capable of accurately detecting all types of vehicles, be it in images or videos, through well-developed pipelines, is of utmost importance. By persistently refining and advancing vehicle detection technologies, we can pave the way for a seamless and productive traffic ecosystem that benefits societies worldwide.

1.5 The Work's Limitations

Although the thesis has successfully accomplished its objectives, there are opportunities for further enhancement. Improving the dataset by augmenting its quality and increasing the diversity of objects within the vehicles class could lead to even more precise results. Additionally, to provide a more thorough and suitable analysis, conducting evaluations across a broader spectrum of algorithms commonly employed in computer vision challenges would have been beneficial. By addressing these aspects, future research can unlock even greater potential in the realm of object detection and classification.

1.6 The Objectives

The work's main goal is to improve the proposed YOLOv5 for detecting and categorizing vehicles in any view images. Additionally, our aim to test the models in different settings, measure their accuracy, and analyze their loss when working with limited resources. The research methodology is based on an exploratory approach, using a quantitative method to assess the models' performance and loss. The study will be conducted using well-known datasets. The work’s contributions of the study are:

- Creation of models that can identify and classify multiple vehicles in any view digital images
- In terms of accuracy and loss, an evaluation and discussion occurred of the proposed models
- A comparison of proposed methods with similar methods.

1.7 Overview of the Work

The work is comprised of six chapters:

Chapter -1: The study starts with a thorough introduction and background, clarifying the research's aim and significance. Additionally, it offers an overview of the entire study, providing a solid foundation for further exploration.

Chapter -2: Reviews past related research, including their techniques and results.

Chapter- 3: Focuses on machine learning techniques, including the theory, formulas.

Chapter- 4: Thoroughly explains the machine learning algorithms and models used in the thesis, including their underlying formulas.

Chapter -5: Presents the results and discussion.

Chapter - 6: Offers recommendations for future improvement of the research topic.

CHAPTER – 2

RELATED WORK OF VEHICLE DETECTION

The study ventures into past related research, exploring their techniques and findings to shed new light on the subject at hand.

2.1 Specialized Vehicle Detection Methods

2.1.1 Vehicle Detection in Faster R-CNN

2.1.2 Vehicle Detection using features map and sliding window

2.1.3 Vehicle Detection using YOLOv3

2.1.4 Vehicle Detection in thickness estimation

2.1.5 Vehicle Detection using SSD algorithm

2.1.6 Vehicle Detection using YOLO

2.1.7 Harris corner detector for Vehicle Detection

2.1.8 Semi-supervised convolutional network for Vehicle Detection

2.1.9 Vehicle detection using (HOG) and (SVM)

2.2 CNN based Vehicle Detection Methods

2.2.1 Convolutional Layer

2.2.2 Pooling Layer

2.2.3 Activation Layer

2.2.4 Connected Layer

2.2.5 CNN Output Layer

2.3 Summary of object detection variants using YOLO

RELATED WORK OF VEHICLE DETECTION 2

The literature review chapter provides a comprehensive understanding of the current state of vehicle detection, focusing on various CNN-based models and their performance. Notable models like Faster R-CNN, SSD, and YOLO have achieved remarkable results on benchmark datasets. The review highlights the significance of object detection models and their potential for further improvement, serving as a valuable resource for researchers and practitioners in the field.

2.1 Specialized Vehicle Detection Methods

In our exploration of the topic of vehicle detection and classification using deep learning, we carefully examined multiple research papers, specifically those employing YOLO algorithms. This review allowed us to gain a deeper understanding of the progress and contributions made by various researchers in the field. Below, we provide a condensed summary of the key ideas and notable aspects found in these past research papers.

2.1.1 Vehicle Detection in Faster R-CNN

Vehicle detection is a modern technology that involves identifying vehicles in digital images or videos in computer vision. This problem is commonly used to determine the presence of a vehicle and recognize its features, such as size and position within the image. The ability to detect vehicles is crucial for traffic situation estimation algorithms. vision Related to vehicle detection technology there have been numerous research efforts and developments in the field of computer. Detection vehicle operates a crucial role in identifying or accurately categorizing vehicles, as well as precisely locating their positions.[16] Object detection has been a significant area within detection vehicle specifically the most fundamental task within detection object, as the varying appearance and conditions of vehicles during the detection process.[17]Three main steps construct a traditional object detection algorithm: (i) As a candidate area sliding window frames use different sizes and scales to scroll the input image at a certain step length; (ii)In each candidate area feature extraction performs on local information; Based on color, texture, shape, and some middle- or high-level semantic features [18,19] In this context, the final output of the algorithm corresponds to the most recently detected target. (iii) For identification classifiers are used, as Support Vector Machine (SVM)models. For example, the traditional histogram-oriented gradient method has been used to detect vehicles by extracting features from multimedia images. For vehicle classification, CNN is used to perform feature extraction, and Soft-max function and (SVM) can also be used. To increase accuracy an updated R-CNN and a new fast R-CNN model were used. [8,9,10]. All the above models implement a selective search algorithm to select a little number of significant areas among all areas of the total image. The image classification algorithm requires a small number of regions to make the model faster. A large number of candidate regions compared to faster and faster R-CNN, slowing down its computation in conventional R-CNN[12,21].

The utilization of a clustering layer within the region of interest (ROI) along with fast R-CNN addresses the challenge of efficiently managing a substantial number of candidate regions. Compared to regular R-CNN, fast R-CNN significantly reduces the number of candidate regions, resulting in improved efficiency. As a result, low computation time is required and requires wiring. However, for real-time vehicle detection, R-CNN is not an efficient algorithm. All of these models require considerable computation time and complex models with several of parameters. YOLO-v5 relies on intelligent CNN for the detection of objects. This algorithm takes the input image, converts it into regions, and then calculates bounding boxes and probabilities for each of these regions. The predicted probabilities associated with the bounding boxes are then utilized. The algorithm needs to go through the neural network to make a prediction in only "looks once" at the image. After non-maximum suppression, which ensures that the object detection algorithm recognizes each object only once at a time, this output is called bounding box objects [22].

2.1.2 Vehicle Detection using features map and sliding window

Vehicle detection, also known as computer vision object recognition, involves employing scientific methods and techniques to enable machines to perceive the environment like humans. The primary goal of a vehicle detection system is to identify one or more vehicles present in input images. The local features-based approach begins by locating the features of individual objects or groups of objects, which are then classified into various classes using classification models. This allows the system to determine the object's category. However, a potential weakness of this method is that it can only detect pre-learned objects. Conversely, the sliding window-based system operates differently. It scans the input image using multiple windows of different sizes and assesses whether the target (vehicle) is present within each window or not. This method provides an alternative way to detect vehicles in the images.[23]

2.1.3 Vehicle Detection using YOLOv3

For real-time object detection, YOLOv3 has been identified as the most precise incremental algorithm, as described in [24]. In [25],Combined YOLOv3 and k-means were used in 3D images for detecting objects in a LIDAR camera which is captured for an autonomous driving system . In [26], an improved version of YOLOv3 was developed, which increased speed while retaining accuracy. A discarding technique using YOLOv3 and tiny YOLOv3 combined and add an extra layer to this improvement was achieved, eliminating the need for training different scales and different configurations algorithms and weights. Our research utilized YOLOv5, preparing a dataset to cover both non-exempted and exempted vehicles for vehicle identification and classification and comparing results. Another study, [27], used YOLOv3 and tiny-YOLOv3 to classify passenger vehicles and trucks/buses by observed a recall of 100% and taking images in a perpendicular camera .However, all the non-exempted classes of vehicles for a toll management system, they did not consider. Our study, on the other hand, considered all vehicle classes and compared the accuracy of traditional techniques that of our classification results.

This technique uses YOLOv3 for image processing and detect several types of vehicles. A real-time detection object techniques which is YOLOv3 algorithm that can identify specific objects in images, videos, or live feeds. It provides better results for rotating or small objects compared to other methods. Using YOLOv3, we can accurately count the high density of vehicles at a fast pace. Based on Darknet which is an open source neural network system in YOLOv3 ,that supports computation both CPU and GPU. Initially, Darknet has a 53-layer network trained on Imagenet. To perform object detection, more layers are added to create a fully convolutional architecture for YOLOv3.

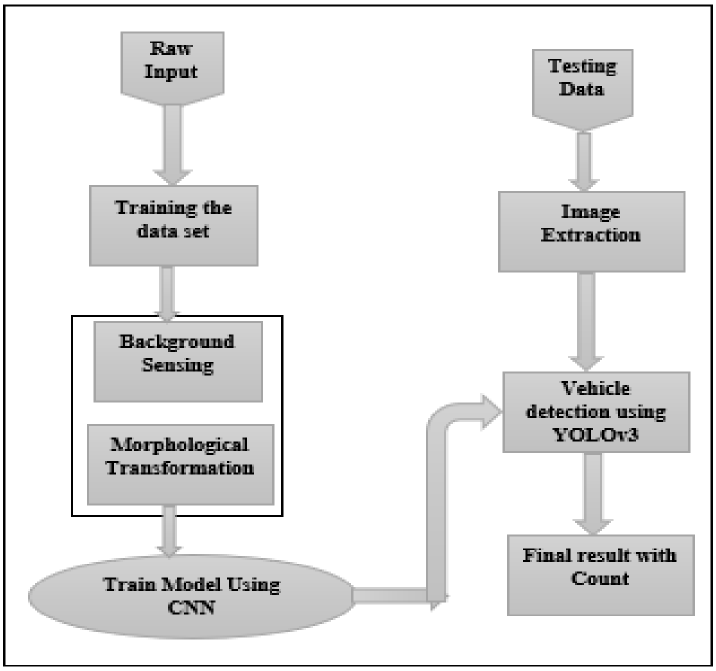

Figure 2. Vehicle Detection using yolov3.

2.1.4 Vehicle Detection in Edge detection

An approach to vehicle detection [28] was proposed based on thickness estimation. The angle vectors were determined of the trained images in the edge guidance. Researchers proposed a method to detect vehicles by analyzing the changes in inclination vectors' headings at the object boundaries and calculating the standard deviation of these slope vectors. This approach allows for the recognition of vehicles based on predefined edges. However, one limitation of these detection methods, which lack prior training, is their difficulty in distinguishing target objects from complex background conditions. Another different approach [29] introduces a vehicle recognition technique featuring an enhanced basic extractor. This methodology utilized an 8-block infill asphalt split method to eliminate road markings, resulting in a more random subdivision. The results indicate that this proposed strategy effectively eliminates identification errors caused by inaccurate distance markers. However, it should be noted that this recognition approach is based on the detection of a stationary camera, which contradicts the underlying principle of this theory but still allows for the utilization of basic extraction systems. This distinctive algorithm stands out from others by its ability to detect objects of varying scales at multiple layers of the network, in contrast to conventional methods that perform detection solely at the top level. By leveraging a wider range of features, it achieves more accurate and comprehensive object detection capabilities.

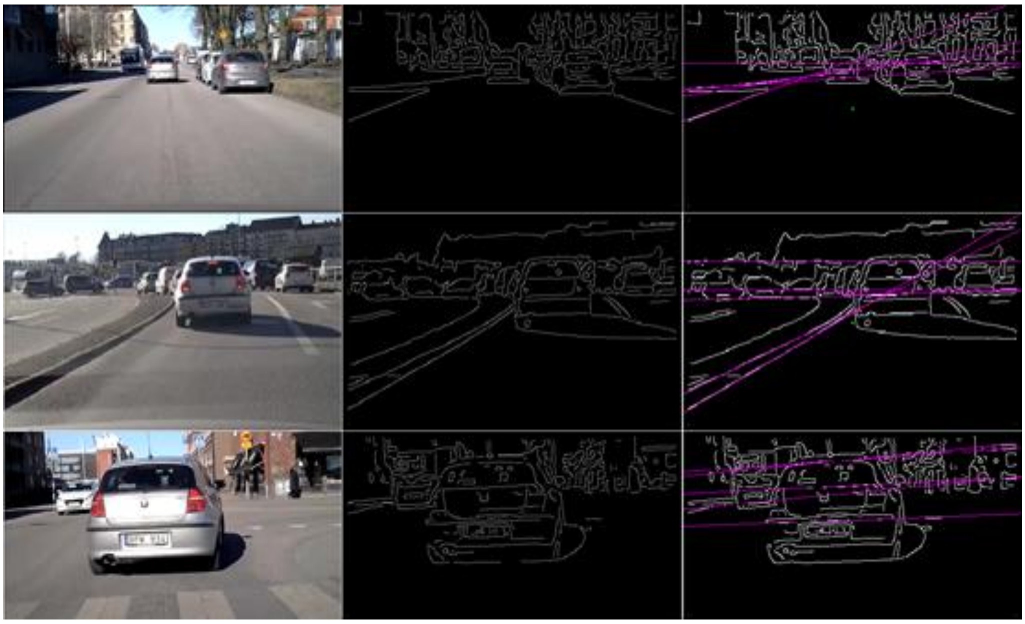

Figure 3. Edge Detection

2.1.5 Vehicle Detection using SSD algorithm

While the SSD algorithm can detect different objects at different levels in the netwk, other algorithms [16] only run the detection at the top layer of the network. The SSD algori thm has undergone many improvements since its release. It is recognized as one of the best singlelevel search products after YOLO, with improvements in two main areas. First, speciy the scale and contrast of the initial target detection and the precheckbox equivalent to the ju nction box in Faster RCNN. Secondly, multiclassification is done using custom map output from CNN to make classification estimation of different parameters.It should be noted that t he accuracy of the traditional method is lower when the car is on the road. To solve this pro blem, we suggest using VGG,SSD (Single Shot Multibox Detector) to detect vehicles. The f ramework of the vehicle counting algorithm is shown in Figure 4. Vehicle movement is mo nitored by comparing vehicle blobs and calculating the minimum distance between successi ve blobs. In the final stage, we use the virtual coil technique to calculate the traffic based on the monitoring results.

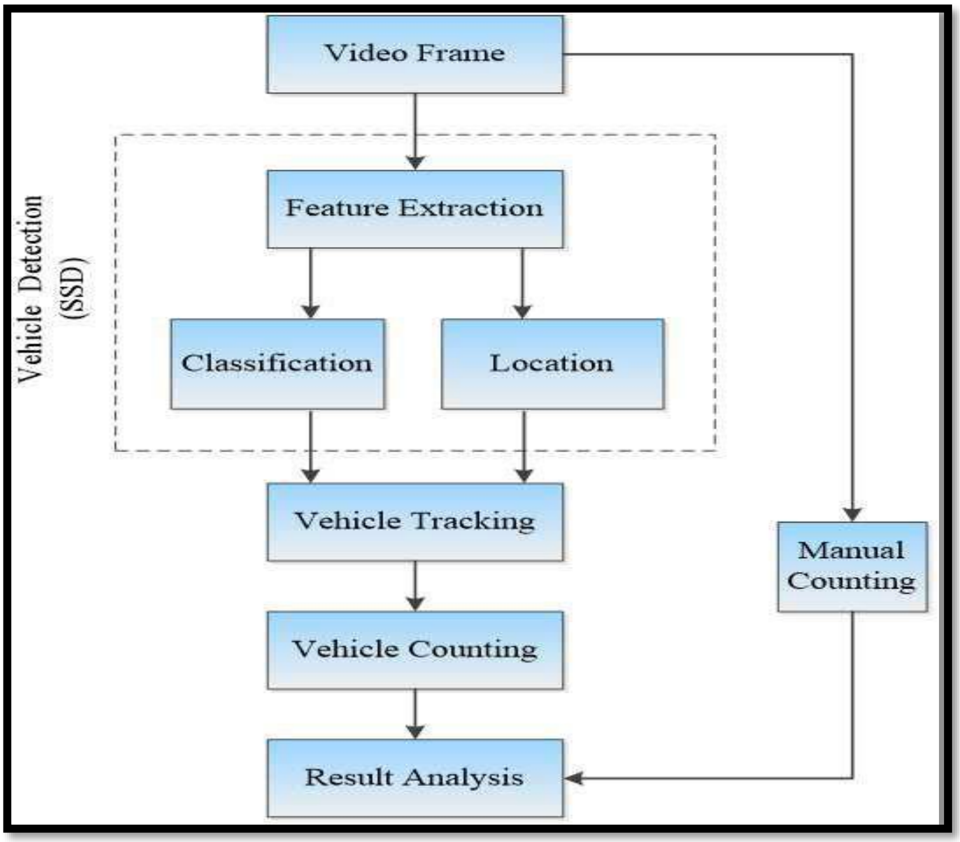

Figure 4. Vehicle counting algorithm Flowchart.

The proposed vehicle counting system offers the following advantages:

- Even with relatively low-resolution to the input images the output is in high accuracy and speed.
- It is related to the original image data, without complex preprocessing.
- It is highly robust and suitable for images of various sizes.
- It is easy to maintain.

2.1.6 Vehicle Detection using YOLO

YOLO, the first single-stage detector, was known for its speed, reaching up to 155 fps. However, it struggled with detecting small objects and had issues with accuracy. CNN which is applied to entire images in YOLO, predicting the direct regression of output features target through classifications and bounding boxes. Over time, several YOLO improvements, leading to the introduction of YOLO9000[30] and YOLOv3[31], which improve the loss function to incorporate FPN and other structures. Similarly YOLO v4[32] the time of 2020, was released, incorporating a range of new backbone network and a data enhancement techniques combinations. That used a combination of FPN and the PAN network ,the Mish activation function, Drop block prevention policies, and develope in the bounding box loss function CIOU_Loss. With an accuracy of 10% higher on using the dataset COCO compared to YOLO v3, Version 5 of YOLO[33] released by Jocher Glenn in same year. Despite its unofficial recognition, it saw further improvements in performance.

In recent years, from both academia and industry the development of YOLO series highlights the significant interest and expectations in this algorithm. Many researchers in the field of detection object based on YOLO [34], focus on improving the fine-tuning algorithm, raising new trends and challenges in the field. Also, the issue of little protest for location is analyzed methodically, presenting calculations over the five perspectives of multi-scale, highlight determination change, relevant semantic data, information upgrade, and organize optimization. The characteristics of calculations and systems are compared. For interstate applications, this ponder assist makes strides discovery precision by utilizing the Flip-Mosaic calculation and the made strides YOLOv5 calculation for observing video streams [35]. A multi-type vehicle target dataset was created for different climate conditions and scenarios, and a real-time location demonstrate for vehicle category data was gotten from the dataset. This consider has imperative suggestions for administering expressway operations and progressing the administration administrations of transportation divisions.

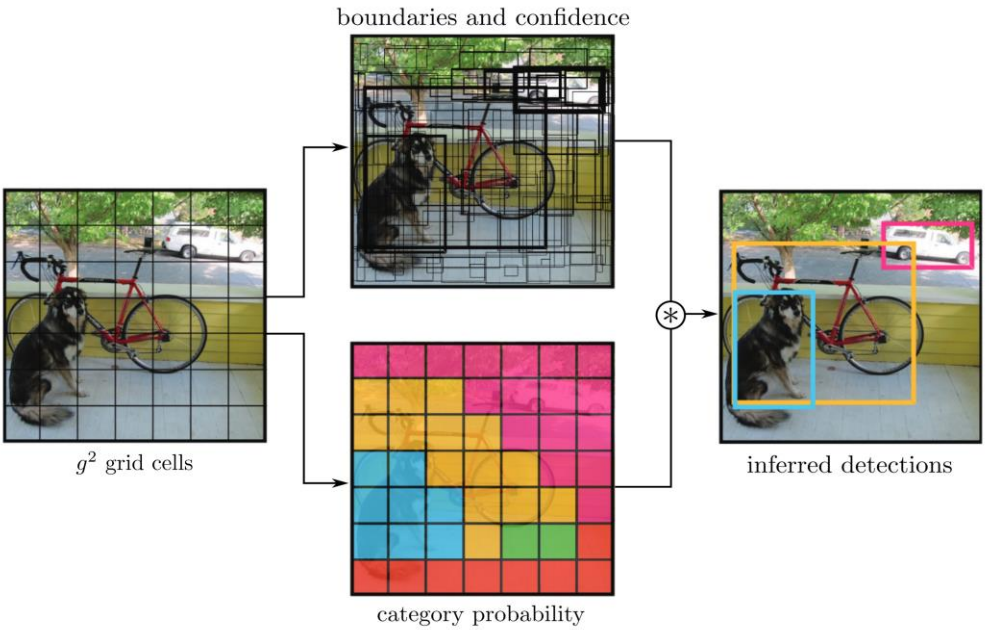

Figure 5. A simplified schema of the YOLO model.

2.1.7 Harris corner detector for Vehicle Detection

Focused on the problem in rural environments of automated detection vehicle [36] and its solution made of two phases: a location calculation and a classification stage. Harris corner detector which is the first stage of the solution, is used to identify in the images key features, as it argues that cars or buses tend to have more edges and corners than natural objects. Then, it employes a approach which is sliding window, to detect high feature density with regions. Based on the color information, these regions are further processed to the background. To determine the presence of a vehicle in resulting regions where image classification algorithms are applied and then fed into the second stage. The creators assessed the execution of two picture highlight descriptors, the altered Histogram of Situated Points (Hoard) and Histogram of Gabor Coefficients, as well as three factual classification strategies: K-Nearest Neighbors (k-NN), Arbitrary Woodlands (RF), and Back Vector Machines (SVM). It is found that this approach had an normal area exactness of 85% and the finest classifier was the Irregular Timberland utilizing Histogram of Gabor Coefficients highlights, which accurately distinguished 98.9% of the vehicles and 61.9% of the foundation pictures.

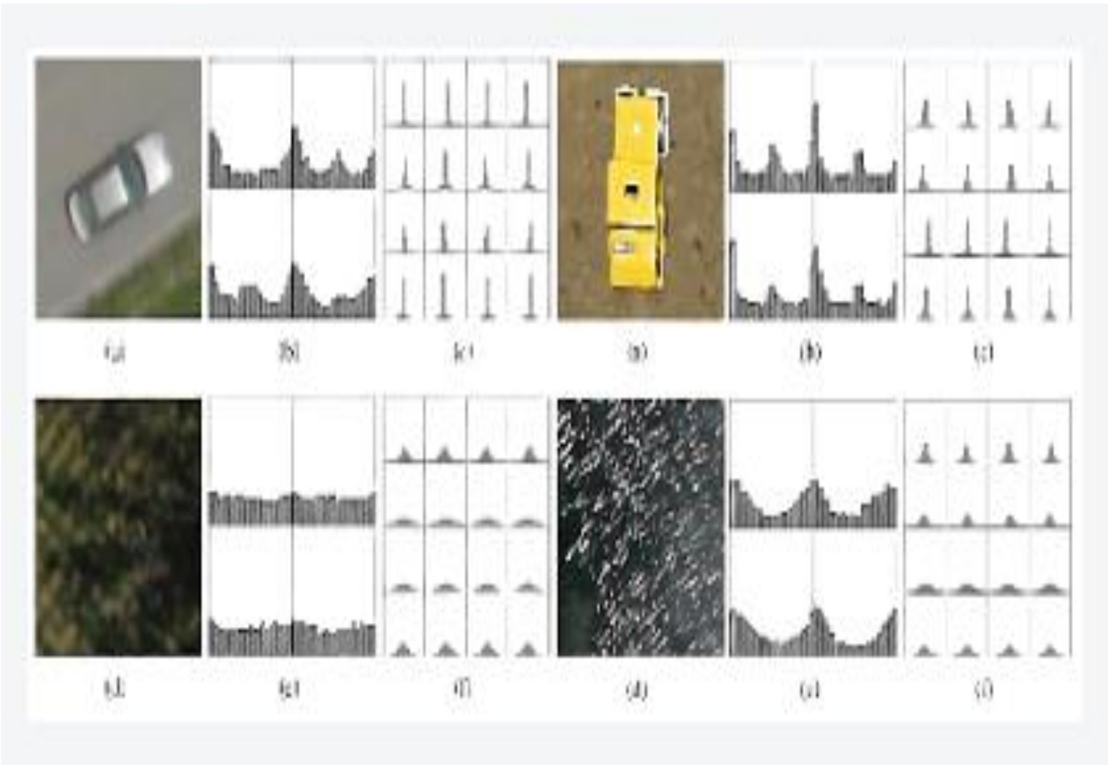

Figure 6. Background image with HoG features

In recent times, considerable attention has been directed towards vehicle recognition in digital images within the computer vision research community. Nevertheless, a pressing challenge arises due to the lack of a standardized evaluation framework, resulting in significant performance variations and hindering unbiased result comparisons. A pioneering approach in vehicle detection leverages [37] Haar type features, effectively identifying vehicles in images and effectively eliminating noise during boundary detection. Despite its success in car identification, this method falls short in classifying vehicles. Remarkably, the model achieved an impressive extraction time of just 66 ms on the latest embedded system.

2.1.8 Vehicle Detection in Semi-supervised convolutional network

Introducing the groundbreaking "Semi-Controlled Convolution Organization" designed specifically for the classification of vehicles using front-end images sourced from the esteemed BIT Tools Dataset. This cutting-edge imaging project unfolds in two distinct convolution phases, each encompassing convolution, nonlinear maximal correction, neighborhood difference normalization, and normal pooling. What sets this approach apart is the integration of a master plan that takes the input image and feeds it into the current plan, culminating in a seamless combination of two organizations, ultimately enabling the identification of six distinct vehicle categories: transport, van, sedan, SUV, and truck.

Pushing the boundaries of accuracy, [38] achieved an impressive 88.11% by skillfully leveraging Laplace channels and a rich repository of unsigned data to initialize the network core. Building on this momentum, [39] proposed an ingenious multimodal traffic monitoring implementation, employing a convolutional system complete with a pre-boot and post-process voting framework. This demonstration incorporates seven free convolutional machines, each endowed with matching features that predict outcomes through a dynamic voting process. Put to the test on an ImageNet dataset housing 11 diverse classes, including trucks, bases, bicycles, transportation, cars, bicycles, non-cars, pedestrians, lorries, monocoque trucks, and commercial vehicles, this approach garnered an extraordinary accuracy rate of 97.84%.

2.1.9 Vehicle detection using (HOG) and (SVM)

Exploring the realm of vehicle recognition in images, a captivating study compared the performance of two models: the HOG + SVM and Tiny YOLO[40]. The HOG + SVM model showcased the prowess of combining the Degree Histogram (HOG) feature descriptor with the Support Vector Machine (SVM) classifier, securing an impressive average accuracy of 97%. In contrast, Tiny YOLO, a streamlined version of the renowned YOLO architecture [41], exhibited its finesse in carrying out both detection and isolation tasks.

The research centered on the TPS dataset, where vehicles were categorized into buses, minivans, vans, trucks, and cars. Remarkably, the HOG + SVM model proved its mettle with a 14% recovery rate and an 87.81% return rate for the TPS dataset. Meanwhile, Tiny YOLO demonstrated commendable results, achieving a 62.83% accuracy, an 86.22% return rate, and an Intersection over Union (IoU) score of 69.

Another noteworthy exploration [42] delved into car type classification in videos, adopting a multimodal approach that ingeniously incorporated video and audio features. The utilization of the AlexNet algorithm for image feature extraction and Mel Frequency Cepstral Coefficients (MFCC) for audio features, along with a support vector machine (SVM) classifier, paved the way for accurate classification of vehicles into distinct types like armored vehicles, construction vehicles, cranes, emergency vehicles, military vehicles, motorcycles, and rescue vehicles. The culmination of this effort yielded an impressive accuracy of 72.1%.

Venturing into the realm of real-time object detection, Project YOLO, aptly named "One Look Only," pioneered the use of a convolutional neural network (CNN) [43]. Unlike conventional methods, YOLO directly predicted object boundaries and classes without resorting to regional proposals, gaining an edge in efficiency. Through a convolutional process, the input image was transformed into a feature map, facilitating object detection. The ingenious use of 1 x 1 convolutions for classification and regression led to the representation of the completion map as a tensor t with dimensions g2 x a x (5 + q), wherein g2 symbolized grid lines, a stood for anchor boxes, and q accounted for the relevant predictions of the model, cementing YOLO's pioneering concept in encoding perception.

2.2 CNN based Vehicle Detection Methods

A Convolutional Neural Network (CNN) stands as a remarkable variant of the feedforward neural network, meticulously tailored to excel in image processing. The architecture of a CNN [44] comprises an input layer, an output layer, and multiple hidden layers, among which reside the fundamental components: convolutional layers, activation layers, and pooling layers. CNNs represent a powerful class of deep learning models, widely acclaimed for their prowess in image recognition, object detection, and various other computer vision endeavors. To unravel the essence of a CNN's design, it can be effectively segmented into the following distinctive layers:

Input layer: This layer takes the image for input and passes it on the next layer.

Convolutional layer: The Convolutional Layer serves as the initial stage, employing filters (also known as kernels) to scan the input image, effectively detecting various features like edges, lines, and textures. Each filter traverses the entire image, performing its designated function. The outcome is a distinctive set of feature maps, highlighting variations within the input images.

Activation layer: The Workflow Layer introduces nonlinearity to the model, incorporating an activation function (such as ReLU) to process the output from the convolutional stage, generating a fresh set of feature maps.

Pooling layer: The pooling layer reduces the size of the feature maps by down sampling them. Subsequently, the Pooling Layer steps in, down sampling the feature maps to reduce their size. This down sampling aids in diminishing the number of parameters within the model, enhancing its resilience to positional changes in the input image.

Fully connected layer: The Fully Connected Layer establishes a vital connection, taking the output from the preceding layer and mapping it to the class scores. This layer operates similarly to the joining operation in a conventional neural network

Output layer: Ultimately, the Output Layer delivers the final model output, presenting the predicted class of the input image.

These layers synergistically combine to form a deep neural network. In addition to these core layers, CNNs can also incorporate other types of layers, such as output layers, batch normalization layers, and crosslinking, to further enhance the model's performance. A comprehensive exploration of each layer follows.

2.2.1 Convolutional Layer

Let's consider a raw image input of size 32 x 32 x 3 (width x height x channels) for the convolutional process. During this process, we can envision a "filtered image" of dimensions 5 x 5 x 3 (width x height x channels). In the forward pass, each filter is shifted (or convolved) over the input data, calculating the element-wise product of the filter and the corresponding input region. This operation generates a 2D feature map for each filter.

In Convolutional Neural Networks (CNNs) designed to handle high-dimensional data like images, the neurons in a convolutional layer are selectively connected to only a small portion of the input, known as the receptive field. This approach allows for more efficient and effective data processing.

Moreover, the spatial arrangement of the network also plays a crucial role in size reduction. For instance, a convolutional operation with a 32 x 32 x 3 input image can yield a 28 x 28 x 6 output map, thereby reducing the output's spatial dimensions.

2.2.2 Pooling Layer

Within CNNs [45], layers play a pivotal role, serving as a form of nonlinear down sampling. Among the various pooling functions, maximum pooling stands as the most prevalent. As depicted in Figure 7, this process involves dividing the input into rectangles and outputting the maximum value within each region. Operating independently across all depths, layers facilitate spatial resizing of concepts.

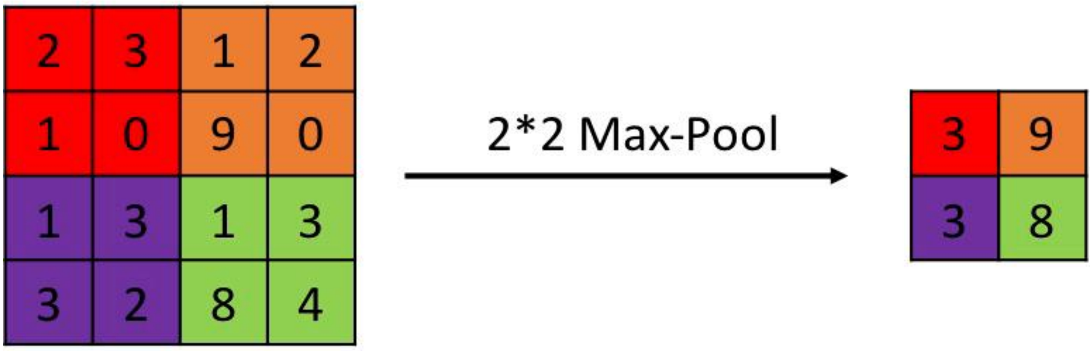

Figure 7. The inputs for max pooling operation.

A specific type of layering involves utilizing a 2 x 2 filter and employing a 2-step pooling process for the input. This strategic approach leads to an impressive 75% reduction in parameters. Pooling layers are introduced based on the idea that the relative position of a feature concerning other features carries more significance than its precise location in the input image.

2.2.3 Activation Layer

The activation layer plays a critical role in the CNN, applying the activation function to the input to introduce essential nonlinear states into the network. Among the plethora of activation functions, ReLU stands as the most commonly used one. Figure 8 illustrates the ReLU activation process, which proves effective for data processing within the network.

2.2.4 Fully-Connected Layer

Within neural networks, the most computationally demanding operation occurs during the connection process [46]. In this process, each neuron in a layer is connected to every neuron in the preceding layer. Figure 9 provides an overview of these interconnected layers.

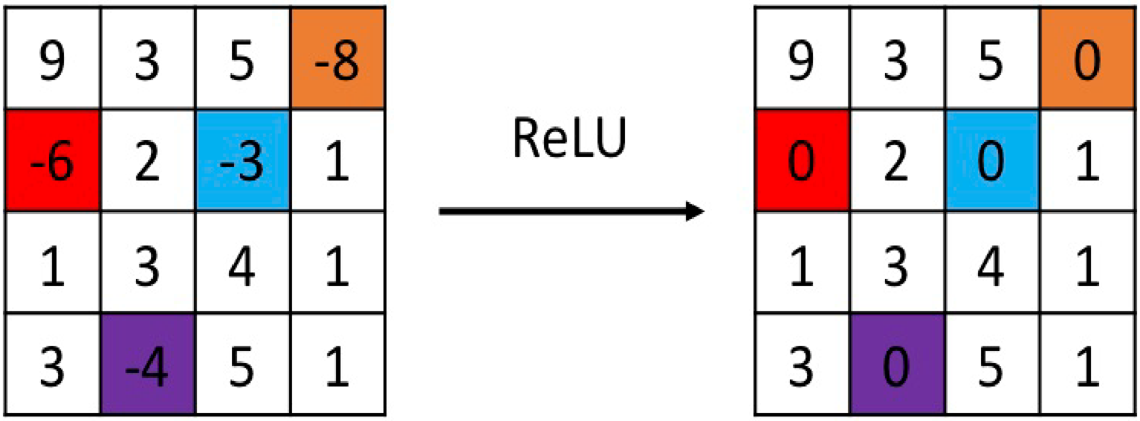

Figure 8. ReLU activation layer applied to input.

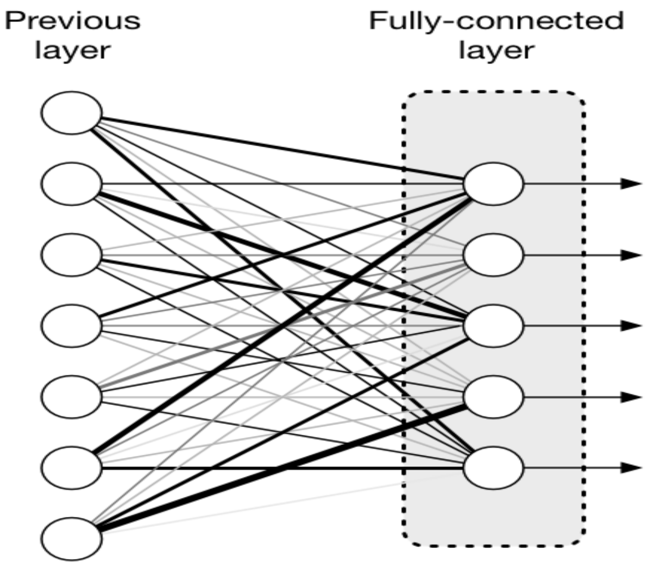

Figure 9. Illustration Fully-connected layer.

2.2.5 CNN Output Layer

The output process of a Convolutional Neural Network (CNN) plays a pivotal role in determining the final output, which is the distribution of possible classes for the classification function. The nature and size of this output are task-dependent, tailored to the specific objective that the CNN [47] aims to accomplish. For instance, in a binary classification task, the output layer will comprise a neuron outputting a value representing the probability that the input belongs to the class. On the other hand, in multi-class classification, the output layer may contain multiple neurons, each representing the probability of the input belonging to a particular class. In certain regression tasks, the output layer may consist of a neuron returning a continuous value rather than a probability.

In this section, we present a comprehensive overview of relevant activities in vehicle detection. We delve into recent advancements in object detection in digital images using convolutional connectivity, with a focus on robust evaluation metrics. The YOLO method takes center stage, as we conduct an in-depth exploration of its properties, characteristics, and understanding. Additionally, we offer a historical perspective on the methods utilized for identifying and classifying objects in digital images, emphasizing their approaches, accuracy, and evolving results over time.

2.3 Summary of object detection variants using YOLO

YOLO (You Only Look Once) stands out as a renowned object detection tool, celebrated for its exceptional time efficiency and remarkable accuracy. It distinguishes itself by utilizing a single neural network to simultaneously predict bounding boxes and classes for objects present in images. The efficiency and accuracy of YOLO have garnered widespread acclaim, making it one of the most popular choices in the field of object detection. Over the years, several versions of YOLO have been developed, each introducing advancements and improvements to enhance object detection capabilities. Different version of YOLO algorithms with timeline is seen in figure 10 .

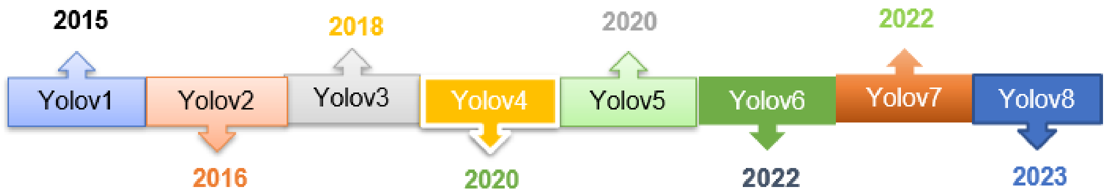

Figure 10. Different version of YOLO algorithms with Timeframe.

YOLO based Detection	Methodology	Analysis & Results	Conclusions
Using YOLOv5 [48]	Real-Time Detection of Heavy-Duty Vehicles for Parking Spot Occupancy Using YOLOv5	The model showcases notable enhancements in detection precision, recall, and average precision (mAP) for object detection tasks. While it excelled in accurately detecting the front cabin with high confidence, it faced challenges when identifying objects situated at significant distances.	Improved Close Range Detection, Limited Performance at Far Distance: The model has demonstrated significant enhancements in detecting objects at close range. However, its detection performance diminishes when it comes to objects located at a considerable distance.
YOLOv5 [49]	Using the YOLOv5 series, the car can be effectively controlled in various ways.	The model achieved high precision and recall, displaying remarkable accuracy and a 96.8% recall rate during testing for the detection of eight types of car engines. It outperformed larger models with the same task.	Comparison of Precision, Recall, and Speed: The experiment revealed that smaller models exhibited superior precision and recall compared to larger models, while still maintaining their advantage in terms of speed due to their compact size.
YOLOv5 [50]	"Testing," "Self-monitoring of the YOLOv5 model," and "Polyp target	The model exhibits excellent recall, accuracy, and precision, achieving scores over 90% accuracy when utilizing full image	Incorporating full integration of image data into each mesh of the model enhances detection ability and minimizes the

	detection mechanism."	data for all network predictions.	occurrence of false positives.
YOLOv5 [51]	In the chapter, flaw detection is tested using YOLOv5 with an additional Squeeze and Excitation (SE) layer.	Model attained an impressive mAP@0.5 score of 94.7%, marking a substantial 9% enhancement compared to the original model.	The inclusion of a compact layer like the SE layer not only enhances the model's detection capability but also accelerates the detection process.
YOLOv5 [52]	In this chapter, an end-to-end real-time helmet detection system for riders is developed using YOLOv5, incorporating the Kmeans algorithm to calculate connectivity.	The model achieved outstanding mAP of 97.7% and F1 scores of 92.7%, surpassing other state-of-the-art models in performance.	Use of well calculated anchors can greatly improve model detection.
YOLOv7 [53]	The paper presents an ensemble model for detecting translated infrared images, combining pix2pix, GAN, and YOLOv7 on visible-infrared image pairs. The model is tailored for low-light visibility conditions.	The model exhibited superior performance when trained solely on original images, outperforming the model trained on translated images, particularly in low-light conditions.	The utilization of translated images can significantly improve the models accuracy, particularly in low-light conditions. Additionally, combining a well-designed ensemble model composed of effective algorithms can outperform the performance of an

			original detection model algorithm.
YOLOv5 and YOLOv7-tiny [54]	In the experiment, a combination of YOLOv5 with a lightweight algorithm was compared to YOLOv7-tiny in fire scenarios.	The model achieved a notable improvement of 6.8% compared to YOLOv7-tiny, showcasing enhancements in both mAP and FPS .	The combination of state-of-the-art algorithms with lightweight algorithms showcases impressive speed and accuracy, highlighting their effectiveness across diverse applications.
YOLOv5 and YOLOv7 [55]	The experiments aim to compare YOLOv5 and YOLOv7 in tracking performance using various evaluation metrics.	The evaluation results revealed that YOLOv7 exhibited better performance in certain metrics like MOTA, MOTP, and IDF1.	Both algorithms demonstrated favorable results in the tracking experiment. The selection of evaluation metrics for an experiment will determine which model is preferable for implementation.
An autonomous rack inspection framework based on computer vision (CV) utilizing YOLOv7 is proposed [56].	A computer vision-based autonomous rack inspection framework is devised, incorporating YOLOv7 and a domain variance modeling mechanism to handle data scarcity.	The model achieved mAP of 91.1%	By automating the process, the model effectively minimized human errors caused by undetected damages, leading to a more reliable outcome.

CHAPTER – 3

MACHINE LEARNING TECHNIQUES

Focuses on machine learning techniques, including the theory, formulas.

3.1 Machine Learning

3.1.1 A Supervised Machine Learning

3.1.2 An Unsupervised Machine Learning

3.1.3 Reinforcement Machine Learning

3.2 Tools Used

3.2.1 High level language Python

3.2.2 Colab Notebook

3.2.3 Computer

MACHINE LEARNING TECHNIQUES 3

In recent times, the widespread adoption of machine learning techniques has led to impressive advancements across various applications, such as image and speech recognition, natural language processing, object detection, and predictive analytics. This chapter aims to provide a distinctive and comprehensive introduction to the realm of machine learning. It delves into fundamental concepts and explores different types of machine learning techniques, encompassing supervised learning, unsupervised learning, and reinforcement learning. Additionally, the chapter offers an exclusive overview of the most commonly employed algorithms, including linear regression, decision trees, and artificial neural networks.

3.1 Machine Learning

The fourth industrial revelation age now has brought about some challenges for humanity. The rapid technological advancement, increased human interaction with electronic devices and the internet, and the proliferation of electronic records have combined to create vast amounts of data every second. In recent decades, companies, universities, scientists, and academics have been seeking new methods and innovations to make use of this data for various purposes, such as detection, recognition, analysis, identification of evidence, and recommendations. Almost every industry now uses AI to improve and streamline their processes and methods, such as in medicine, engineering, finance, and manufacturing.

The term "Machine Learning" was first coined by Arthur Samuel in 1959 and has since undergone continuous development and refinement by numerous researchers. As a subfield of Artificial Intelligence, Machine Learning employs statistical techniques to empower computer applications with the ability to dynamically learn from data without explicit instructions or predefined programs [57]. By discovering patterns and relationships between data and events through computer algorithms, ML facilitates predictions on similar data based on the training it receives. Unlike traditional applications governed by rigid pre-defined rules, ML algorithms have the unique capacity to learn and improve over time, enabling the creation of data-driven applications like computer vision or email filtering, which would be exceedingly challenging using conventional programming methods. These algorithms play a vital role in aiding decision-making processes and enhancing reliability.

Machine Learning is broadly categorized into three main types: Supervised, Unsupervised, and Reinforcement Learning.

3.1.1 A Supervised Machine Learning

For AI problems ,the concept of supervised learning is frequently used. It involves two elements: inputs and outputs. The algorithms attempt to establish a mapping relationship between these two elements via a mapping function.[58] The algorithms are trained with data sources and are evaluated based on the output they produce. During the training phase, the algorithms are monitored like a teacher or coach. When the algorithms make predictions, they are evaluated to determine if they are correct, near desired result . To learn and improve their performance this process helps the algorithms continuously. The process of learning will be ends when a satisfactory accuracy level is achieved. When new data is introduced, the algorithms use their past learnings and the estimated mapping to predict the output.

Regression and classification can be grouped by Supervised learning algorithms. A concern with these algorithms is that they only work with labeled data, and acquiring data for learning can be expensive. Many supervised algorithms have been tested, each with its own strengths and weaknesses. Choosing the right algorithm is crucial in ML, as no single algorithm works best for all tasks.

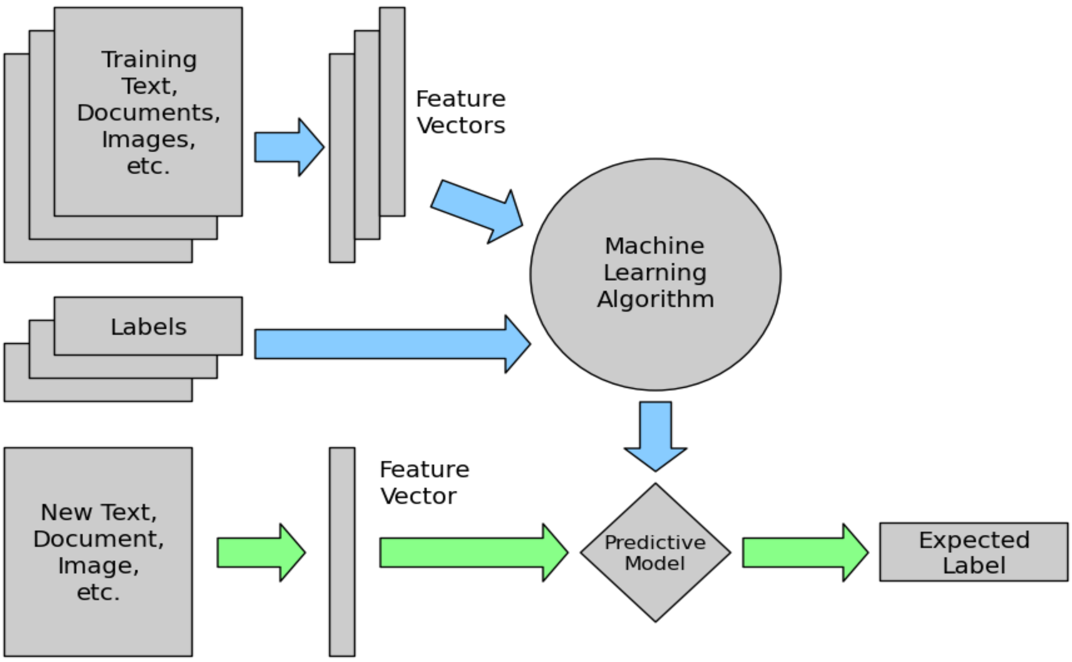

Figure 11. A Supervised Machine Learning Diagram.

3.1.2 An Unsupervised Machine Learning

Unlike supervised learning, A correct answers do not have pre-determined by unsupervised algorithms. To correct mistakes there is no output variable and no supervisor in this algorithms .To understand the data by finding patterns and structures within the data is the main goal of the algorithm. They do this without the need for pre-labeled data, making unsupervised [59] learning a powerful tool for uncovering hidden information. Clustering and association problems are the classification of Unsupervised learning.

Clustering: This type of task involves grouping similar data points together, such as grouping customers based on their purchasing behavior.

Association: The algorithms aim to discover the underlying relationships and patterns in the data, such as the relationship between buying shirts and pants. Some unsupervised learning algorithms include:

• Clustering of K-means • Association algorithm with A priori

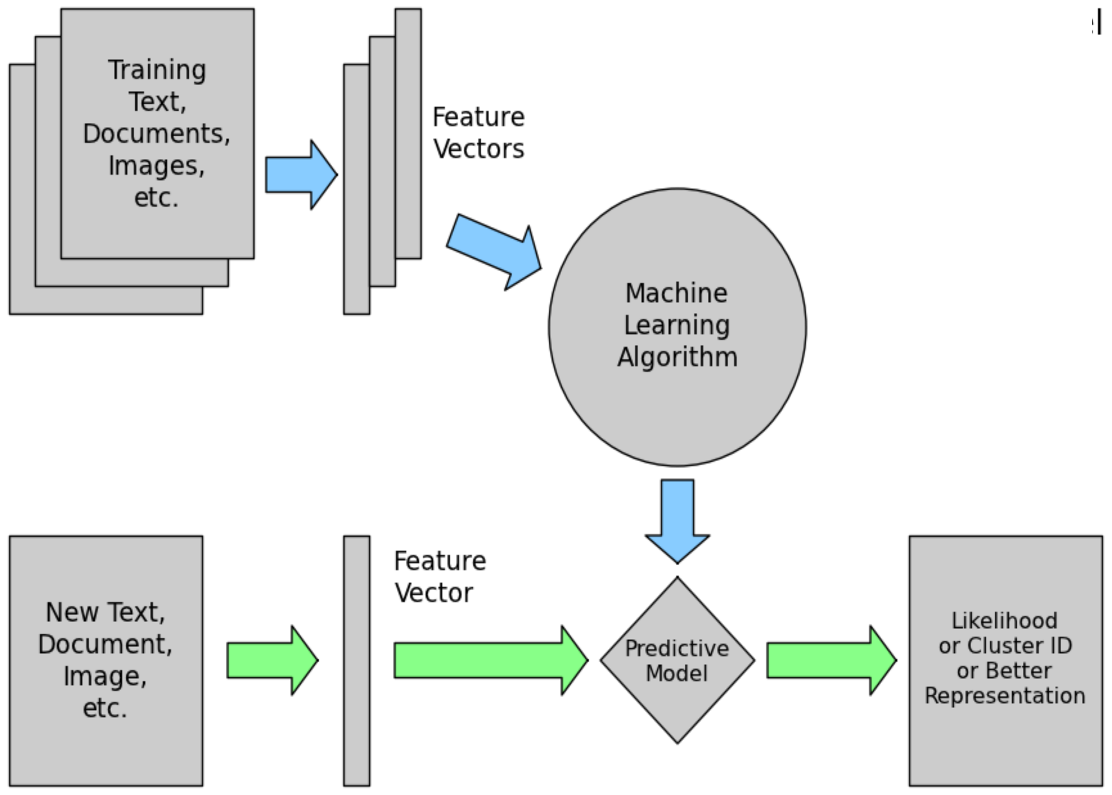

Figure 12. Diagram of An Unsupervised Machine Learning model

3.1.3 Reinforcement Machine Learning

Reinforcement learning [60] draws parallels to the learning process of a young child in their early stages of development. Much like a child receiving rewards for good behavior and guidance for mistakes, reinforcement learning algorithms function similarly, with an agent acting as the learner. This agent interacts with its environment, earning rewards for successful task completion and facing consequences for suboptimal performance. The overarching objective is for the agent to maximize rewards while minimizing penalties.

For example, envision a self-driving vehicle: it receives rewards when it reaches its destination without accidents, deviations from the road, or improper stops. However, if it makes any of these mistakes, it incurs penalties. This process allows the vehicle to learn and refrain from repeating actions that resulted in penalties. These algorithms, also referred to as dynamic programming, remain the subject of ongoing research and improvements, promising a wide array of potential applications in the near future.

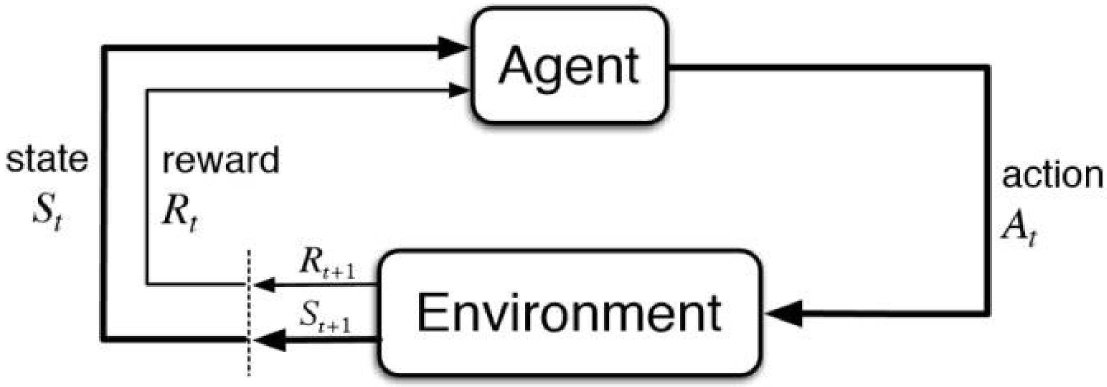

Figure 13. Reinforcement Learning diagram

3.2 Tools Used

This thesis employed various tools and applications in the creation of models and experiments, like most research and studies. A number of tools are necessary for these types of studies, such as the programming language Python, which was utilized by the author to develop the model. A large dataset of vehicle images was also used for training purposes. Additionally, several Python libraries, deemed necessary or helpful in creating machine learning models, were utilized.

3.2.1 Python High level Language

Python, a versatile and dynamic programming language created by Guido van Rossum, was crafted with the noble vision of making programming accessible to everyone. Its simplicity, clear indentation-based syntax, and user-friendly nature have garnered widespread acclaim, particularly among individuals with no prior programming background. Throughout the years, Python has risen to remarkable popularity, especially in the realms of Artificial Intelligence and Machine Learning, courtesy of its efficient and convenient libraries that expedite development processes.

In this thesis, the author thoughtfully chose Python as the programming language for model development due to their profound knowledge and genuine interest in the language. The research delves into the utilization of specific libraries, meticulously selected to enhance and streamline the implementation process. The following libraries have played a pivotal role in this research

3.2.1.1 Numpy

An open-source library which is called Numpy that allows multi-dimensional matrices and arrays for efficient computing. To make working with this type of data more straightforward. In data analysis, arrays are often used to speed up operations and increase efficiency, so NumPy is a crucial tool for data scientists when working with large amounts of data. It is particularly useful for detection and forecasting tasks, where models need to be optimized to perform quickly.

3.2.1.2 Matplotlib

Matplotlib, a widely favored plotting library, serves as a powerful tool for generating an array of graphs and figures. Its user-friendly interface enables the creation of high-quality plots and figures with minimal lines of code, making it the preferred choice for data visualization. In this thesis, Matplotlib is harnessed for extracting color features and constructing histograms.

3.2.2 Colab Notebook

Colab, short for Google Colaboratory, is a free online platform for the development of machine learning and data science. It provides the user to write ,run code in Python using Jupyter notebooks in the cloud, without having to install any software on their own devices.

Colab offers access to powerful GPUs and TPUs for training machine learning models and running computationally intensive tasks.

In addition to being free and easy to use, Colab provides a number of useful features for data scientists and machine learning engineers. For example, it integrates with Google Drive, allowing users to save their notebooks and data to the cloud. It also provides easy access to a variety of popular machine learning libraries and tools, including TensorFlow, PyTorch, and Keras. Another great feature of Colab is the ability to collaborate on notebooks with others. Multiple people can work on the same notebook at the same time, making it a great tool for teamwork and collaboration in data science projects.

Overall, Colab is a valuable resource for anyone interested in data science and machine learning. Its combination of powerful computing resources, integration with Google Drive and other Google services, and collaboration capabilities make this a popular choice among scientists and engineers.

3.2.3 Computer

The computer utilized training and testing for the models which has the following specifications:

Device: HP ProBook 13-5378

Memory: 8 GB RAM

CPU: Core i5 Quad and Core Graphics: Intel HD 4 GB"

Moving forward to Chapter 4, we will delve into an in-depth presentation of our proposed model. This comprehensive exploration will encompass its architecture, the dataset employed for training and evaluation, as well as the research methodology utilized.

CHAPTER – 4

PROPOSED VEHICLE DETECTION METHODS

A comprehensive explanation is provided for the machine learning algorithms and models employed, complete with their underlying formulas thoroughly elucidated.

PROPOSED VEHICLE DETECTION METHOD 4

This chapter focuses on the proposed vehicle detection methods and materials that have been developed in recent years. The chapter provides an overview of You Only Look Once version5 (YOLOv5), and their performance. In addition, this chapter introduces the materials that have been used for evaluating and benchmarking vehicle detection methods, such as datasets, evaluation metrics. The chapter also provides a discussion on the performance of the modified methods and materials. It is also provide as a comprehensive resource for vehicle detection, providing a detailed understanding of the proposed methods and materials and their current status.

4 Proposed Model

Here we proposed YOLOv5 model that can be used for detecting vehicle in images and videos. In detecting vehicle context, YOLOv5 can be trained on a dataset of images and videos containing annotated vehicles, and then used to detect vehicles in new images or videos. YOLOv5 [61] offers several advantages for vehicle detection. For vehicle detection an intelligent model is developed which is proposed YOLOv5,Though there are several types of YOLO algorithms like YOLOv5m, YOLOv5s, YOLOv5l, YOLOv5n, and YOLOv5x, In this study we used YOLOv5 original that means YOLOv5s for modifying and also used Python coding, OpenCV libraries, TensorFlow, and Keras libraries and so on. Intel processor, 32GB RAM, and GTX x GPU with 12GB VRAM. This article uses a combination to achieve better quality and accuracy in vehicle detection of frame recognition and edge detection algorithms. It uses the YOLO5 algorithm to accurately estimate and track each vehicle's location. It can also be used to classify detected vehicles into various designated groups. A diagram of this detection system is shown in Figure 14, this includes the following: enhancement of image, detection of edge, using a combination of different methods of motion analysis, and complete image preprocessing. Image labeling is done using annotation methods. The definition of the detection area will be selected by the bounding box, and finally, the types of vehicles will be detected and classified according to the vehicle types. It must be said that several assumptions are made here: That is (i) Random change of direction is observed (ii) There is no accident (iii) There are physical and legal restrictions on the vehicle (iv) motion is recorded with a top-down view of the pavement. YOLO-v5 relies on CNN for vehicle tracking. Dividing the image into regions the algorithm computes bounding boxes for each region. The algorithm only needs to go through the neural network once to make a prediction, and it "sees" the image only once. Their output is a known object with a bounding box after erasure to zero [62, 63]. A state-of-the-art detection object model that can be used for vehicle detection in images and videos, offering a combination of accuracy, speed, and robustness.

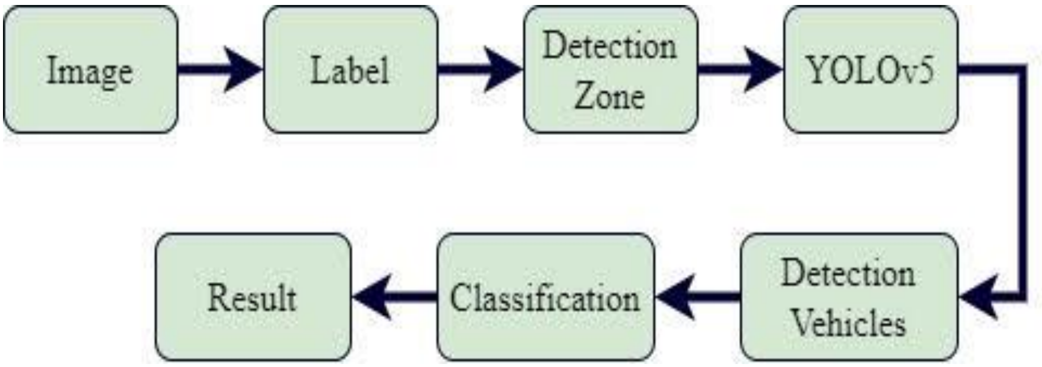

Figure 14. Flowchart of Vehicle detection

4.1 Structure of YOLOv5 Model's

Detection of objects is the classic digital tasks for determining objects which are present and where they are located. Object detection tasks are more complex than classification tasks, which recognize objects but do not specify their location in an image. Also, there are many objects in the image that cannot be classified. The models family consists of three major building blocks: i) Backbone, ii) Neck, and iii) Head [64]. Where backbone uses Darknet CSP as a basis for extraction of feature from images composed to interlaced LANs. YOLOv5 Head: A class that generates predictions from anchor blocks for detection objects;

Backbone: This is an important layer of this model where it is responsible for extracting feature maps from the input image. The model uses a pre-trained deep neural network as the backbone[65], such as ResNet, MobileNet, or EfficientNet.

Neck: This is the next layer of the model where it is responsible for connecting backbone network to the head.[66] It typically includes a series of layers like convolutional layers and pooling layers where spatial dimensions of the feature maps can be reduced.

Head: Head which is mainly responsible for making final predictions, including the class probabilities and the bounding boxes in the input image objects. [67] The layer also includes layers of convolutional and layers of fully connected that produces the predictions for each cell in the grid.

Loss Function: The loss function which is used in training section to evaluate the accuracy of the model's predictions. YOLOv5 uses a combination of classification and regression loss functions [68], such as the class probabilities, cross-entropy loss and mean squared error loss for the bounding box predictions.

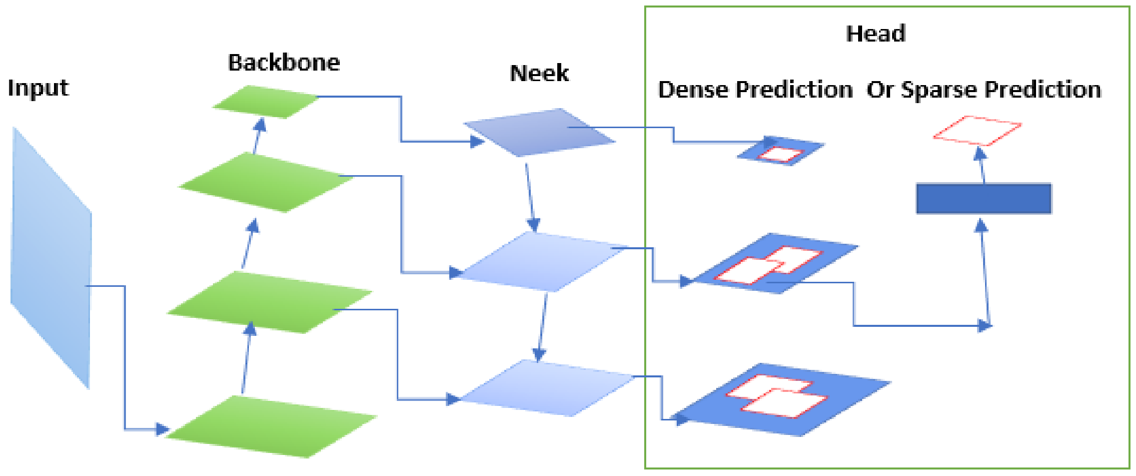

Figure 15. YOLO5 object detection process

YOLO-v5[69] the latest update of the YOLO series uses here. A combination of object recognition and bounding box estimation is an end-to-end identifiable network which is occurred by the recognition model YOLO which is written and maintained in the Darknet environment. In previous YOLO variants, YOLO-v5 [70] is the first YOLO model developed using the Py-Torch framework and is lighter and easier to use. YOLOv5 is designed to be highly scalable, with multiple hyperparameters that can be adjusted to control the size and accuracy of the model. Additionally, for a wide range of object detection tasks it's designed to be easy to use, with pre-trained weights and pre-built architectures that can be used.

4.1.1 Function of Backbone

In figure 16. It represents the general function of image processing of the backbone section in YOLOv5.

- Suppose we have an input image of various objects, such as cars, bus, and ambulance and so on. The images are required to passed through the backbone network, this network includes some layers like convolutional, pooling, and non-linear activation functions. A set of feature maps can be transformed to the input image and each layer in the backbone network applies operations like convolutions and activations.
- These transformation capture different levels of information, such as high-level semantic concepts and low-level visual patterns.
- The backbone might identify edges, corners, or more complex features like car shapes or pedestrian body structures.

- Extracting informative features from the input image is the main purpose of the backbone that will be used by subsequent components for object detection.

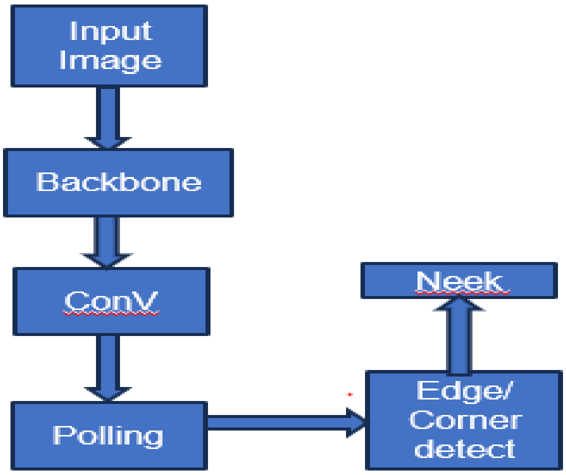

Figure 16. Function of Backbone

4.1.2 Function of Neek

Following the backbone section then the neek section comes,In figure 17 which define the work of YOLOv5 neek.

- The neck component takes these feature maps and performs operations like upsampling, concatenation, and convolutional layers.
- The neck fuses the features from different scales to capture both fine-grained details and high-level contextual information of the objects in the image. This helps improving model's ability to detect objects accurately.
- Feature pyramid which generated by the neck is then passed to the head component.

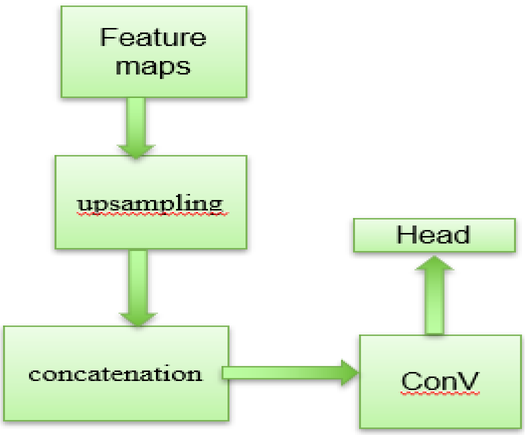

Figure 17. Function of Neek[71]

4.1.3 Function of Head

After completing the previous section's function then the image goes to the final section which is Head. Figure 18 denotes the general function of head section.

- The feature maps from the backbone are now processed by the head component.
- The head which consists of some special layers like convolutional, fully connected, and other operations.
- Predicting bounding box coordinates, class probabilities, and scores of confidence for each cell of the feature maps are occurred in this head section.
- Anchor boxes are utilized by the head which can be predefined by the bounding box shapes at different scales. Based on the feature maps, the head refines the anchor boxes by adjusting their coordinates using learned offsets.
- Indicating a specific object class it also performs classification to assign object class probabilities.
- After that an important function called NMS is applied. The function of NMS is to overlapping bounding boxes, to filter out redundant and keeping only the most confident and accurate detections.

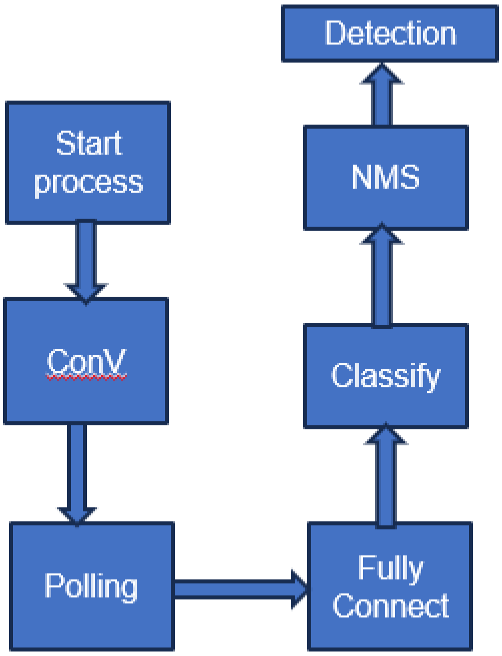

Figure 18. Function of Head

Combining the three major parts then the YOLOv5 established and we get a simplified block of YOLOv5 which is given in figure 19.

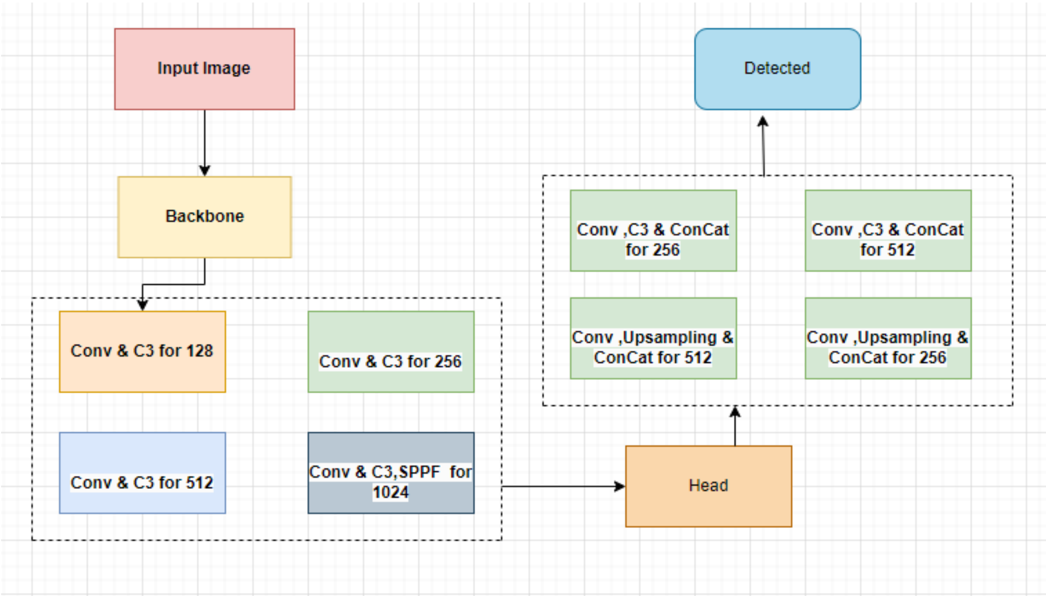

Figure 19. Structure of YOLOv5 Original

4.1.4 YOLOv5s

YOLOv5s (small) is a object detection system in real-time that is designed for both accuracy and speed. This is the YOLO variant of detection algorithm, and is part of a larger family of YOLO algorithms. That indicates a lighter and faster version of the original YOLOv5 algorithm.[72]

For detecting objects in an image or video frame.[73] YOLOv5s uses a single convolutional neural network (CNN) architecture. The input image into a grid of cells is splitted by the network and assigns each cell a predicted bounding box for objects class probabilities that may be present within the cell. To improve accuracy and reduce false positives the algorithm also uses anchor boxes.

One of the key benefits of YOLOv5s is its ability to run efficiently on a variety of hardware, including CPUs and GPUs, making it suitable for deployment on a wide range of devices and platforms. Additionally, the YOLOv5s algorithm is highly customizable, allowing users to train their own models for specific object detection tasks.

4.1.5 YOLOv5m

Another version 5 models is YOLOv5m (medium) that is a variant of the YOLO detection algorithm. The "m" in YOLOv5m stands for "medium", indicating that it is a more powerful and accurate version of the algorithm compared to the smaller YOLOv5s, but still not as complex as the original YOLOv5.[72]

Like other YOLO algorithms, YOLOv5m uses a convolutional neural network (CNN) architecture for detecting objects in an image or video frame. A grid of cells is occurred in the network for spliting the input image and also splits each cell to predicts bounding boxes and class probabilities. To reduce false positive and to improve accuracy, Non-Maximum Suppression (NMS) [74] and anchor boxes are used in this algorithms.

YOLOv5m is designed to be a balance between accuracy and speed, making it suitable for a wide range of object detection tasks. Additionally, the YOLOv5m algorithm is highly customizable and can be trained on specific data sets to improve accuracy for specific object detection tasks.

4.1.6 YOLOv5n

YOLOv5n (You Only Look Once version 5 nano) stands out as a distinctive adaptation of the YOLO (You Only Look Once) object detection algorithm. The "nano" designation signifies its lightweight and high-speed characteristics, optimized to operate efficiently on devices with constrained computational resources.

Similar to other YOLO algorithms, YOLOv5nano utilizes a single convolutional neural network (CNN) architecture to detect objects in images or video frames. By dividing the input image into a grid of cells, it predicts bounding boxes and class probabilities for each cell [75]. Employing anchor boxes and Non-Maximum Suppression (NMS), the algorithm enhances accuracy while reducing false positives.

With its focus on maintaining a balance between accuracy and speed, YOLOv5nano emerges as an excellent choice for diverse object detection tasks on low-power devices. Furthermore, the YOLOv5nano [76] algorithm offers high adaptability, enabling customization and fine-tuning on specific datasets to elevate accuracy for targeted object detection objectives.

4.1.7 YOLOv5l

The YOLOv5large (You Only Look Once version 5 large) is an advanced variant of the YOLO detection object algorithm, designed specifically to handle complex object detection tasks. Unlike its smaller counterparts, YOLOv5s and YOLOv5m, the "large" in YOLOv5large denotes its increased complexity and accuracy.

Within the framework of YOLOv5large, a singular convolutional neural network (CNN) architecture is harnessed, intelligently dividing the input image into a grid of cells. It then proceeds to predict bounding boxes and class probabilities for each cell. Employing anchor boxes and Non-Maximum Suppression (NMS), this algorithm effectively enhances precision and minimizes the occurrence of false positives.

While YOLOv5large [77] offers the highest accuracy for object detection tasks, it may be slower than smaller YOLO variants. Nonetheless, the algorithm's customizability allows it to be tailored to specific data sets, resulting in improved accuracy for specific object detection tasks.

4.1.8 YOLOv5x

YOLOv5x is a state-of-the-art real-time detection object model developed by the YOLO (You Only Look Once) team. It is designed to run efficiently on a wide range of hardware, including edge devices, making it suitable for real-world applications that require quick and accurate detection object.

The core idea behind YOLOv5x is to use a single convolutional neural network to detect objects in an image.[78] This allows the model to process images in real-time, making it suitable for tasks such as video analysis or autonomous driving. The model uses anchor boxes and anchor-based detection, which allows it to make multiple predictions per anchor and perform well on small objects. This is particularly important in real-world scenarios, where small objects, such as traffic signs or pedestrians, can be critical to detect.

YOLOv5x which is a robust as well as efficient real-time detection object model, known for its high accuracy and fast processing speed. Its high average precision (AP) on benchmark datasets demonstrates its accuracy, while its real-time performance makes it suitable for various applications. This combination of accuracy and speed makes YOLOv5x a valuable tool for object detection in real-world scenarios, such as surveillance, autonomous vehicles, and robotics. With its exceptional performance and efficiency, YOLOv5x is a gamechanger in the field of computer vision, and its potential applications are numerous, particularly in fields that require real-time object detection. Therefore, YOLOv5x represents a significant advancement in the field of detection object, with its ability to provide accurate and efficient real-time detection.

4.1.9 Proposed_ YOLOv5

The proposed version of YOLOv5 introduces a novel approach by expanding the architecture with an extra convolutional layer added to both the backbone and head sections. This enhancement augments the model's capabilities and sets it apart from the original YOLOv5.

Backbone Modifications: In the traditional YOLOv5 backbone network, feature extraction relies on several convolutional layers. However, in this modified version, an additional convolutional layer is introduced into the backbone. This extra layer aims to capture more intricate patterns from the input images and further enhance the representation of extracted features. The added convolutional layer empowers the model to achieve a deeper understanding of the underlying data, improving its overall performance in feature extraction.

Head Modifications: The head of YOLOv5 typically includes convolutional layers responsible for predicting bounding boxes, class probabilities, and other relevant information. In the modified version, an extra convolutional layer is introduced into the head. This additional layer can enable the model to learn more complex spatial relationships or encode higher-level features.

By adding an extra convolutional layer to both the backbone and head, the proposed YOLOv5 model can potentially improve its ability to detect objects accurately. The added layers allow for more expressive power and increased capacity for capturing intricate details in the input data. The diagram of proposed _YOLOv5 is seen to figure 19.It's important to note that the specific configuration, hyperparameters, and design choices of the added convolutional layers can vary based on the requirements and goals of the modification. Experimentation and fine-tuning may be necessary to determine the optimal architecture and parameters for the proposed YOLOv5 model.

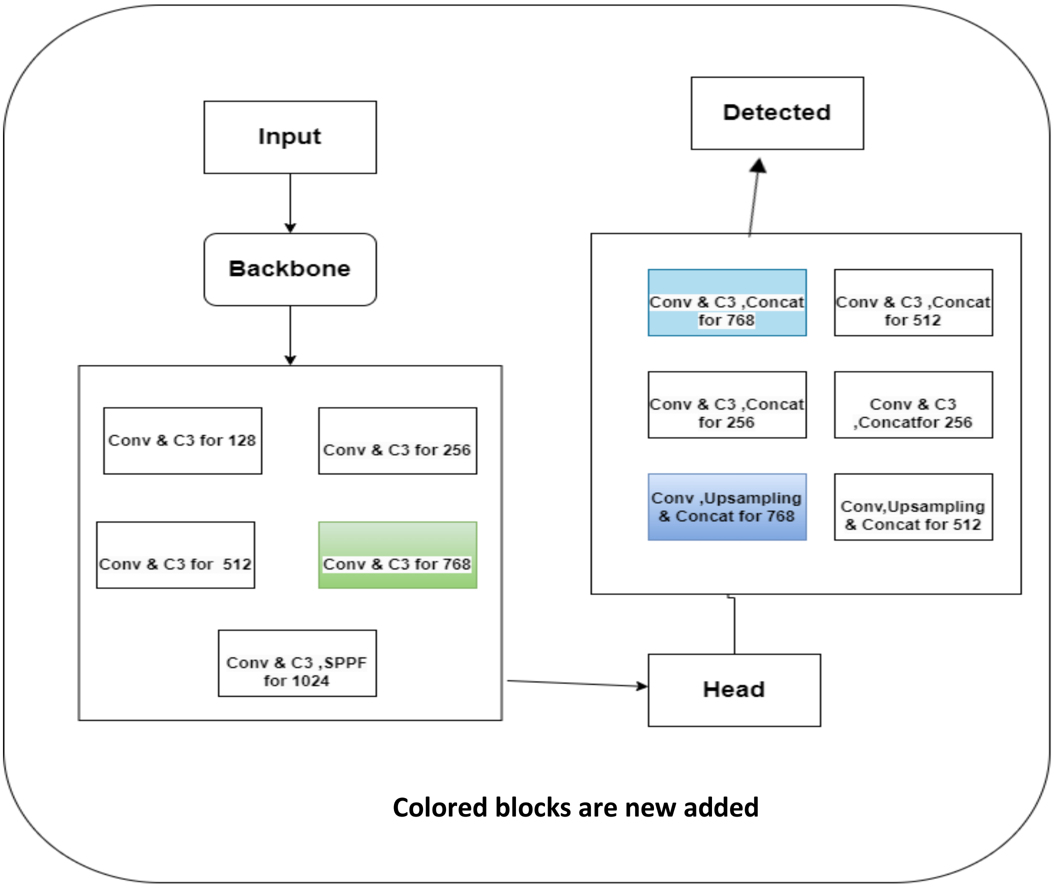

Figure 20. Structure of Proposed YOLOv5

4.2 Dataset and Preprocessing

The YOLOv5 patterns were detected and trained using the ROBOFLOW dataset design. This dataset comprises 3001 images, including 2600 training images, 251images for validation, and 125 images for testing. All images are sized at 416x416 pixels, and the dataset includes six classes: Bus, Truck, Motorcycle, Ambulance, Bicycle, and Car which is distributed in table 2. To ensure that the model is trained accurately, the dataset includes images of vehicles captured from various angles, including front, rear, side, and top views. It is crucial for each class to have sufficient representation to enable proper training of the vehicle detection model. The vehicle selection dataset including Bus, Truck, Motorcycle, Ambulance, Bicycle, and Car is a structured dataset that includes information about different types of vehicles given in figure 21.

The preprocessing steps for this dataset may include:

- Removing missing or invalid data points: This could involve removing rows with missing values or removing duplicates to ensure that the data is complete and accurate.

- Encoding categorical variables: The make and model variables are likely to be categorical, so they may need to be encoded using techniques such as one-hot encoding or label encoding.
- Normalizing numerical variables: If the numerical variables such as engine size and price are on different scales, they may need to be normalized to a common scale to ensure that they have similar impact on the analysis.
- Encoding nominal variables: The type of vehicle is a nominal variable that needs to be encoded into numerical values for analysis. One way to do this is by using one-hot encoding.
- The process involves dividing the data into three distinct sets: training, validation, and test sets. The dataset is partitioned, with one set designated for model training, another for model validation, and the final one for model testing.
- Performing feature selection: This could involve identifying the most important features that impact vehicle selection, such as price, engine size, and fuel type.

To ensure these preprocessing steps it can be helped that the vehicle selection dataset is clean, consistent, and ready for analysis. They also help to ensure that any machine learning models trained on the data are accurate and reliable.

Table 2. Vehicle Distribution Sample

Class Name	Label Count
Bicycles	184
Car	2200
Bus	186
Truck	158
Motorcycle	241
Ambulance	148

Figure 21. Vehicle labelling category of the dataset

4.3 Evaluation Details

Throughout this study, we rigorously assess the effectiveness of our advanced vehicle detection system using the average precision (AP) metric. The forthcoming section provides a thorough overview of crucial concepts, encompassing the confusion matrix, precision, recall, Intersection over Union (IoU), and mean average precision (mAP) [78]. It is vital to acknowledge that the evaluation hinges on the Roboflow dataset object detection benchmark, establishing it as the benchmark for evaluating the system's performance in this specific category.

4.3.1 Confusion Matrix

In the detection object task, we aim accurately identify the presence and location of objects within an image. The goal of detection object is to identify objects within an image and accurately determine their location. This is typically accomplished through binary predictions, which indicate whether an object has been successfully detected or not. However, there may be cases where the model's predictions do not match any actual objects present in the image. These instances can result in false positives, where the model mistakenly identifies an object that is not there, or false negatives, where fails to detect an object that is present in the input image.[79] Therefore, to improve the accuracy accurately of detection object models is a challenging task that requires careful consideration of various factors such as dataset quality, model architecture, and training parameters.

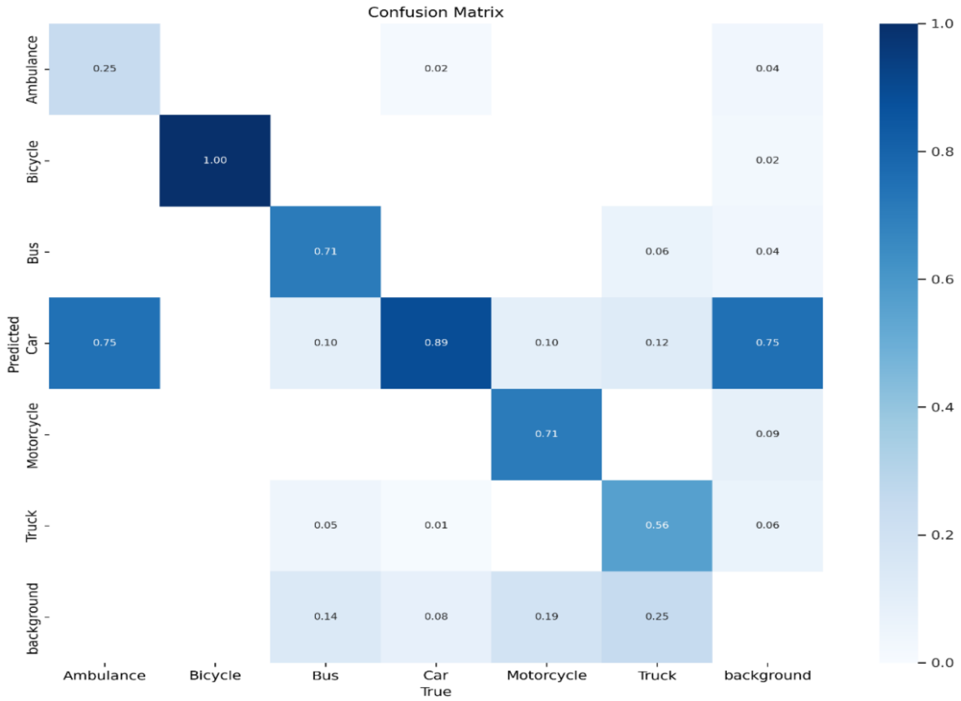

Figure 22. Confusion Matrix YOLOv5 original

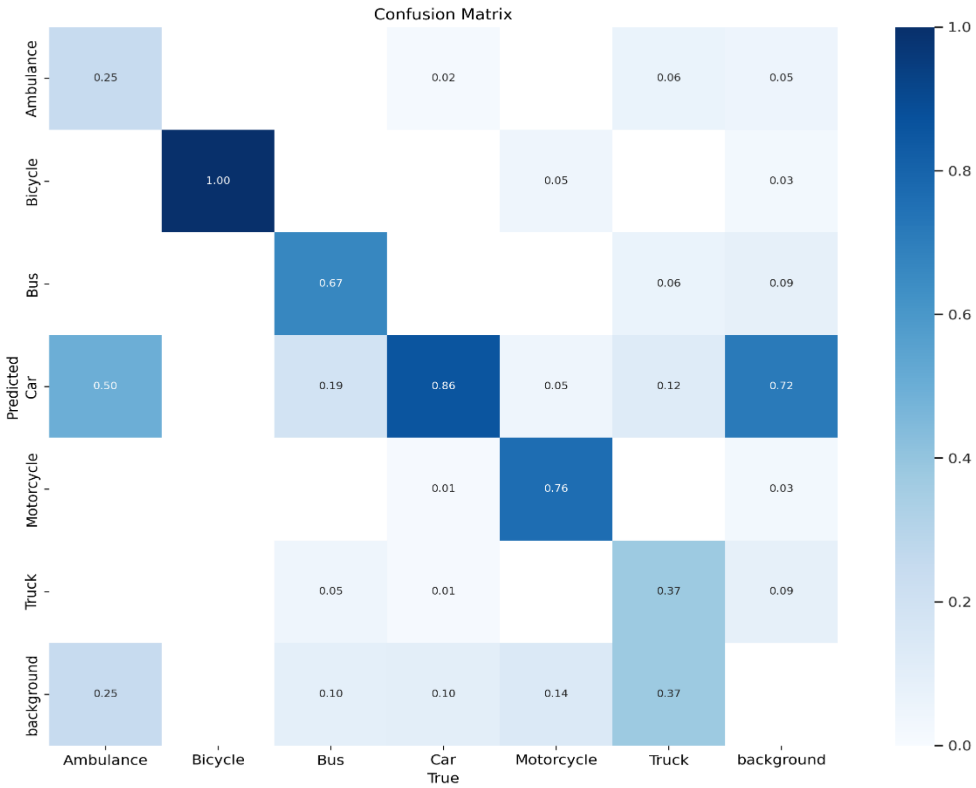

Figure 23. Confusion Matrix of proposed YOLOv5

4.3.2 Precision, Recall and F1 Score

Evaluating result to this work, following formula was used to calculate the overall mean accuracy for vehicle detection as [80]

$$mAP = \frac{1}{n}\sum_{k=1}^{k=n} AP_k \qquad (1)$$

Where n, APk, and K bear the meaning as the number of classes, the average precision for class k.

Precision which is the parameter for detecting accuracy is used to calculate how fast the objects are predicted. There is also another reason to use Precision here. It can highlight how good the model is at predicting the positive class.

$$\text{Precision(P)} = \frac{TP}{TP+FP} \qquad (2)$$

The recall will set the object or layer detection quality. This means that the recall or the sensitivity that calculates the positive rate is actually found correctly.

$$\text{Recall (R)} = \frac{TP}{TP+FN} \qquad (3)$$

In the context of our evaluation, TP, FN, and FP signify true positives, false negatives, and false positives, respectively. TP (True Positive) represents the number of correctly detected objects, while FP (False Positive) indicates the count of incorrect detections of objects from different classes, resulting in false alarms. Conversely, FN (False Negative) denotes instances where objects that should have been detected are missed, and TN (True Negative) indicates the correct absence of detections when no objects were present. These metrics play a crucial role in assessing the performance of our vehicle detection system.

$$\text{F1-score} = \frac{2*\text{Precision}*\text{Recall}}{\text{Precision}+\text{Recall}} \qquad (4)$$

The F1-score, unlike the arithmetic mean, is determined by taking the harmonic mean of precision and recall. With a range between zero and one, the F1-score serves as a measure of the accuracy in object detection, where higher values signify greater accuracy.

4.3.3 Intersection over Union (IoU)

Within the domain of object detection, the Intersection over Union (IoU) [81] stands as a commonly utilized evaluation metric. Its purpose is to measure the extent of overlap between two regions, enabling a comprehensive assessment of our detector's performance in relation to the ground-truth. The intricate definition of IoU is visually illustrated in Figure 24, contributing to the meticulous evaluation of our advanced vehicle detection system.

4.3.4 Average Precision (AP)

The Average Precision (AP) stands as a multifaceted and widely-used evaluation metric in the realm of object detection. It takes into account essential aspects like precision, recall, and Intersection over Union (IoU) to offer a comprehensive assessment of our detector's performance.

To compute the AP, we put our detector to the test, making predictions on each ground-truth object and evaluating the accuracy of these predictions based on their IoU values against a predefined threshold. This meticulous process yields a thorough understanding through the calculation of the confusion matrix, precision, and recall. Moreover, the inclusion of a confidence score reflects our detector's level of certainty in its predictions.

To derive the AP, we systematically vary the score threshold from 0 to 1 with a step size of 0.05, computing corresponding precision and recall values. This iterative process generates a precision-recall (PR) [82] curve, and the AP is then calculated as the area under this curve. This approach provides an insightful and holistic evaluation of our advanced vehicle detection system, enabling us to make data-driven improvements for optimal performance.

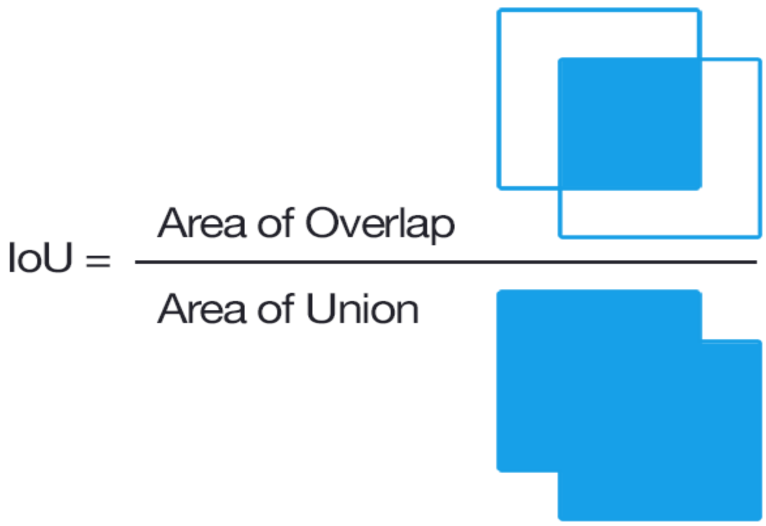

Figure 24. Illustration of IoU

In this chapter, proposed modified model is introduced. The architecture as well as the structure of the method are discussed in Section 4.1, including the models and the reasoning behind the design choices made. Section 4.2 focuses on the dataset used, addressing any challenges encountered and the preprocessing steps taken. Finally, Section 4.3 describes into the evaluation metrics and methodology applied to the model. Moving forward, the accuracy and inference time results of the proposed models will be presented in Chapter 5.

CHAPTER – 5

RESULT AND DISCUSSION

This section of the report showcases the results obtained and engages in an insightful discussion surrounding them.

RESULT AND DISCUSSION

5

The Result and Discussion chapter serves as a critical part to the study, The study also carried out experiment result that is obtained from the discussion and presentation in this chapter. The chapter presents the results of new proposed object detection specially vehicle detection methods and materials, including their performance on trained datasets and evaluation metrics. The results are compared with the performance and analyzed identifying the best of the proposed methods. This chapter also helps to draw conclusions and make recommendations, contributing to the advancement of the field of vehicle detection.

5.1 Results

The result describes an analysis of proposed versions of the YOLOv5 model, which is a popular vehicle detection algorithm. The analysis which is focused on the accuracy of the model in identifying vehicles in images. The YOLOv5 models were trained on 3001 images for 50 epochs, and the accuracy of the mAP, recall, and precision was evaluated. The mAP, recall, and precision all are metrics used commonly evaluating the performance of object detection models. Finally the results of the analysis which is seen that our proposed YOLOv5 models performed well in identifying vehicles, with high values for mAP, recall, and precision.

5.1.1 YOLOv5 s

Figure 25 used to illustrate the training of YOLOv5s models is a line plot of accuracy(mAp), recall and precision versus the number of epochs. The number of epochs are represented in the x-axis (i.e., in the entire training dataset the number of times the model has gone through), while the other axis that means axis -y that axis represents the model accuracy on the validation set (i.e., a portion of the dataset that is used to evaluate the performance of the model).The figure 25 usually starts with a low accuracy and gradually increases to the number of epochs. However, the number of epochs continues to increase, the accuracy may reach a plateau or even decrease, indicating that overfitting to the training data. This figure is useful for identifying optimal numbers of epoch training the model and for monitoring the model's performance during training.

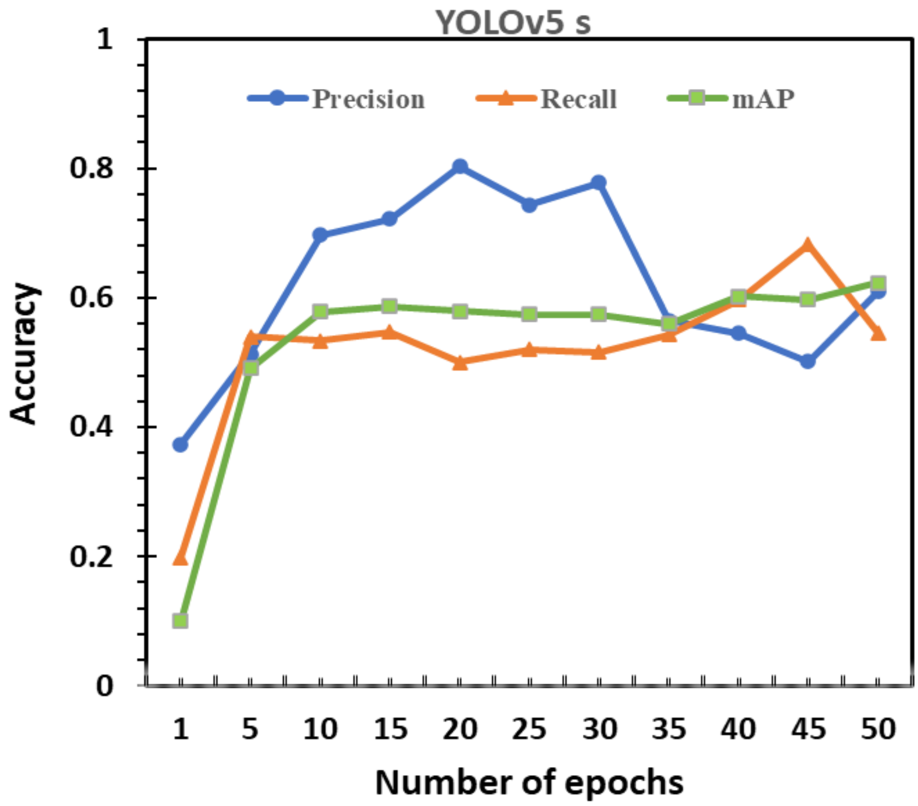

Figure 25. YOLOv5s Accuracy.

In YOLOv5s model, the model is performing well on the training dataset by measuring the loss function, where better performance are indicated by the lower values. Figure 26 may show the line plot of the train loss (i.e., the training set loss) versus the number of epochs. The axis both x and y represent the number of epochs and the value of the loss function.

Similarly, Figure 27 which is seen below the line plot of loss in the validation (i.e., the validation set loss) versus the epochs number. This figure is useful for performance monitoring of the model during training and training data overfitting for preventing. Both figures 25,26 typically show that the model is learning and improving by indicating a decreasing trend in the loss as the number of epochs increases.

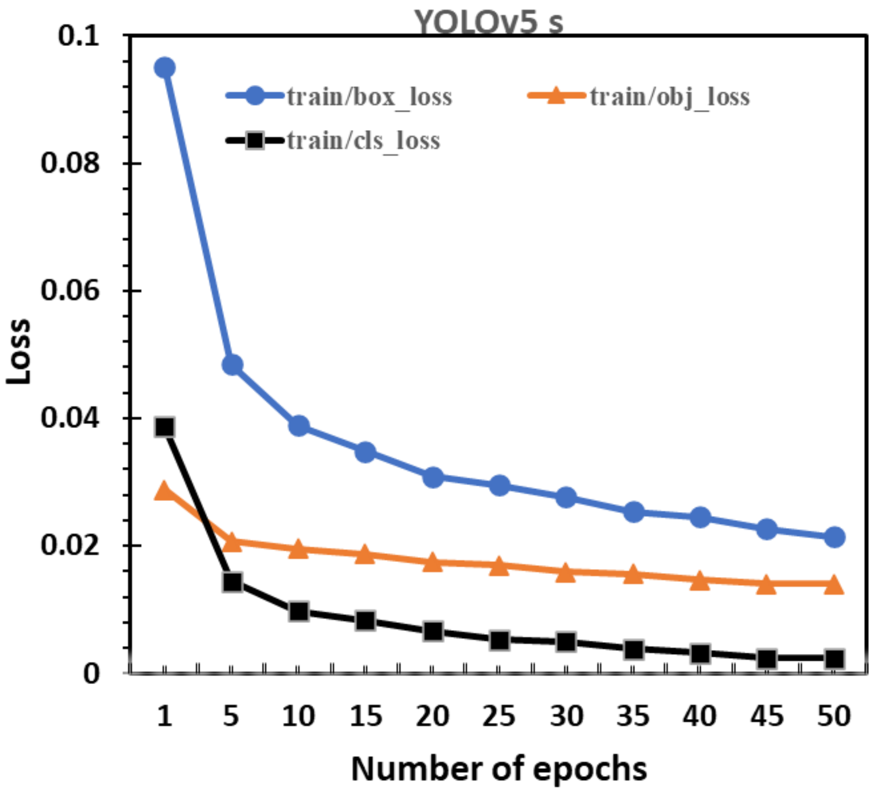

Figure 26. YOLOv5s train loss .

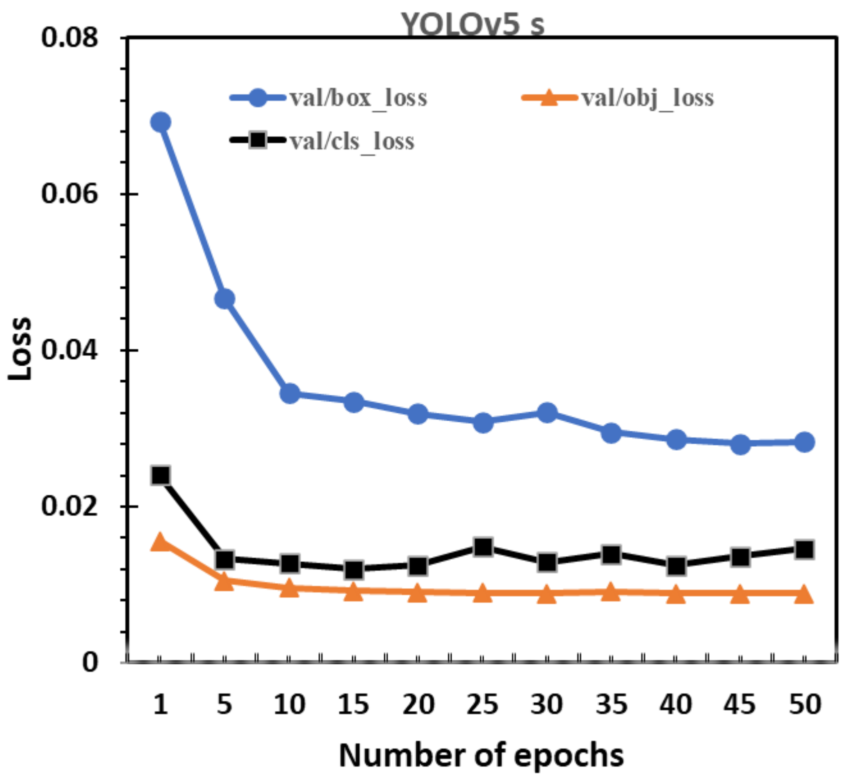

Figure 27. YOLOv5s val_ loss .

5.1.2 Proposed _ YOLOv5

In our Proposed YOLOv5s algorithm the accuracy and losses are seen in figure 28,29 and 30 respectively used for object detection. With respect to the number of epoch. the accuracy, training losses, and validation losses are described, we typically use a line plot. The plot of accuracy versus the number of epochs are seen in Figure 28, which is an increasing trend as increases the number of epochs, the model which indicating that is improving its accuracy on the training and validation sets. However, after reaching the optimal number of epochs, the accuracy may start to plateau or even decrease if the model is overfitting to the training data. The plot of training losses versus the number of epochs shows in figure 29. The loss function value of the loss function on the training set, which should decrease as the model is trained for more epochs. Similarly, the plot of validation losses versus the number of epochs shows in figure 30 the value of the loss function on the validation set, which should also decrease, but may eventually start to increase if the model is overfitting.

However, in general, we expect to see a decreasing trend in both the training and validation losses as the number of epochs increases, and an increasing trend in accuracy until reaching an optimal number of epochs. Monitoring the accuracy metrics, as well as the train and validation losses, with respect to the number of 50 epochs is essential to ensure that the model is improving.

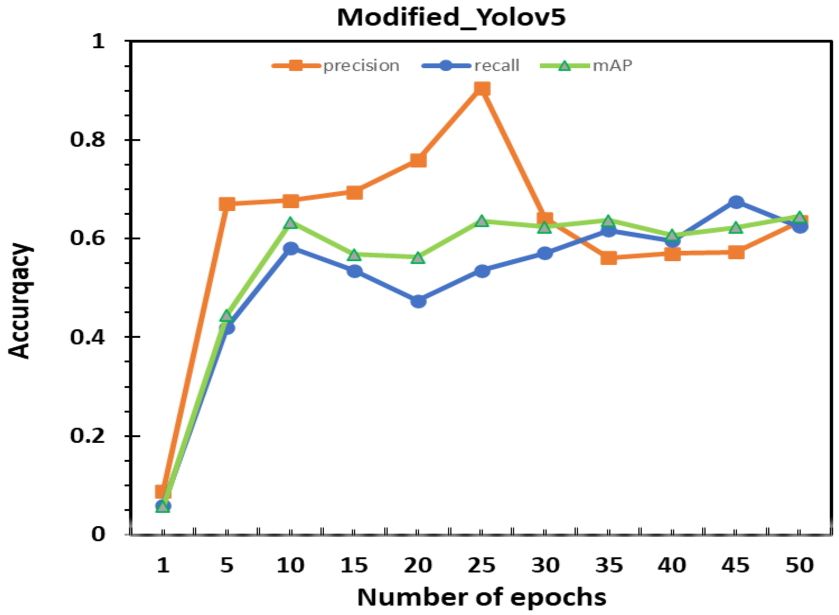

Figure 28. Proposed_YOLOv5 Accuracy

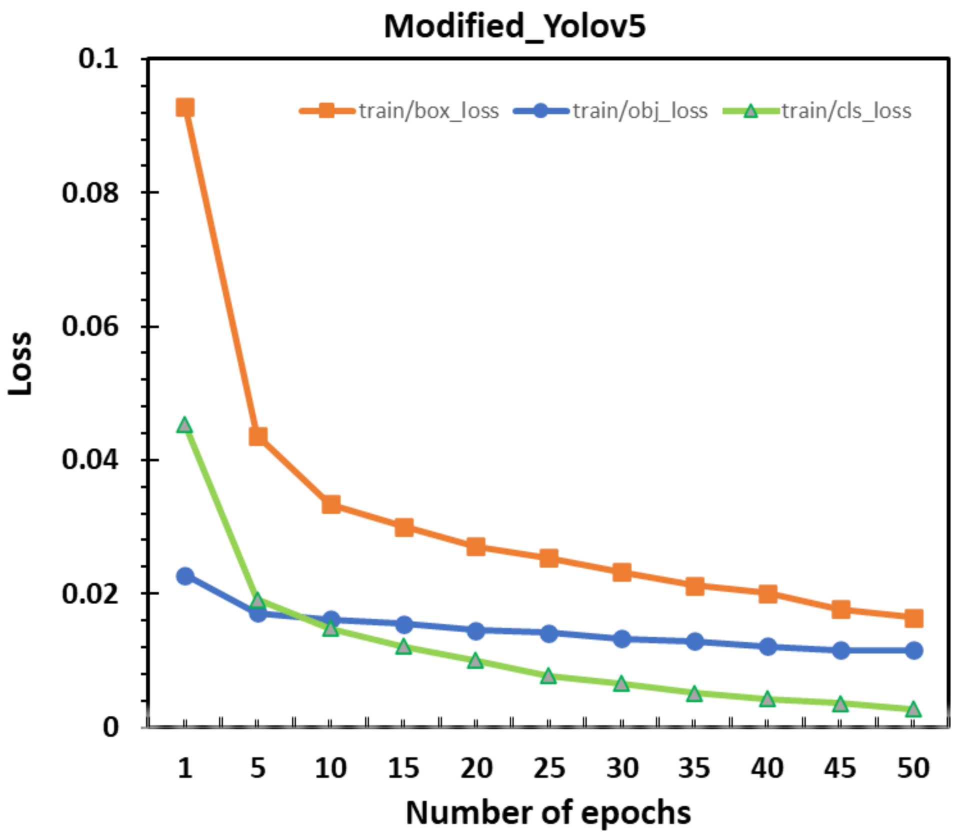

Figure 29. Proposed_YOLOv5m train_ loss .

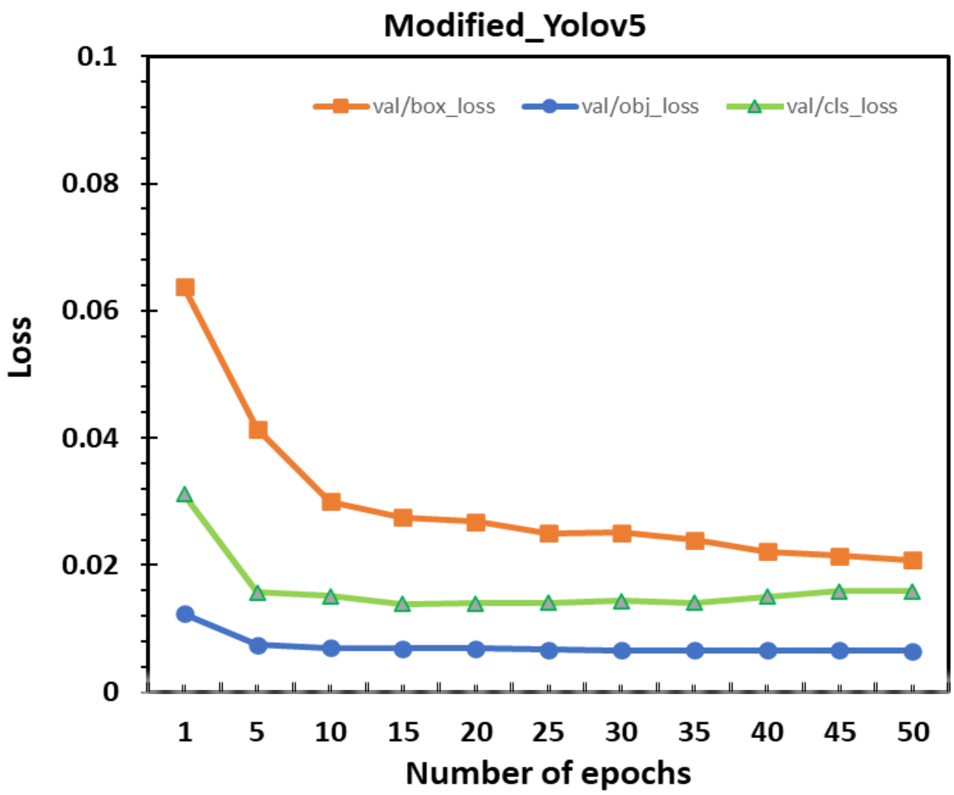

Figure 30. Proposed_YOLOv5m val_Loss.

The mAP values for the YOLOv5s and Modified_YOLOv5 models were 61.4%, and 67.4%, respectively. Comparison of the models performance is given in table 3. The analysis also found that the Modified_YOLOv5 model generally performed better than the other model in identifying vehicles. This means that the Modified_YOLOv5 model is more accurate and reliable in detecting vehicles in images compared to the other versions of the model.

Table 3. Comparison of the models performance

Model Name	Precision	Recall	mAP
YOLOv5s	0.649	0. 531	0.614
Modified_YOLOv5	0.728	0.652	0.674

In this project, we chose Python as the primary programming language for its user-friendly nature and rich library support. Numpy, matplotlib, and Scikit-Learn are utilized for various tasks, such as data processing, visualization, and model creation.

5.2 Comparison Results of the models

The overall outcome of the models we have discussed is summarized below in Figure 31. After careful consideration, we have concluded that proposed_YOLOv5 is the best model due to its exceptional performance. The comparison is based on the accuracy of the YOLOv5 object detection algorithm. The accuracy values reported are as follows: YOLOv5s has an accuracy of 61.40%, and our proposed YOLOv5 has an accuracy of 67.40%. These values indicate how well of the algorithm performs in detecting and identifying objects in an image or video. The higher the accuracy value, the better the algorithm is at correctly identifying objects. Therefore, our YOLOv5 is the most accurate versions. It's worth noting that accuracy is not the only metric that should be considered when evaluating an object detection algorithm. Other important factors to consider include speed, memory usage, and ease of use, among others. Ultimately, the choice of which version of YOLOv5 to use will depend on the specific requirements of the project at hand. As we can observe, the accuracy of the models we get the accuracy YOLOv5s 61.40%, and proposed_YOLOv5 67.40%, and it is clear that new proposed_YOLOv5 has the highest accuracy than other.

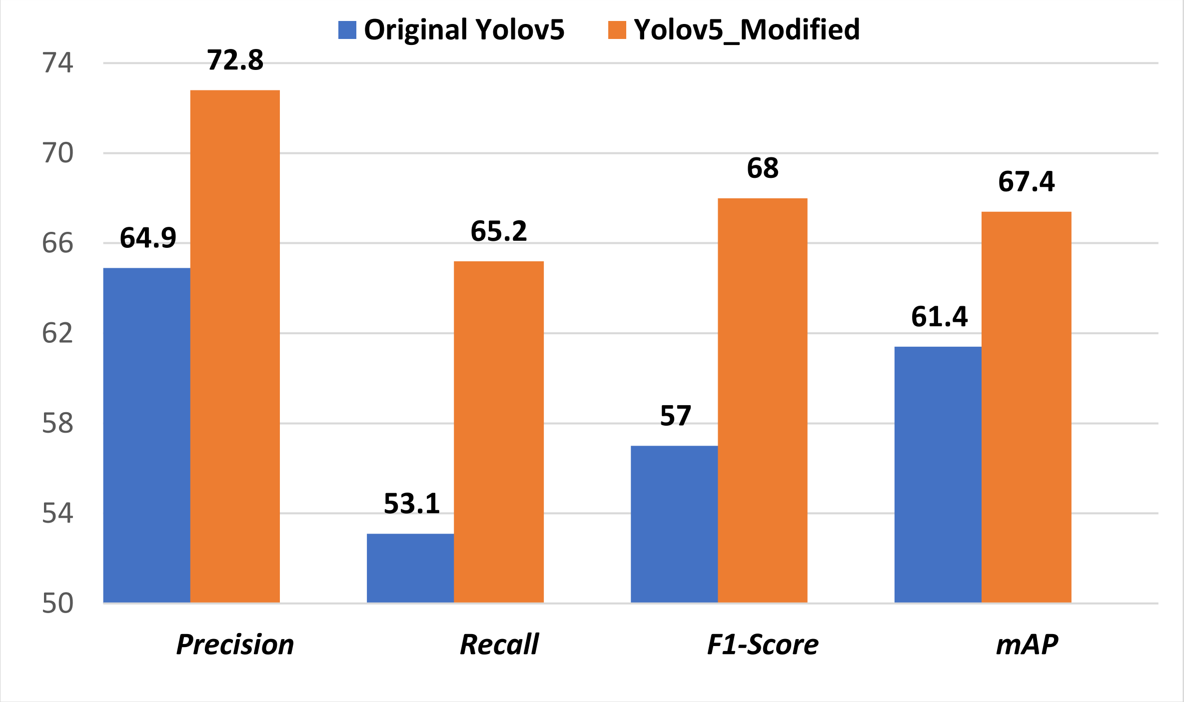

Figure 31. Accuracy Comparison Results of the models

5.3 Time Complexity for YOLOv5

YOLOv5 is primarily concerned with object detection, including vehicle detection. Its computational complexity depends on several factors, including the input image size and the number of objects (vehicles) to be detected in an image. YOLOv5, like many deep neural networks, is primarily concerned with the inference time complexity, which is a measure of how long it takes to process an input through the network.

In the case of YOLOv5, the inference time complexity can be roughly described as O(N), where N is the number of objects or bounding boxes detected in an image. This is because YOLO models process an image in a single forward pass and predict bounding boxes for all objects in the image simultaneously.

The computational complexity of the YOLOv5s model for vehicle detection can be characterized as follows:

Input Image Size: The input image size significantly affects the computational complexity. YOLOv5s operates by executing a singular forward pass through the neural network to analyze an image. As a result, its computational complexity scales linearly with the image's pixel count. If it is necessary to increase the image size by a factor of X, the computational time will roughly increase by that factor as well.

Number of Objects (Vehicles): YOLOv5s detects all objects (vehicles) in the image simultaneously. The computational complexity scale is roughly linear fashion with the quantity of detected objects. If there are more vehicles in the image, it will take longer to process them.

However, it's essential to note that the actual inference time also depends on other factors:

Model Architecture and Size: Different YOLOv5 variants (e.g., YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x) have different model architectures and sizes. Larger models with more parameters generally take longer to perform inference.

Hardware: The inference time will vary depending on the hardware used. GPUs are typically faster than CPUs for deep learning inference. Additionally, specialized hardware like TPUs or inference accelerators can further speed up inference.

Batch Size: If you perform inference on multiple images in a batch, it can lead to more efficient GPU utilization and potentially reduce the overall inference time.

Optimizations: Model optimizations, such as quantization or model pruning, can also impact inference time.

In summary, YOLOv5's time complexity is typically O(N) with additional factors that affect the actual inference time, making it essential to consider various aspects when evaluating its performance for a specific application.

In terms of Big O notation:

For the input image size, YOLOv5s can be considered to have a computational complexity of O(N), where N represents the number of pixels in images. This correlation arises because the processing time for an image directly aligns with the number of pixels it contains.

For the number of detected objects (vehicles), YOLOv5s can be considered to have a computational complexity of O(M), where M represents the number of detected objects. This is because the time it takes to process objects scales linearly with the number of objects.

These complexities are approximate and depend on various factors such as hardware, optimizations, and the specific implementation details. Additionally, YOLOv5s is optimized for real-time object detection and is designed to provide fast inference times, making it suitable for tasks like vehicle detection in real-world applications.

5.4 Comparison with Previous YOLO Models

In order to ensure a reliable evaluation of the study, it is essential to compare the performance of ourYOLOv5 model with previous iterations. Specifically, the mean Average Precision (mAP) metric was utilized for comparison against YOLOv3 and YOLOv4 models. Each experiment was conducted independently to maintain consistency. The results of these comparisons are presented in Table 4, showcasing the data outcome values for easy reference. From other researchers outcome, we try to compare. The performance of our proposed YOLOv5 and the previous yolo performance are analyzed and hopefully we get better result due to added extra convolution layer in the head and backbone section. Upon scrutinizing the final values, it was evident that our YOLOv5 model outperformed all other models, firmly establishing its supremacy in terms of accuracy and effectiveness.

Version	Anchor	Framework	Backbone	AP (%)
YOLO	No	Darknet	Darknet24	63.4
YOLOv2	Yes	Darknet	Darknet24	63.4
YOLOv3	Yes	Darknet	Darknet53	36.2
YOLOv4	Yes	Darknet	CSPDarknet53	43.5
YOLOv5	Yes	Pytorch	YOLOv5CSPDarknet	**55.8**
YOLOv6	No	Pytorch	EfficientRep	52.5
YOLOv7	No	Pytorch	YOLOv7Backbone	54.8
YOLOv8	No	Pytorch	CSPDarknet	53.9
Modified_ YOLOv5	Yes	Pytorch	YOLOv5CSPDarknet	**67.4**

Figure 32. Comparison with Previous YOLO Models

5.5 Practical Application

In order to validate the efficacy of the proposed method in this paper, an object search task was conducted using the YOLOv5 algorithm. The first step of the process was to identify the target object, which was followed by matching it and determining its precise location. Once the location of the object was obtained, it was labeled with the appropriate size information. Finally, all detected objects were sorted in a logical order. The actual results of the object detection task are presented in Figure 32, which shows the different types of vehicles that were detected within the detection frame. The output demonstrates the effectiveness of the YOLOv5 algorithm in detecting and recognizing different types of vehicles accurately. By employing the advanced YOLOv5 model, we have the capability to accurately detect vehicles that are traveling at speeds greater than 30km/hr. YOLOv5 (You Only Look Once) is a state-of-the-art object detection algorithm that excels in real-time object recognition tasks. Its efficient design allows for fast and precise vehicle detection, making it an ideal choice for monitoring high-speed vehicles on the road.

When applied to video or image data, which is seen in figure 33 and 34, YOLOv5 processes the input frames and generates bounding boxes around objects of interest, such as vehicles. Additionally, it assigns class labels to these objects, distinguishing them from other elements in the scene. With this information, we can extract specific attributes, including the speed of detected vehicles.

To identify vehicles traveling above the 30km/hr threshold, we can incorporate additional speed estimation techniques. These may involve leveraging motion analysis algorithms, utilizing optical flow calculations, or even employing dedicated speed measurement sensors. By combining the power of YOLOv5's object detection with speed estimation methods, we can effectively identify and track vehicles that exceed the specified speed limit. By implementing such a system, we can enhance road safety and enforce speed regulations. It enables us to monitor high-speed vehicles, gather valuable data on their behavior, and take appropriate actions when necessary. With the detailed insights provided by YOLOv5, we can contribute to a more efficient traffic management system and ensure a safer environment for all road users.

Overall, the object search task performed using YOLOv5 provided reliable results that confirmed the accuracy and effectiveness of the proposed method. This algorithm has a wide range of applications, including but not limited to, traffic monitoring, parking lot management, toll collection, and autonomous driving.

Figure 33. Visualization results from label dataset

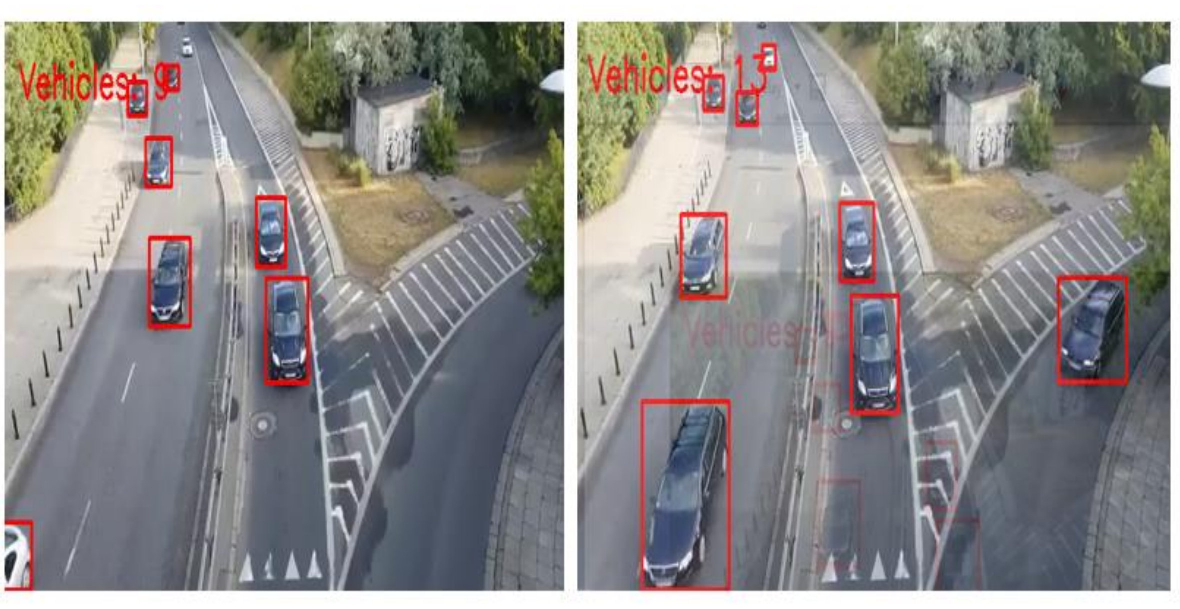

Figure 34. Visualization results from Video Frame

```
Vehicle 1: Speed = 142.88 km/hr
Vehicle 2: Speed = 104.77 km/hr
Vehicle 5: Speed = 124.27 km/hr
Vehicle 6: Speed = 130.74 km/hr
Vehicle 1: Speed = 143.73 km/hr
Vehicle 2: Speed = 103.92 km/hr
Vehicle 6: Speed = 135.23 km/hr
Vehicle 4: Speed = 124.59 km/hr
Vehicle 5: Speed = 125.25 km/hr
Vehicle 3: Speed = 102.61 km/hr
Vehicle 3: Speed = 101.80 km/hr
Vehicle 5: Speed = 99.30 km/hr
Vehicle 6: Speed = 145.29 km/hr
Vehicle 1: Speed = 158.66 km/hr
Vehicle 3: Speed = 141.23 km/hr
Vehicle 5: Speed = 148.54 km/hr
Vehicle 1: Speed = 157.79 km/hr
Vehicle 3: Speed = 162.89 km/hr
Vehicle 4: Speed = 117.32 km/hr
```

Figure 35. Speed detection from Video Frame

CHAPTER – 6

CONCLUSION AND FUTURE WORKS

Offers recommendations for future improvement of the research topic

6.1 Conclusion

6.2 Future Works

6

CONCLUSION AND FUTURE WORKS

In this chapter, we present the conclusion of our research and provide an overview of the results obtained from the study. The main aim of this chapter is to summarize the key findings and present the implications of the study for both theory and practice. We also outline the limitations of the study and discuss the potential avenues for future work.

6.1 Conclusion

The significance of vehicle detection and tracking, a subset of object detection, is continually increasing in traffic and urban areas. This technology finds versatile applications across various fields, contributing to improved human comfort and an enhanced quality of life. Industries, smart cities, government, research, academia, and environmental sectors are among the domains benefiting from object detection advancements.

This paper specifically focuses on vehicle detection methods based on the YOLOv5 neural network. A custom dataset comprising diverse subway scenes with various vehicles is prepared, followed by training multiple YOLOv5 models. To gauge the effectiveness of the model detection, a dedicated test set is utilized. The obtained results are compared with the YOLOv5s and Modified_YOLOv5 models, offering valuable insights for optimizing vehicle detection performance and fostering progress in this vital field.

Through training, the Modified_YOLOv5 model with the best characteristics was selected. Then, in validation, the best weights are applied to the Modified_YOLOv5 model and tested. We also get Compute time for YOLOv5s is 0.27 hours, Modified_YOLOv5 is 0.313 hours, As a result, compared to the YOLOv5s, Modified_YOLOv5 models, mAP increased more and the YOLOv5 Modified model was improved. It was analyzed that Modified_YOLOv5 was more efficient and had less loss than the YOLOv5s models. The rapid development in the car and traffic industries, coupled with the growth of the world population, has led to the need for various tools and technologies, particularly technology solutions, to manage traffic in cities and populated areas. With this research, our aim to contribute the improvement of the algorithms and models accuracy through the use of available techniques and tools.

6.2 Future Works

YOLO (You Only Look Once) has emerged as a widely used object detection algorithm, undergoing continuous improvements and updates. Among the latest advancements in the YOLO series is YOLOv5, which has showcased promising results, particularly in vehicle detection tasks. However, this field still holds untapped potential for further research and improvements in leveraging YOLOv5 for vehicle detection. To pave the way for progress, exploring these potential directions for future work becomes essential:

- **Improving object tracking:** YOLOv5 is mainly designed for object detection and not object tracking. Future work could focus on integrating YOLOv5 with object tracking algorithms to produce more accurate and robust vehicle tracking results.
- **Handling occlusion and partial visibility:** Vehicles in real-world scenes often occlude each other or are partially visible. Research could be done to improve the ability of YOLOv5 to handle these cases and produce more accurate results.
- **Increasing accuracy in small-sized objects:** Small-sized vehicles, such as motorcycles and bicycles, can be challenging to detect accurately with YOLOv5. Further research could be done to improve the accuracy of YOLOv5 modified for small objects detection.
- **Incorporating additional data sources:** Incorporating additional data sources, such as LIDAR and radar data, could improve the accuracy and robustness of YOLOv5 for vehicle detection tasks.
- **Real-time implementation:** Implementing YOLOv5 for real-time applications is a challenging task, as object detection algorithms are computationally expensive. Future work could focus on improving the computational efficiency of YOLOv5 to make it suitable for real-time applications.

Reflecting on the limitations of the study, there are areas of improvement and additional tasks that can be pursued. The field of vehicle detection and tracking is constantly evolving and researchers are continuously seeking ways to improve the accuracy of the process. To stay up-to-date with the latest advancements, it is important for researchers to closely follow the theoretical development in the field. When new theories are published, they should be incorporated into the algorithms the accuracy of detection and tracking are tested to improve.

Another avenue for future work could be to test the models using a larger and more diverse dataset that includes a wide variety of vehicle and non-vehicle images captured from different angles, locations, cameras, distances, etc.

Overall, there is a lot of potential for further research in the field of vehicle detection using YOLOv5, and many exciting advancements can be expected in the future. Additionally, other models can be included in the comparison to make the analysis more comprehensive and trustworthy.

REFERENCES

1. R.Timofte KZ, Luc Van Gool. Multi-view traffic sign detection, recognition, and 3D localisation. 2009 Workshop on Application of Computer Vision (WACV). 2009
2. Rahman, R., Bin Azad, Z., Bakhtiar Hasan, M. (2022). Densely- Populated Traffic Detection Using YOLOv5 and Non-maximum Suppression Ensembling. Proceedings of the International Conference on Big Data, IoT, and Machine Learning. Lecture Notes on Data Engineering and Communications Technologies, vol 95. Springer, Singapore. https://doi.org/10.1007/978-981-16-6636-0_43
3. S. Sivaraman and M. Trivedi, ―Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis‖, IEEE Transactions on Intelligent Transportation Systems, vol. 14, no. 4, pp. 1773-1795, 2013.
4. Y. Liu, B. Tian, S. Chen, F. Zhu, and K. Wang, ―A survey of vision- based vehicle detection and tracking techniques in ITS‖, 2013 IEEE International Conference on Vehicular Electronics and Safety (ICVES), pp. 72-77.
5. M. Lei, D. Lefloch, P. Gouton, and K. Madani, ―A Video-Based Real- Time Vehicle Counting System Using Adaptive Background Method‖, 2008 IEEE International Conference on Signal Image Technology and Internet Based Systems, pp. 523-528.
6. Jung, H.-K.; Choi, G.-S. Improved YOLOv5: Efficient Object Detection Using Drone Images under Various Conditions. Appl. Sci. 2022, 12, 7255. https://doi.org/10.3390/app12147255
7. S. Gupte, O. Masoud, R.F.K. Martin, and N.P. Papanikolo poulos, ―Detection and classification of vehicles‖, IEEE Transactions on Intelligent Transportation Systems, vol. 3, no. 1, pp. 37-47, 2002.
8. S. Ren KH, R. Girshic, J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017;39:1137- 49.
9. J. Dai YL, K. He, Jian Sun. R-FCN: Object Detection via Region-based Fully Convolutional Networks. 30th Conference on Neural Information Processing Systems (NIPS2016), Barcelona, Spain. 2016.
10. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, et al. SSD: Single Shot MultiBox Detector. Computer Vision – ECCV 2016 ECCV 2016 Lecture Notes in Computer Science. 2016;9905:21-37.

11. Wang, C., Zhang, Y., Zhou, Y. et al. Automatic detection of indoor occupancy based on improved YOLOv5 model. Neural Comput & Applic 35, 2575–2599 (2023). https://doi.org/10.1007/s00521-022-07730-3
12. A.Pramanik , S.Sarkar , Maiti J. A real-time video surveillance system for traffic pre-events detection. Accid Anal Prev. 2021 May;154:106019. doi: 10.1016/j.aap.2021.106019. Epub 2021 Mar 31. PMID: 33798983.
13. L.Shen ,L. You , Peng B, Zhang C. Group multi-scale attention pyramid network for traffic sign detection. March 2021Accident; Analysis and Prevention 154(5):106019 DOI:10.1016/j.aap.2021.106019
14. K. Simonyan AZ. Very deep convolutional networks for large- scale image recognition. CoRR. 2014.
15. Nepal, U.; Eslamiat, H. Comparing YOLOv3, YOLOv4 and YOLOv5 for Autonomous Landing Spot Detection in Faulty UAVs. Sensors 2022, 22, 464.
16. Liu, P.; Fu, H.; Ma, H. An end-to-end convolutional network for joint detecting and denoising adversarial perturbations in vehicle classification. Comput. Vis. Media 2021, 7, 217–227.
17. Lee, D.S. Effective Gaussian mixture learning for video background subtraction. IEEE Trans. Pattern Anal. Mach. Intell. 2005, 27, 827–832.
18. Ultralytics. YOLOv5 2020 [Available from: https://github.com/ultralytics/yolov5.
19. R. Girshick. Fast R-CNN". In: 2015 IEEE International Conference on Computer Vision (ICCV). 2015, pp. 1440{1448. doi: 10.1109/ICCV. 2015.169. url: https://ieeexplore.ieee.org/document/7410526.
20. S.Ren Faster R-CNN: Towards Real- Time Object Detection with Region Proposal Networks”. In: IEEE Transactions on Pat- tern Analysis and Machine Intelligence 39.6 (2017), pp. 1137{1149. doi: 10.1109/TPAMI.2016.2577031.url:https://ieeexplore.ieee.org/docment/7485869
21. J. Redmon et al.\You Only Look Once: Uni_ed, Real-Time Object Detection". In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).2016, pp.779{788doi:10.1109/CVPR.2016.91.url:https://ieeexplore.ieee.org/document/7780460.
22. G. Jocher ultralytics/yolov5: v5.0 - YOLOv5-P6 1280 models,AWS, Supervise.ly and YouTube integrations. Version v5.0. Apr. 2021. doi:10.5281/zenodo.4679653. url: https://doi.org/10.5281/zenodo.4679653.

23. Su, A., Sun, X., Liu, H., Zhang, X., & Yu, Q. (2015). Online cascaded boosting with histogram of orient gradient features for car detection from unmanned aerial vehicle images. Journal of Applied Remote Sensing, 9(1), 096063.
24. Viraktamath, D.S.V.; Yavagal, M.; Byahatti, R. Object Detection and Classification using YOLOv3. Int. J. Eng. Res. Technol. IJERT2021, 10. Available online: https://www.academia.edu/download/66162765/object_detection_and_classification_using_IJERTV1 0IS020078.pdf (accessed on 22 April 2022).
25. Yin, X.; Sasaki, Y.; Wang, W.; Shimizu, K. YOLO and K-Means Based 3D Object Detection Method on Image and Point Cloud. In Proceedings of the JSME Annual Conference on Robotics and Mechatronics (Robomec), online, June 2019; p. 2P1-I01.
26. Martinez-Alpiste, I.; Golcarenarenji, G.;Wang, Q.; Alcaraz-Calero, J.M. A dynamic discarding technique to increase speed and preserve accuracy for YOLOv3. Neural Comput. Appl. 2021, 33, 9961–9973. [CrossRef]
27. Chattopadhyay, D.; Rasheed, S.; Yan, L.; Lopez, A.; Farmer, J.; Brown, D.E. Machine Learning for Real-Time Vehicle Detection in All-Electronic Tolling System. In Proceedings of the 2020 Systems and Information Engineering Design Symposium (SIEDS), Charlottesville, VA, USA, 24 April 2020.
28. Kanistras, K., Martins, G., Rutherford, M. J., & Valavanis, K. P. (2015). Survey of unmanned aerial vehicles (UAVs) for traffic monitoring. Handbook of unmanned aerial vehicles, 2643-2666.
29. Wei, P., Lu, X., Tang, T., Li, C., & Song, J. (2015, August). A highway vehicle detection method based on the improved visual background extractor. In 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD) (pp. 1519-1524). IEEE.
30. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. arXiv 2016, arXiv:1612.08242.
31. Li, J.; Huang, S. YOLOv3 Based Object Tracking Method. Electron. Opt. Control 2019, 26, 87–93.
32. Bochkovskiy, A.; Chien-Yao, W.; Liao, H.Y.M. YOLOv4: Optimal speed and accuracy of object detection. arXiv 2020,arXiv:2004.10934.
33. Zhan, W.; Sun, C.; Wang, M.; She, J.; Zhang, Y.; Zhang, Z.; Sun, Y. An improved Yolov5 real-time detection method for small objects captured by UAV. Soft Comput. 2022, 26, 361–373. [CrossRef]

34. Rajput, S.K.; Patni, J.C.; Alshamrani, S.S.; Chaudhari, V.; Dumka, A.; Singh, R.; Rashid, M.; Gehlot, A.; AlGhamdi, A.S. Automatic Vehicle Identification and Classification Model Using the YOLOv3 Algorithm for a Toll Management System. Sustainability 2022, 14, 9163. https://doi.org/ 10.3390/su14159163
35. Zhang, Y.; Guo, Z.;Wu, J.; Tian, Y.; Tang, H.; Guo, X. Real-Time Vehicle Detection Based on Improved YOLO v5. Sustainability 2022, 14, 12274. https://doi.org/10.3390/ su141912274
36. Chen, X., & Meng, Q. (2015, November). Robust vehicle tracking and detection from UAVs. In 2015 7th International Conference of Soft Computing and Pattern Recognition (SoCPaR) (pp. 241-246). IEEE.
37. Han, S.; Han, Y.; Hahn, H. Vehicle detection method using haar-like feature on real time system. World Academy of Science, Engineering and Technology, v. 59, p. 455{459, 2009.
38. Dong, Z.; Wu, Y.; Pei, M.; Jia, Y. Vehicle type classi_cation using a semisupervised convolutional neural network. IEEE Transactions on Intelligent Transportation Systems, v. 16, n. 4, p. 2247{2256, 2015.
39. Kim, P. K.; Lim, K. T. Vehicle type classi_cation using bagging and convolutional neural network on multi view surveillance image. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 2017, p. 914{919.
40. Roecker, M. N.; Costa, Y. M. G.; Almeida, J. L. R.; Matsushita, G. H. G. Automatic vehicle type classi_cation with convolutional neural networks. In: 2018 25th International Conference on Systems, Signals and Image Processing (IWSSIP), 2018, p. 1{5.
41. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, p. 6517{6525.
42. Selbes, B.; Sert, M. Multimodal vehicle type classi_cation using convolutional neural network and statistical representations of MFCC. In: 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Lecce, Italy, 2017, p. 1{6.
43. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. Technical report, University of Washington, Seattle, USA, 2018. Dispon__vel em https://pjreddie.com/darknet/yolo/

44. Min Lin, Qiang Chen, and Shuicheng Yan. "Network in network." arXiv preprint arXiv:1312.4400, 2013.
45. Sergey Ioffe, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." arXiv preprint arXiv:1502.03167, 2015.
46. Jürgen Schmidhuber, “Deep learning in neural networks: An overview.” Neural networks: the official journal of the International Neural Network Society, 61, 2015: 85-117.
47. Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. Deep learning. Vol. 1. Cambridge: MIT press, 2016.
48. Kasper-Eulaers, M.; Hahn, N.; Berger, S.; Sebulonsen, T.; Myrland, Ø.; Kummervold, P.E. Short Communication: Detecting Heavy Goods Vehicles in Rest Areas in Winter Conditions Using YOLOv5. *Algorithms* **2021**, *14*, 114. https://doi.org/10.3390/a14040114
49. Malta, Ana, Mateus Mendes, and Torres Farinha. "Augmented reality maintenance assistant using yolov5." Applied Sciences 11.11 (2021): 4758.
50. Wan, Jingjing, Bolun Chen, and Yongtao Yu. "Polyp detection from colorectum images by using attentive YOLOv5." Diagnostics 11.12 (2021): 2264.
51. Yao, J., Qi, J., Zhang, J., Shao, H., Yang, J., & Li, X. (2021). A real-time detection algorithm for Kiwifruit defects based on YOLOv5. Electronics, 10(14), 1711.
52. Jia, W., Xu, S., Liang, Z., Zhao, Y., Min, H., Li, S., & Yu, Y. (2021). Real-time automatic helmet detection of motorcyclists in urban traffic using improved YOLOv5 detector. IET Image Processing, 15(14), 3623-3637.
53. Patel, K., Bhatt, C., & Mazzeo, P. L. (2022). Improved ship detection algorithm from satellite images using YOLOv7 and graph neural network. Algorithms, 15(12), 473.
54. Wang, Y., Hao, Z., Zuo, F., & Pan, S. (2021, September). A fabric defect detection system based improved yolov5 detector. In Journal of Physics: Conference Series (Vol. 2010, No. 1, p. 012191). IOP Publishing.
55. Wang, Y., Wang, H., & Xin, Z. (2022). Efficient detection model of steel strip surface defects based on YOLO-V7. IEEE Access, 10, 133936-133944.
56. Hussain, M., Al-Aqrabi, H., Munawar, M., Hill, R., & Alsboui, T. (2022). Domain feature mapping with YOLOv7 for automated edge-based pallet racking inspections. Sensors, 22(18), 6927.

57. Koza, J. R., Bennett, F. H., Andre, D., & Keane, M. A. (1996). Automated design of both the topology and sizing of analog electrical circuits using genetic programming. In Artificial Intelligence in Design’96 (pp. 151-170). Springer, Dordrecht.
58. Russell, S. J., & Norvig, P. (2016). Artificial intelligence: a modern approach. Malaysia; Pearson Education Limited.
59. Noh. S, Shim. D and Jeon. M, (2015), 'Adaptive Sliding-Window Strategy for Vehicle Detection in Highway Environments', IEEE IEEE Transactions on Intell. Transport.Syst., Ofor, e. (2018). Machine learning techniques for immunotherapy dataset classification (Master dissertation, Near East University).
60. UCBWiki. (2016, February). Reinforcement Learning with Function approximation. Retrieved April 22, 2018, from UCB WIki: http://wiki.ubc.ca /images /b/ bd/Rldiagram1.png
61. Aleksandr N. Grekov , Yurii E. Shishkin, Sergei S. Peliushenko, Aleksandr S. Mavrin, Application of the YOLOv5 Model for the Detection of Microobjects in the Marine EnvironmentarXiv:2211.15218 [cs.CV] (or arXiv:2211.15218v1 [cs.CV] for this version) https://doi.org/10.48550/arXiv.2211.15218
62. Zhao, J.; Li, C.; Xu, Z.; Jiao, L.; Zhao, Z.; Wang, Z. Detection of passenger flow on and off buses based on video images and YOLO algorithm. Multimed. Tools Appl. 2021, 3, 1–24.
63. F.Zhou, H. Zhao, Zhen Nie, ―Safety Helmet Detection Based on YOLOv5‖ , 2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA).
64. Ultralytics LLC. "YOLOv5: Object Detection with EfficientDet Backbone." GitHub, 2020, https://github.com/ultralytics/yolov5#introduction.
65. Wang, Xiaolong, Ross Girshick, Abhinav Gupta, and Kaiming He. "Non-local neural networks." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7794-7803.
66. Girshick, Ross, Jeff Donahue, Trevor Darrell, and Jitendra Malik. "Rich feature hierarchies for accurate object detection and semantic segmentation." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580-587.
67. Wang, C., and H. Zhang. "Research on Poultry Distribution and Retail System Based on Machine Learning." Journal of Physics: Conference Series, vol. 1829, no. 1, 2021, doi:10.1088/1742-6596/1829/1/012078.

68. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788
69. Nepal U, Eslamiat H. Comparing YOLOv3, YOLOv4 and YOLOv5 for Autonomous Landing Spot Detection in Faulty UAVs. Sensors (Basel). 2022 Jan 8;22(2):464. doi: 10.3390/s22020464. PMID: 35062425; PMCID: PMC8778480.
70. Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao. "YOLOv5: Improved real-time object detection." arXiv preprint arXiv:2011.08036 (2020).
71. Jocher, G., Wong, A., & Musa, M. (2021). YOLOv5: A Universal Object Detector. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2021, 214-215.
72. Tsung-Yi Lin et al "Focal Loss for Dense Object Detection". (2018),
73. Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg "SSD: Single Shot MultiBox Detector" (https://arxiv.org/abs/1512.02325)
74. Wang, G., Zhang, Z., Chen, W., Zhang, Y., Wang, Y., & Qiao, Y. (2021). YOLOv5nano: A Highly Customizable Object Detection Algorithm for Embedded Systems. IEEE Transactions on Circuits and Systems for Video Technology, 31(1), 424-438.
75. Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934.
76. Wang, G., Zhang, Z., Chen, W., Zhang, Y., Wang, Y., & Qiao, Y. (2020). YOLOv5: End-to-end real-time object detection with YOLO. arXiv preprint arXiv:2012.09626.
77. S. Ren, K. He, R. Girshick, and J. Sun, "Object Detection Networks on Convolutional Feature Maps," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 7, pp. 1476-1481, Jul. 2017.
78. S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," in Advances in Neural Information Processing Systems, 2015, pp. 91-99.
79. Zhou, S., Bi, Y., Wei, X. et al. Automated detection and classification of spilled loads on freeways based on improved YOLO network. Machine Vision and Applications 32, 44 (2021). https://doi.org/10.1007/s00138-021-01171-z

80. Neupane, D.; Seok, J. A Review on Deep Learning-Based Approaches for Automatic Sonar Target Recognition. Elecronics 2020, 9, 1972. DOI:10.3390/electronics9111972
81. R. Rahman, Z. Azad, and Md. B. Hasan "Densely- Populated Traffic Detection using YOLOv5 ans Non-Maximum Suppression Ensembling" arXiv:2108.12118v1 [cs.CV] 27 Aug 2021, DOI: 10.1007/978-981-16-6636-0_43
82. Li, S.; Li, Y.; Li, Y.; Li, M.; Xu, X. YOLO-FIRI: Improved YOLOv5 for Infrared Image Object Detection. IEEE Access 2021, 2021, 3120870. [CrossRef]

APPENDIX

```
import os
import glob as glob
import matplotlib.pyplot as plt
import cv2
import requests
import random
import numpy as np


TRAIN = True
# Number of epochs to train for.
EPOCHS = 50
```

```
if not os.path.exists('train'):
    !curl -L "https://app.roboflow.com/ds/uTo78m76Pw?key=y4NPqPzoij" > roboflow.zip; unzip roboflow.zip; rm roboflow.zip

    dirs = ['train', 'valid']

    for i, dir_name in enumerate(dirs):
        all_image_names = sorted(os.listdir(f"{dir_name}/images/"))
        for j, image_name in enumerate(all_image_names):
            if (j % 2) == 0:
                file_name = image_name.split('.jpg')[0]
                os.remove(f"{dir_name}/images/{image_name}")
                os.remove(f"{dir_name}/labels/{file_name}.txt")
def download_file(url, save_name):
    url = url
    if not os.path.exists(save_name):
        file = requests.get(url)
        open(save_name, 'wb').write(file.content)
    else:
```

```
        print('File already present, skipping download...')
```

```
class_names = ['Ambulance', 'Bicycle', 'Bus', 'Car', 'Motorcycle', 'Truck']
colors = np.random.uniform(0, 255, size=(len(class_names), 3))
```

```
# Function to convert bounding boxes in YOLO format to xmin, ymin, xmax, ymax.
def yolo2bbox(bboxes):
    xmin, ymin = bboxes[0]-bboxes[2]/2, bboxes[1]-bboxes[3]/2
    xmax, ymax = bboxes[0]+bboxes[2]/2, bboxes[1]+bboxes[3]/2
    return xmin, ymin, xmax, ymax
```

```
def plot_box(image, bboxes, labels):
    # Need the image height and width to denormalize
    # the bounding box coordinates
    h, w, _ = image.shape
    for box_num, box in enumerate(bboxes):
        x1, y1, x2, y2 = yolo2bbox(box)
        # denormalize the coordinates
        xmin = int(x1*w)
        ymin = int(y1*h)
        xmax = int(x2*w)
        ymax = int(y2*h)
        width = xmax - xmin
        height = ymax - ymin

        class_name = class_names[int(labels[box_num])]

        cv2.rectangle(
            image,
            (xmin, ymin), (xmax, ymax),
            color=colors[class_names.index(class_name)],
            thickness=2
        )
```

```
        font_scale = min(1,max(3,int(w/500)))
        font_thickness = min(2, max(10,int(w/50)))

        p1, p2 = (int(xmin), int(ymin)), (int(xmax), int(ymax))
        # Text width and height
        tw, th = cv2.getTextSize(
            class_name,
            0, fontScale=font_scale, thickness=font_thickness
        )[0]
        p2 = p1[0] + tw, p1[1] + -th - 10
        cv2.rectangle(
            image,
            p1, p2,
            color=colors[class_names.index(class_name)],
            thickness=-1,
        )
        cv2.putText(
            image,
            class_name,
            (xmin+1, ymin-10),
            cv2.FONT_HERSHEY_SIMPLEX,
            font_scale,
            (255, 255, 255),
            font_thickness
        )
    return image
```

```
# Function to plot images with the bounding boxes.
def plot(image_paths, label_paths, num_samples):
    all_training_images = glob.glob(image_paths)
    all_training_labels = glob.glob(label_paths)
    all_training_images.sort()
    all_training_labels.sort()
```

```
    num_images = len(all_training_images)

    plt.figure(figsize=(15, 12))
    for i in range(num_samples):
        j = random.randint(0,num_images-1)
        image = cv2.imread(all_training_images[j])
        with open(all_training_labels[j], 'r') as f:
            bboxes = []
            labels = []
            label_lines = f.readlines()
            for label_line in label_lines:
                label = label_line[0]
                bbox_string = label_line[2:]
                x_c, y_c, w, h = bbox_string.split(' ')
                x_c = float(x_c)
                y_c = float(y_c)
                w = float(w)
                h = float(h)
                bboxes.append([x_c, y_c, w, h])
                labels.append(label)
        result_image = plot_box(image, bboxes, labels)
        plt.subplot(2, 2, i+1)
        plt.imshow(result_image[:, :, ::-1])
        plt.axis('off')
    plt.subplots_adjust(wspace=0)
    plt.tight_layout()
    plt.show()
```

```
# Visualize a few training images.
plot(
    image_paths='train/images/*',
    label_paths='train/labels/*',
    num_samples=4,
)
```

```
def set_res_dir():
    # Directory to store results
    res_dir_count = len(glob.glob('runs/train/*'))
    print(f"Current number of result directories: {res_dir_count}")
    if TRAIN:
        RES_DIR = f"results_{res_dir_count+1}"
        print(RES_DIR)
    else:
        RES_DIR = f"results_{res_dir_count}"
    return RES_DIR
```

```
def monitor_tensorboard():
    %load_ext tensorboard
    %tensorboard --logdir runs/train
```

```
if not os.path.exists('yolov5'):
    !git clone https://github.com/ultralytics/yolov5.git
```

```
%cd yolov5/
!pwd
```

```
!pip install -r requirements.txt
```

```
RES_DIR = set_res_dir()
if TRAIN:
    !python train.py --data ../data.yaml --weights yolov5s.pt \
    --img 416 --epochs {EPOCHS} --batch-size 16 --name {RES_DIR}
```

```
RES_DIR = set_res_dir()
if TRAIN:
    !python train.py --data ../data.yaml --weights modified_yolov5.pt \
    --img 416 --epochs {EPOCHS} --batch-size 16 --name {RES_DIR}
```

List of Publication

1. Md. Milon Rana, Md. Dulal Haque and Md. Mahabub Hossain, “Comparative Study of Vehicle Detection with Different YOLOv5 Algorithms”, 2nd International Symposium on Sustainable Energy and Technological Advancements ISSETA 2023, National Institute of Technology Meghalaya, India, to be held during 24th-25th Feb. 2023. The accepted, registered and presented paper in the symposium will be published in the Springer Book Series of “Digital Communication and Soft Computing Approaches towards Sustainable Energy Developments: Proceedings of ISSETA 2023” Indexed by Scopus and ESCI.