Classification and Detection of Vehicles in Images and Videos by Employing Transfer Learning on YOLO Algorithm

Author

Annam Farid

00000274599

Supervisor

Dr. Farhan Hussain

DEPARTMENT OF COMPUTER ENGINEERING

COLLEGE OF ELECTRICAL & MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

ISLAMABAD

MARCH 2022

Classification and Detection of Vehicles in Images and Videos by Employing Transfer Learning on YOLO Algorithm

Author

Annam Farid

00000274599

A thesis submitted in partial fulfillment of the requirements for the degree of

MS Computer Engineering

Thesis Supervisor

Dr. FARHAN HUSSAIN

Thesis Supervisor's Signature: _____

DEPARTMENT OF COMPUTER ENGINEERING

COLLEGE OF ELECTRICAL & MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY, ISLAMABAD

MARCH 2022

Declaration

I certify that this research work titled "*Classification and detection of vehicles in images and videos by employing transfer learning on YOLO algorithm"* is my own work. The work has not been presented elsewhere for assessment. The material that has been used from other sources it has been properly acknowledged / referred.

Signature of Student

Annam Farid

00000274599

Language Correctness Certificate

This thesis has been read by an English expert and is free of typing, syntax, semantic, grammatical, and spelling mistakes. Thesis is also according to the format given by the university.

Signature of Student

Annam Farid

00000274599

Signature of Supervisor

Dr. FARHAN HUSSAIN

Copyright Statement

Acknowledgements

All praise and glory to Almighty Allah (the most glorified, the highest) who gave me the courage, patience, knowledge, and ability to carry out this work and to persevere and complete it satisfactorily. Undoubtedly, HE eased my way and without HIS blessings I can achieve nothing.

I would like to express my sincere gratitude to my advisor Dr. Farhan Hussain for boosting my morale and for his continual assistance, motivation, dedication, and invaluable guidance in my quest for knowledge. I am blessed to have such a co-operative advisor and kind mentor for my research.

Along with my advisor, I would like to acknowledge my entire thesis committee for their cooperation and prudent suggestions.

My acknowledgement would be incomplete without thanking the biggest source of my strength, my family. I am profusely thankful to my beloved parents who raised me when I was not capable of walking and continued to support me throughout in every field of my life, my supportive brother who stand beside me and hold me back, my loving sisters who were with me through my thick and thin.

Finally, I would like to express my gratitude to all my friends and the individuals who have encouraged and supported me through this entire period.

Dedicated to my beloved parents, siblings, and adored husband whose tremendous support and cooperation led me to this accomplishment.

Abstract

This thesis aims to target the detection and classification of vehicles in images and videos of local traffic of Rawalpindi/Islamabad by utilizing the YOLO-v5 architecture. The YOLO-v5 has surpassed other traditional object detection algorithms. The YOLO-v5 is computationally faster in comparison of other YOLO algorithms. We propose to employ transfer learning to fine tune the weights of the pre-trained YOLO-v5 fine-tune the weights of the YOLO-v5 network that has already been trained network so that they are accustomed according to our local traffic patterns. For this purpose, extensive data sets of images and videos of the local traffic patterns were collected. These data sets were made comprehensive by targeting various attributes like high density traffic patterns, low density traffic patterns, occlusion, and various weather conditions. All of these data sets were manually annotated. By fine-tuning the pre-trained network weights with the help of our data sets we achieved better detection and classification results.

Object detection and recognition is one of the most difficult applications of computer vision, machine learning, and artificial intelligence, and it is widely employed in a variety of fields. For example, Robotics, security, surveillance, and to guide visually impaired people. Aforedescribed methods works differently with their network architectures with the main aim to detect multiple objects that appear in an image. With the rapid development of deep learning, many algorithms are consistently improving the relationship between video analysis and image understanding. We are optimistic that our developed method is one of the latest additions in this domain.

Key Words: Deep Neural Network, Object classification, Object detection, YOLO-v5

Table of Contents

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

Every minute, on average, at least one person dies in a motor vehicle accident. Every year, at least 10 million people are injured in automobile accidents, with two or three million of them being seriously injured. [4]. According to statistics, the primary threat that a motorist confronts is from other vehicles. As a result, building an on-board motor vehicle detection system to warn individuals about potential collisions has gotten a lot of interest [5]. The initial step with these systems is robust and reliable vehicle detection. Due to large interclass or intraclass variances in vehicle appearance, vehicle detection under a variety of settings is difficult. Vehicles come in a variety of shapes, sizes, and colors. A vehicle's appearance is determined by its attitude and is influenced by neighboring objects. Outdoor conditions that are difficult to control include non-uniform illuminations, unpredictable interactions amongst traffic participants, and the background. Because the processing rate is constrained by the vehicle speed, on-road vehicle detection necessitates faster processing. Another important consideration is the vehicle's ability to withstand motions and drifts.

Objects present in an image can be detected and identified by humans. Besides being fast and accurate, the human visual system also performs complex tasks, such as detecting obstacles and identifying several items with little conscious attention. It is now easier for computers to recognize and classify several items inside a picture with high accuracy thanks to the availability of capabilities like as massive data sets, faster GPUs, and better algorithms. [1].

The task of object detection and classification is to identify and classify specific objects in a digital image or video. Object detection is the process to locate the object, such as person, cars people, birds, animals, or anything that is available in an image, or a video as shown in Figure 1.1 [17]. In according to the detection of the object, later the object is classified as the respective class as living object, animal, or non-living objects. The specific class could be classified after detection using this technique. Also, in this technique the location of the object is also identified in the large number of undefined categories. Not only in images, but it has also been applied on video system.

Objects detection and classification aims for the detection and classification of the objects in accordance with their class labels in an image or video [1]. This domain has obtained considerable attention in important fields, for instance, Intelligent Transportation Systems (ITS) [1], traffic handling [2], and sports entertainments [3].

Figure 1. 1: Object detection example

Recent advances in machine learning methods have found substantial applications in the aforementioned fields. Recently, encouraging results are achieved in object detection in controlled environments. However, for uncontrolled environments, such as complex background, non-uniform illuminations, and multiple shapes, object detection becomes a challenging task [4].

In recent times, deep neural networks have gained polarity in various object detection and classification tasks, such as vehicles and surveillance [5]. For object detection and recognition, large number of algorithms, for instance, Regional Convolutional Neural Networks (R-CNNs), Support Vector Machine (SVM), and You Only Look Once (YOLO) have been proposed [6]. Depending upon the nature of the task, it is important to choose a right algorithm for specified application [7]. Comparing to the state-of-the-art algorithms, YOLO based methods have reported the high detection accuracy along with less execution time [35]–[40]. Typically, the YOLO based methods split an image into different grid sizes. Later, every cell in the grid calculates bounding boxes around object(s). in Figure 1.2 an example of object detection and classification has been shown.

Inspired from the aforementioned facts, the main focus of this study is development of a deep learning supervised model, which uses YOLO algorithm to detect objects on traffic scenes databases. Specifically, this paper investigates object classification and detection in context of vehicles, motorcycles, and pedestrians that appear in an image or video on dense Pakistani

highways. A real time image is inserted as an input, and a bounding box corresponding to all objects in the image, along with the class of object in each box, comes up as an output.

Figure 1. 2: Object detection and Classification example

1.1 Motivation

Recently, with rapid developments of urbanization, traffic blockage, incidents, and on road violations pose great challenges to traffic management systems. Computer vision and machine learning has attracted much attention in Intelligent Transportation Systems (ITSs). Computer vision techniques are used to gather traffic parameters to analyze traffic manners for surveillance. A crucial component of traffic surveillance is precise and reliable vehicle detection. In ITSs, there are still certain issues with vehicle detection. It's difficult to train a detection model because of the variety of vehicle appearances and positions. Similarly, complicated metropolitan environments, severe weather, and poor/strong illumination conditions significantly limit detection performance. Vehicles are partially/fully blocked-in extreme traffic congestion, allowing individual vehicles to combine into a single vehicle. As a result, learning parameters to detect vehicles is a critical challenge. In most cases, a detection approach with complicated parameters is impractical. Our motivation is to devise a method of performing vehicle classification that generalize well to our traffic patterns. A methos which could also embed into autonomous cars along with other surveillance and security operations. The one-of-a-kind strategies for object detection can be grouped into two classes [10]. The first class is algorithms primarily based on classifications. CNN and RNN come underneath this category. We have to choose the areas of interest from the photograph and then classify them the usage of CNN. A prediction is used for each and every area we select, so this technique is very gradual. Second, we have algorithms that are based totally on regressions. The YOLO approach comes beneath this category. In this case, we will no longer choose the preferred

areas from the image. The category and bounding box of the complete photo in one run of the algorithm is predicted, and a single neural network is used to discover a couple of objects. Therefore, inspired from the aforementioned facts, it is much needed to detect and classify vehicles in real-time along with reduced computational cost. The YOLO algorithm is quicker than different classification algorithms and is the focus of present study to detect and classify various vehicles that appear on Pakistani roads.

1.2 Problem Statement

Classification and detection of vehicles in images and videos have been trained on massive dataset via deep learning algorithm. They still need to be fine-tuned by employing transfer learning for using in our local environment. Hence, we need to augment these datasets with our local datasets. We target to compare the efficiency of our trained model with the baseline results.

1.2 Aims and Objective

Use transfer learning to fine tune weights of existing deep learning algorithm to classify and detect various vehicles in different weather conditions and traffic scenarios such as to detect vehicle with low illumination, partial occlusion, and large group of vehicles irrespective of size, shape, and color.

1.4 Thesis Structure

The layout of the proposed work is as follows:

Chapter 2 covers the background of Machine learning and Artificial Intelligence
Chapter 3 gives review of the literature and the significant work done by researchers in past few years for the classification and detection of vehicle
Chapter 4 entails the detail of the proposed methodology.
Chapter 5 introduces the databases used for evaluation purposes. All the experimental outcomes are mentioned in detail with all desired figures and tables.
Chapter 6 is the conclusion of the work done contributions and uncovers the future scope of this research.

CHAPTER 2

RELATED BACKGROUND

In computer vision, deep learning algorithms are widely used. Prior to discussing object detection related tasks, we will look at the basics of machine learning.

2.1 Evolution of AI: Deep Learning and Machine Learning

AI refers to artificial intelligence, which is system programmed by humans to operate human functions. This synthetic AI is included into computer structures to create AI structures that eventually function as "thinking machine" units. AI systems are designed by humans to make selections from historic or real-time statistics or both. AI systems can analyze and adapt as they collect records and make decisions. AI structures frequently include computing device learning, deep learning and records analytics with AI that allow smart choice making. This intelligence is now not human intelligence. It's the machine's best approximation to human talent [11].

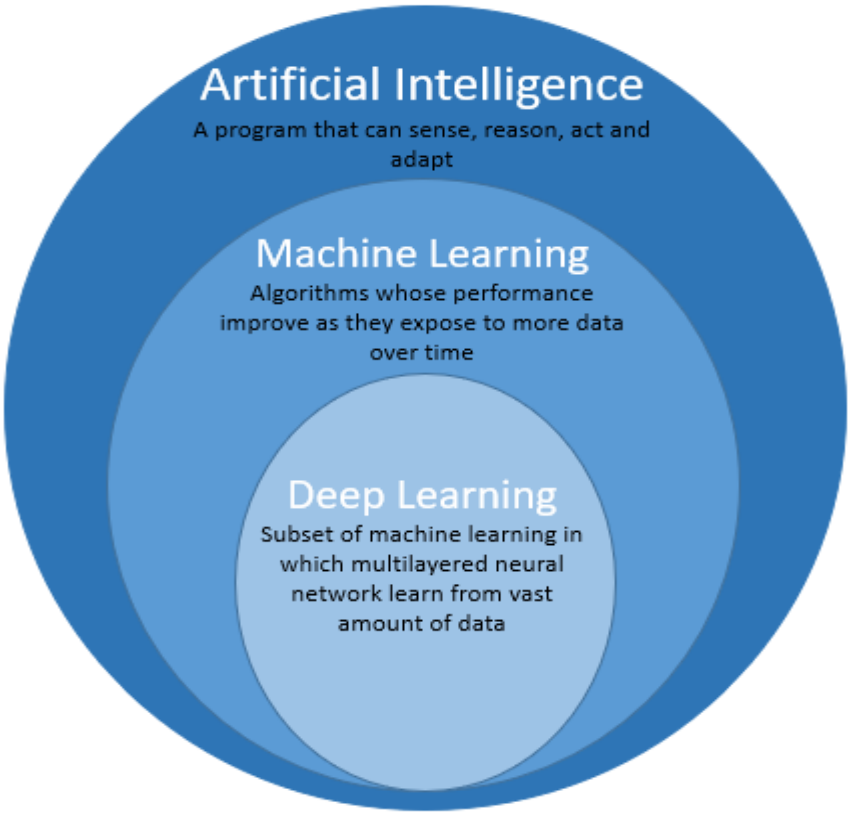


Figure 2. 1: Levels of Artificial Intelligence

As shown in Figure 2.1, AI's subset is machine learning, which entails the truth that we can construct intelligent machines that can examine based totally on a supplied dataset on its own. In addition to, you will be aware that Deep learning is a subset of Machine Learning the place comparable ML algorithms are used to educate Deep Neural Networks to obtain higher accuracy in these instances where the former was performing up to the mark.

2.1.1 Machine Learning

It is an application of artificial intelligence that gives the AI device the ability to analyze from the environment and apply that knowledge to make higher level decisions. To do this effectively, there are three classes of machine learning algorithms that make this possible known as Supervised Machine Learning, Unsupervised Machine Learning, and Reinforcement Learning explained here [12].

Some sort of pre-processing is almost continuously needed. Feature extraction, a technique of pre-processing the data into a new, simpler variable place. Every so often, it is impractical or no longer feasible to use the full-dimensional data directly. Instead, to extract interesting features from the data different detectors are programmed, and these facets are used as input to the algorithms. In the past, the detectors of features have been often hand-crafted. The trouble with this technique is that we do not constantly recognize in advance about the relevant and interesting features.

The trend in machine learning has been towards learning the feature detectors as well, which enables using the complete data. A vary of algorithms that machine learning makes use of to learn in iterations, describe, and improvement of data to predict greater outcomes. These algorithms use statistical strategies to spot patterns and then perform actions on these patterns.

Supervised Machine Learning

"Supervised" ability that an instructor assists the application during the coaching process: There is a coaching set with labeled dates. For example, you prefer to train the laptop, put red, blue and green socks in individual baskets. First you expose each of the objects to the machine and say what it is about. Then run the software on a validation set that examines whether the discovered functionality was correct or not. This type of learning is typically used for classification and regression. This type of learning is commonly used for classification and regression. This method's algorithms are Naïve Bayes, Support Vector Machine, Decision Tree, K-Nearest Neighbors, Logistic Regression, Linear and Polynomial regression etc.

Unsupervised Machine Learning

With unsupervised learning, you no longer allow the application to find patterns independently. Imagine you have a large laundry basket that the program needs to divide into different categories: socks, t-shirts, jeans, etc. This is called clustering, and unsupervised learning is regularly used to divide records based on similarities between agencies. Unsupervised knowledge acquisition is also suitable for in-depth analysis of information. For example, fraudulent transactions can be discovered, revenues and discounts can be predicted, or customer preferences can be analyzed primarily based on their search history. The programmer no longer understands what he is trying to find, but there are certainly patterns that the machine can pick up. This method's algorithms are K-means clustering, DBSCAN, Mean-Shift, Principal Component Analysis (PCA), Singular Value Decomposition (SVD) etc.

Reinforcement Learning

It's very similar to how humans learn: through experimentation. People don't want constant supervision to learn effectively like they do with supervised learning. By only giving us good or bad reinforcement alerts in response to our actions, you always check effectively. For example, a baby is now learning not to come into contact with a hot pan after experiencing pain. One of the essential components of mastering reinforcement is being able to move away from coaching with static datasets. Instead, the computer can analyze in a dynamic and noisy environment like the real world. Algorithms used for these methods are for self-driving cars, games, robots, resource management etc.

2.2 Deep Learning

Deep learning is a subset of machine learning. Deep learning modes can make their personal predictions completely unbiased by humans. However, previous machine study models require human intervention in many cases to achieve the highest quality result. In-depth understanding of the modes used for synthetic neural networks. The format of this community is stimulated using the organic neural community of the human brain. It analyses the data with a logical structure like how a human would conclude, as shown in below Figure 2.2. [13].

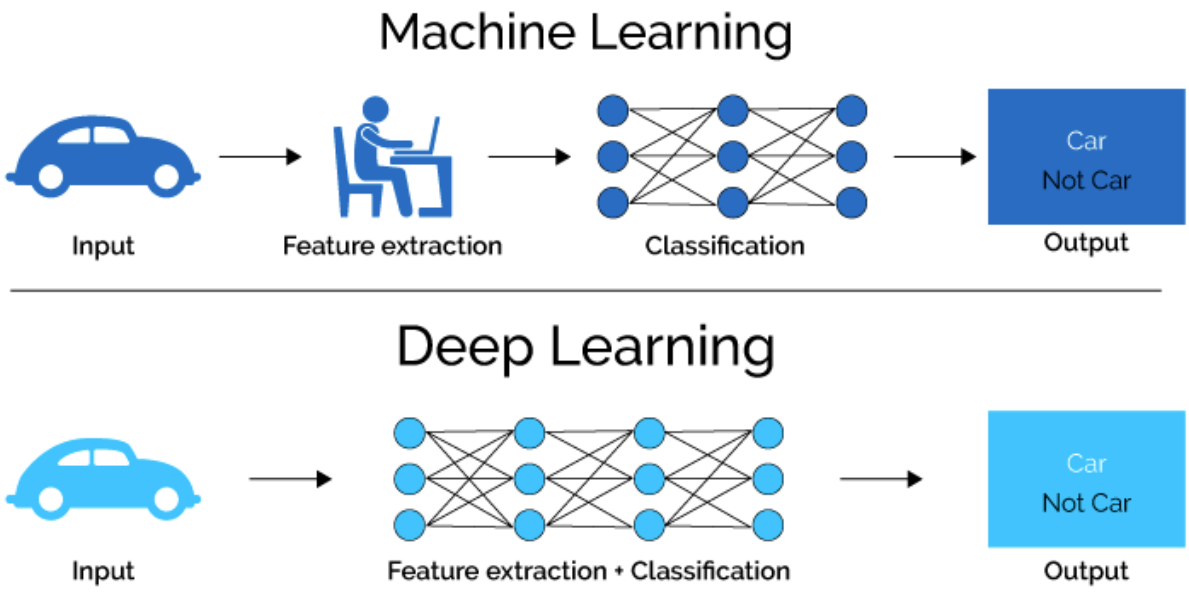


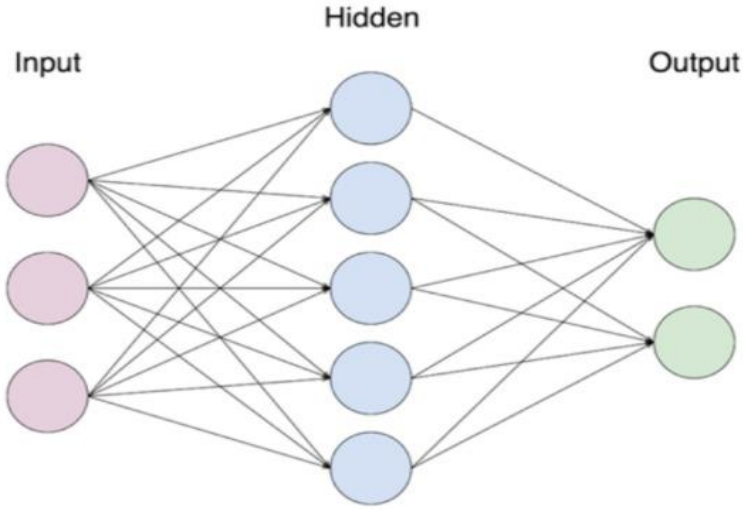
Figure 2. 2: Working Methodology difference b/w Machine Learning and Deep Learning

Deep knowledge acquisition is the next era of computing devices that enable multi-layered knowledge algorithms to extract higher-level aspects from raw inputs. For example, in applications of image processing, instead of just recognizing the pixels in the matrix, a deep knowledge of algorithms will understand the edges at one level, the nose at another level, and the object at all other levels. With the ability to recognize statistics on the degree of decline along the chain, a thorough study algorithm can improve its overall performance over time and achieve decisions at any time. Deep learning algorithms use complex multilayer neural networks, where the level of abstraction is steadily increased by nonlinear transformations of the input data. In a neural network, data sets are passed from one layer to another through connection channels, and they are recognized as weighted channels because there is a cost associated with each one. All neurons have a particular variety known as bias. This bias provided to the weighted sum of the inputs reaching the neuron is then used for the activation function. The end result of the trait determines whether the neuron is activated.

Each activated neuron transmits data to the subsequent layers. This continues until the 2nd last layer. The output layer in a synthetic neural community is the final layer that produces outputs for the program. Most deep study strategies use community neural architectures, so gaining deep insights into modes is often referred to as “DNN”.

The term “deep” usually refers to the number of hidden layers in the neural network, as shown in Figure 2.3. Conventional neural networks consist of only 2-3 hidden layers, while deep

networks can have up to 150 deep learning modes, trained through the use of massive units of labeled datasets and d neural community architectures that simultaneously examine recorded items alongside the desire for guidance function extraction.



2.2.1

Figure 2.3: Basic Structure of Neural Networks

Convolutional Neural Networks (CNN)

The CNN have broad applications in video and image recognition, natural language processing, speech recognition, and computer vision including Facial Expression Recognition. From several studies, it is found that CNN is robust to vehicle location changes and scale variations and behaves better than the multi-layer perceptron (MLP) in the case of previously unseen vehicle location. CNN have several advantages over DNN including the very similarity of the human visual processing system, which is well adapted to the structure of 2D and 3D image processing, and the effective learning and extraction of 2D features [15].

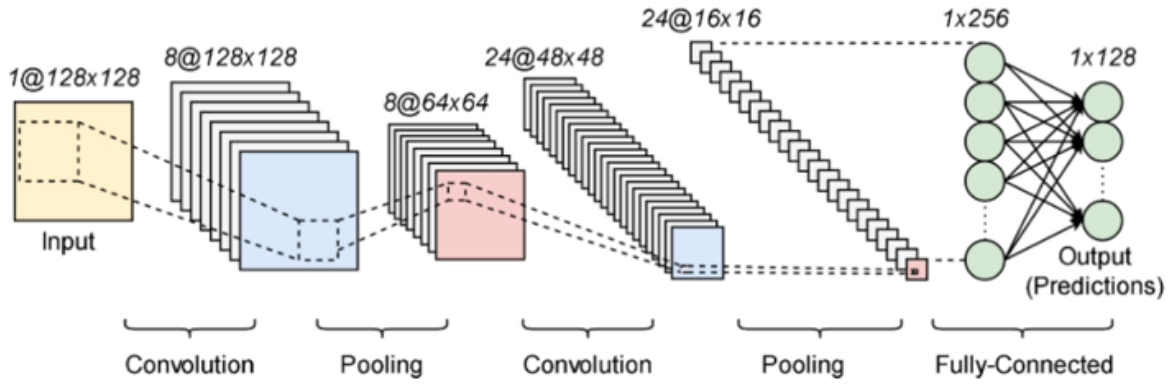


Figure 2. 3: Basic CNN Architecture

As shown in figure 2.4, [15] Convolutional layers, grouping layers, and closely bound layers are the three types of layers used by CNN. A series of training filters in the convolution layer convolve the entire input image and produce a set of specific sorts of activation feature maps. Neighborhood connection, which learns correlations between neighbouring pixels; Weight distribution in the identical map, which considerably decreases the number of parameters to learn; and displacement invariance to object space are the three key advantages of the convolution process. Following the convolution layer, the clustering layer is used to reduce the spatial measurement of feature maps and the network's computational cost. The two most common nonlinear subsampling strategies for translation invariance are average pooling and maximal pooling. The fully bound layer is typically overlaid at the end of the community to ensure that all neurons in the layer are fully bound to activations from the previous layer, as well as to allow for a 2D functional map to 1D functional map transformation for additional characteristic representation and classification.[15].

CNN has a number of advantages over traditional approaches, including the ability to extract features simultaneously, reduce data dimensionality, and classify in a single network structure. Furthermore, given to CNN's remarkable ability to decrease noise during picture capture, it simply requires minimum image processing. [15].

Convolutional Layer:

The convolutional layer is the core building block of a CNN. Convolution is the first layer that extracts elements from an input photo, convolving the pixels of an input photo with the small home-related location known as the neuron's respective field. In CNN terminology, this subject is additionally referred to as a “kernel” or “filter,” which is used as a known detector. The result of the scalar product is the so-called "feature map" that is obtained by sliding these filters over images. Each neuron shares a constant set of weights with the respective fields in an internally linked layer called the weight-sharing scheme. A mathematical operation that takes two inputs: image matrix and a filter or kernel, as shown in Figure 2.5. [16]

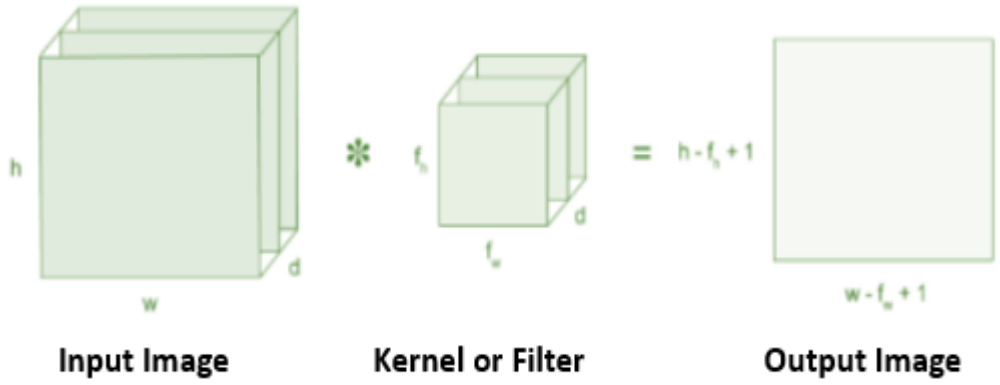


Figure 2. 4: Convolution operation with filter or kernel

This layer aims to detect facets of the image, e.g. B. vertical/horizontal edges, color gradients, etc. In order to examine certain elements, there will be a number of unique filters. Together they will shape the output of the neurons associated with the neighborhoods in the input. In other words, the output after this level are the elements extracted from the input of the areas in the images. To get the result, dot-product is performed between the Conv-Layer and the input layer, as shown in Figure 2.6.

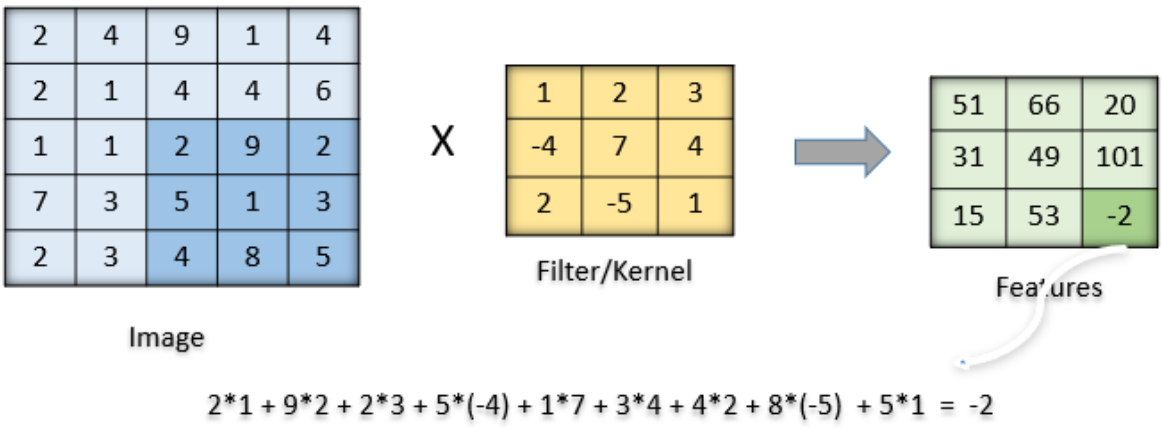


Figure 2. 5: Dot product of image and kernel or filter

Pooling Layer:

When the photos are too big, it's sometimes necessary to minimise the amount of trainable parameters. Between consecutive convolution layers, it is then desirable to incorporate pooling layers on a regular basis. Pooling is done solely for the aim of shrinking the image's spatial

size. Because pooling is done independently on each depth dimension, the image's depth remains unaffected. Subsampling or down-sampling is another term for spatial pooling, which minimizes the size of each map while retaining critical information. There are several types of spatial pooling: Max Pooling, Average Pooling, and Sum Pooling are three different types of pooling. The largest element from the corrected feature map is used in max pooling. Average pooling is the process of calculating the average value of a set of components. Sum pooling is the sum of all elements in a feature map. The example of maximum pooling, minimum pooling, and average pooling as shown below in Figure 2.7.

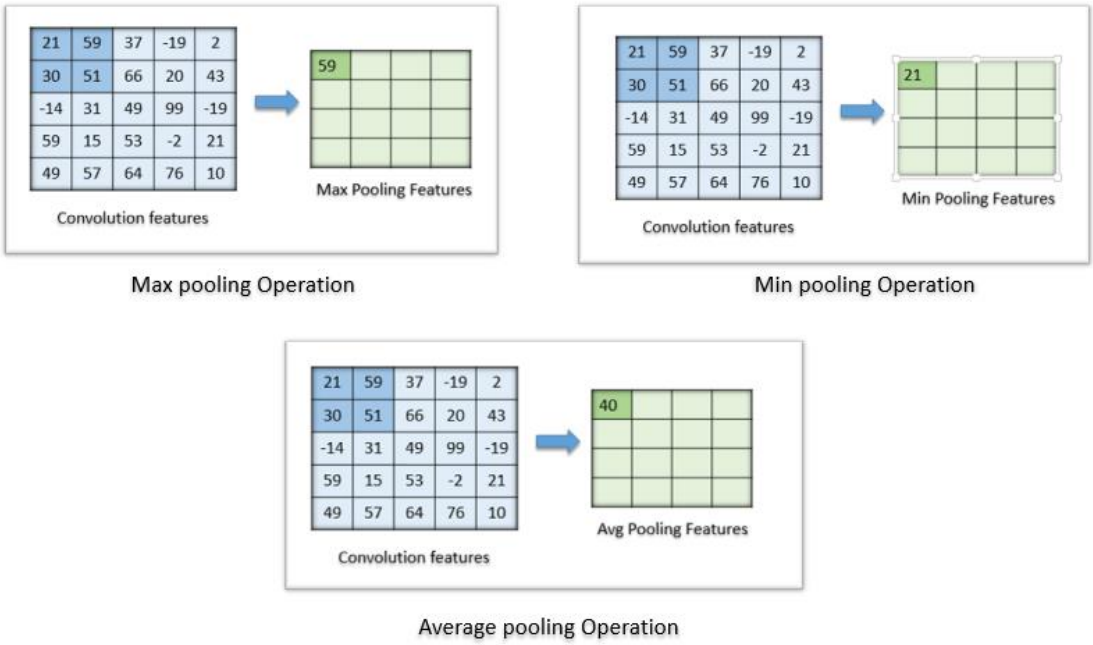


Figure 2. 6: Example of Max pooling, Min pooling and Average pooling

The pooling operation has two advantages: the first being helping prevent the model from over-fitting by providing as it makes an abstraction of the input volume. And the second, it reduces the input volume hence reducing the number of learnable parameters and saving computation resources.

Fully Connected Layer:

CNN relies on fully linked layers, which have been shown to be particularly effective at detecting and categorising images. Convolution and pooling are the first steps in the CNN process, which break down the image into features and analyse them separately. The output of this procedure is fed into a fully connected neural network structure, which is used to make the final classification decision. The output from the last pooling or convolution layer is sent into

the fully connected layer, which "flattens" them and transforms them into a single vector that may be used as an input for the following stage.

Following the fully connected layers, the last layer uses the activation function to obtain probability of the input being in a specific class, a process known as classification, which classifies the outputs into the appropriate label.

2.2.2 YOLO-v1

The YOLO algorithm, which is commonly used for object detection, was used in this study [5]. It frames object detection in images with a separated bounding box. It displays an image with a caption and highlights the object with the likelihood of right detection. The speed of the technique is crucial when it comes to real-time object identification and video processing. The detection and classification of objects in older object detection systems were done in two separate processes, which resulted in a longer processing time, which is why the YOLO was invented. The input is supplied forward to the network in the YOLO method, and the localization and detection are done in a single phase, resulting in a faster processing time. Figure. 2.8 [5] shows the basic architecture of YOLO.

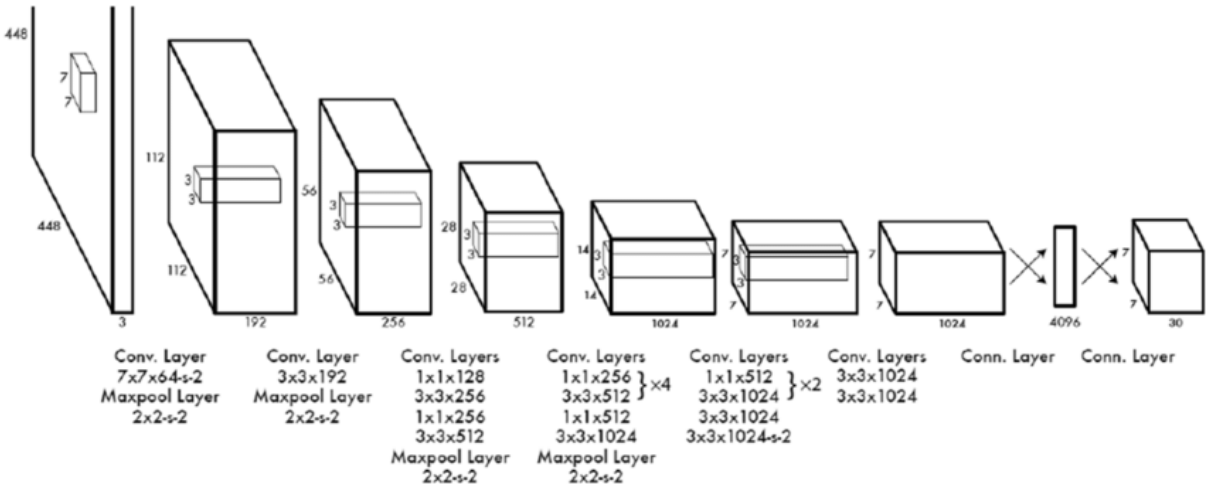


Figure 2. 7: YOLO Basic Architecture

Because of its quick inference, YOLO detectors are used in the majority of computer vision applications. Because we place a premium on real-time performance in order to satisfy the demands of the mobile robot. This timeline depicts YOLO's evolution over the last few years. Shown in Figure. 2.9 [18].

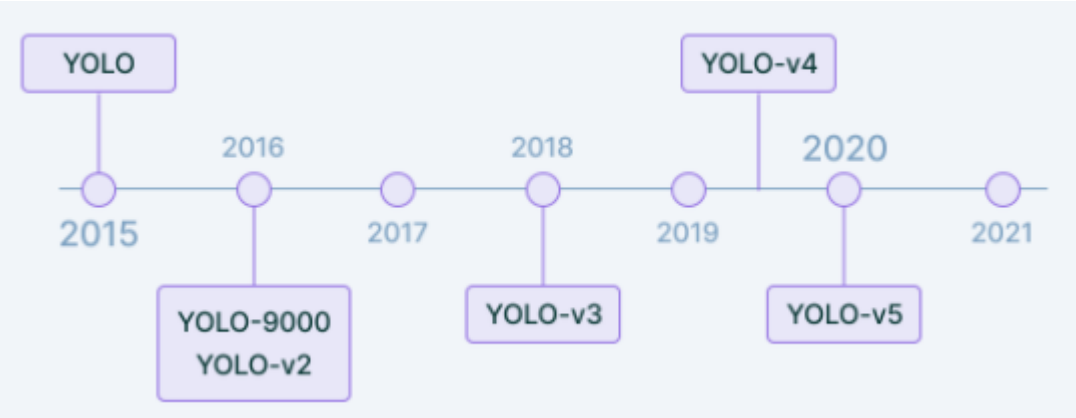


Figure 2. 8: YOLO Timeline

YOLO derivates have shown a fantastic compromise between accuracy and runtime speed from the early YOLOv2 to the most recent YOLOv4. The basic notion behind YOLO detectors is that they are one-stage detectors that handle detection as a straight regression problem. In fact, the YOLO detector is made up of three elements that are completely independent of one another.

Backbone: is the network in charge of forming characteristics. It learns relevant features that will be adjusted in the new task of detection after being trained on ImageNet classification.

Neck: integrates and blends the features created in the CNN backbone to collect both spatial and semantic information and deliver it to the detection stage.

Head: uses three different network scales to recognize multiple-size objects in an anchor-based manner.

In general, YOLO descendants follow the same rules. They do, in fact, access the entire image and divide it into a S x S grid. Rather than anticipating random boxes, they anticipate offsets to a collection of pre-selected boxes called anchors.

2.2.3 YOLO-9000\YOLO-v2

YOLO-9000 or YOLO-v2 [19] uses multi-scale training, which allows it to accept images of various sizes by deleting fully connected layers. Batch normalization, a high-resolution classifier, anchor boxes for detection, Darknet19, and multi scale training were all included to YOLO-v2 in comparison to the original YOLO. As stated in [19], YOLO v2 progresses are better, faster, and stronger in various categories. With Multi-Scale Learning, the network can now recognize and classify objects of diverse configurations and measurements. YOLO-v2 considerably enhanced the accuracy of identifying tiny objects compared to its preceding version.

2.2.4 YOLO-v3

Later, the YOLO-v3 [20] adopts a feature pyramid network (FPN) technique, allowing it to generate predictions on three different scales. YOLOv3-SPP, a modified version of YOLO-v3, recognizes objects of various scales using a slightly different technique by simply adding a spatial pyramid pooling (SPP) layer. To concatenate multi-scale local and global features, the SPP block is integrated shortly after the final features map. The main goal in developing the YOLO-v3 was to increase detection accuracy while keeping processing speed as high as possible. The independent logistic classifiers will be utilised for class identification on this version of the YOLO, and binary cross entropy loss will be employed for training. The feature extractor has been modified from Darknet 19, which has 19 convolutional layers, to Darknet 53, which has 53 layers, increasing the network's depth.

2.2.5 YOLO-v4

The next version of YOLO, YOLO-v4 [21], adds a few of new blocks, including the path aggregation network (PAN), which may be used to transport data from lower layers to higher layers. The takeaway here is the YOLO's modularity, in which small blocks can be assembled and integrated in many ways to analyze data together.

2.2.6 YOLO-v5

YOLO-v5 [22] is far most the latest model for object detection in the YOLO family, released in June 2020 by Ultralytics. Our research is based on the YOLO-v5, is explained in detailed in methodology.

CHAPTER 3

LITERATURE REVIEW

This chapter consists of the literature work from journal articles, conference papers, case studies and books. The main objective of this research is to collect, synthesize and organize the existing knowledge related to the usage of vehicle image or video-based database and the methods that are used for object detection. It is an area of computer vision that has been highly researched upon during the last few years. A lot of research work has been published regarding this area. Many ML and DL algorithms have been developed for object detection and recognition. In the first part of chapter, we are discussing conventional methods for object detection such as gaussian mixture model (GMM), SIFT algorithm, Histogram of oriented gradient (HOG), and Support vector machine (SVM) classifier. Later we discussed deep learning algorithms for object detection. Now a day's various computer vision tasks are being accomplished by deep learning. such as R-CNN, Fast R-CNN, Faster R-CNN, and CNNs, etc. In the last part of this chapter, we would be discussing the state of art object localization algorithms which are based on YOLO methods. The YOLO algorithm is one of such algorithms that achieve object classification and detection in real time. We aim to use the YOLO algorithm for the classification and detection of vehicles in images and videos. Different works agree on different methods to provide best detection results. In this section we will briefly discuss the work contribution by different researchers by utilizing different methods. Mainly three different methods are being used for object detection and classification, conventional, deep learning and YOLO based methods, this chapter will summarize all the valuable research in this domain.

3.1 Conventional Methods

This section briefly lists few recent conventional vehicle detection methods. Object detection brings together computer vision and deep learning methods, and it has attracted the interest of many academics due to its practical applications. Shibani Hamsa, et al., [23] uses the Gaussian Mixture Model for background reduction to explain a method for automatic vehicle detection from aerial pictures. Mohamed ElMikaty et al., [24] proposes a framework for detecting and localising autos in high-resolution airborne data utilising an ensemble of image descriptors that reflect the gradients, colours, and texture of cars. Çaglar Ari et al., [25] employing a Gaussian

mixture model with spectral and geographical constraints, describes a new approach for detecting heterogeneous compound structures such as diverse types of residential, agricultural, commercial, and industrial regions. [25] describes a GMM-based picture segmentation algorithm [25]. This work provides a method to recognize the front-vehicle in a video, focusing on a scene where the vehicle speed on the highway is high. To begin, the approach determines the vehicle's driving region using Canny edge detection and the Hough transform Hbaieb, A., et al., [26].

Second, we train an SVM classifier using the multi-feature acquired by combining the vehicle's histogram of oriented gradient (HOG) feature, color feature, and Harr feature, and then the classifier recognizes the cars in the driving area. The experimental results by Xiong, L., et al., [27] show that the SVM classifier trained using the multi-feature fusion method has a superior detection result than a single feature, and that the detection time in the driving region may be considerably reduced as compared to the detection of vehicles in the complete image. [27]. In [28], Yawen.T.,et al, vehicle detection was carried out using the SIFT technique, which extracts features in the form of eigen values. After that, SVM is used to classify the data. A mix of advanced pyramid pooling, sliding windows, and a non-max suppression technique is employed to improve the identification rate.

The extensive summary of the literature review is shown in the Table no. 3.1

Table 3. 1: Summary of Literature Review for conventional methods

Paper	**Year**	**Method(s)**	**Database**	**Accuracy**
Çaglar Ari et al., [25]	2014	GMM models		
Mohamed ElMikaty et al., [24]	2017	Framework for detection and localization of cars (SVM).	Vaihingen dataset OIRDS dataset	AP inter: 64.49%
Shibani Hamsa, et al., [23]	2018	Gaussian Mixture Model (GMM) and Support Vector Machine (SVM)	Aerial Images	Proposed system gives 87.5% hit rate, 94.7% accuracy and 100% precision values.
Yawen, T., et al [28]	2018	SIFT algorithm SVM Gaussian pyramid		Accuracy Rate: 93.4%.

Hbaieb, A., et al [26]	2019	Histogram of Oriented Gradients (HOG) descriptor with the linear Support Vector Machine (SVM) classifier	The car dataset built by Brad Philip and Paul Updike and taken on the freeways of southern California and consists of 526 images at 360x240 pixels constant resolution.	Pedestrian accuracy: 90% vehicle accuracy: 88%
Xiong, L., et al., [27]	2021	Support vector machine (SVM) classifier	in-car video in high-speed driving scenarios as a data set	Feature Dimension: 3328 Precision: 81.54% Recall: 83.24%

3.2 Deep Learning Based Methods

In this section the work done in the deep learning domain regarding object detection is briefly listed. In [29] L. Zhang et al., proposed a method to identify vehicles from satellite images using two CNN, SegNet and Mask R-CNN. To further improve the CNN results SVM is used with vehicle shape features. The dataset which was used are the satellite images with RGP channels in which central Manchester was covered the area of about 89km square.. The images were downloaded from google earth. Total of 1167, 934 are utilized for training purpose and 233 are for testing. Since images were less in number so the augmentation technique of flipping horizontally and vertically was also applied. After applying the three techniques the results were further improved from Mask RCNN to Mask RCNN and SVM. The testing results shows 1104 true positives and 1679 false positive with mask RCNN and SVM combined. It shows that false alarm appeared in less number in proposed method.

P. Saini et al., [30], developed a system for stolen vehicles identification without human involvement. For this deep learning algorithm single short detector (SSD) is integrate with K-Nearest Neighbors algorithm and convolutional neural network classifier. The experiment was performed in two stages, first the vehicle license number and color is detected from google API with help of vehicle features extracted from a video, then the previous detected features are being compared with the record. The dataset used was the real time camera image, 47605 images of 36 classes. The proposed method is tested on different vehicles, and it was able to detect the stolen vehicle color and license number.

Deep learning R-CNN Inception V2 model via transfer learning is used for efficient classification and detection of vehicles for self-driving cars by R. Kulkarni et at,., [31]. Different images of traffic signals in accordance with Indian Traffic Signals of 5 Classes

including 2 colors, red and yellow, and 3 directions, straight, left and right, were used in this experiment. The image frames were collected in daytime from the Puna and Maharashtra roads in India. The model was trained in 12 hours for the iterations of 120,000. The loss recorded was 0.01 which is far better than the previous range.

In [32] Q. Tan et al,. employs a deep learning algorithm to detect cars in high-resolution satellite remote sensing photos to detect vehicles more precisely, give accurate and effective data information to relevant departments, and improve traffic conditions. Firstly, The Alexnet network model is used to classify the images in the photographs, and then the vehicle target is determined. The Faster R-CNN model algorithm's detection performance is examined, and the algorithm is optimized by the model pruning and quantization approach, resulting in an average accuracy rate of 70.34%, and has strong reliability. In [33], proposed a CNN based model that include two steps: vehicle area detection and vehicle brand categorization. several common network models such as RCNN, Faster RCNN, AlexNet, Vggnet, GoogLenet, and Resnet were used in training and classification tests. The suggested method accurately identifies car models, brands, and other information with an average accuracy of 93.32%.

The extensive summary of the literature review is shown in the Table no. 3.2

Table 3. 2: Summary of Literature Review Deep Learning methods

Paper	Year	Method(s)	Database	Accuracy
M. Sheng, et al,. [33]	2018	RCNN, Faster RCNN, AlexNet, Vggnet, GoogLenet and Resnet	Vehicle Dataset	Average accuracy: 93.32%
R. Kulkarni et al,. [31]	2018	(R-CNN) Inception V2 model in TensorFlow for transfer learning	Different images of traffic signals in accordance with Indian Traffic Signals of 5 Classes	Loss: 0.01
P. Saini et al,. [30]	2019	SSD algorithm is coupled with K-Nearest Neighbors algorithm + CNN	47605 images of 36 classes	Observes any anomaly in two sets or matched with stolen vehicle complaint record

Q. Tan, et al,. [32]	2020	Alexnet network model, Faster R-CNN	Remote sensing image classification	Average Accuracy rate: 70.34%		
L. Zhang et al,. [29]	2021	SegNet and Mask R-CNN CNN results are further improved by SVM	The real satellite images covering central Manchester area	DET.	TP	FP
				R-CNN	1243	1788
				SegNet+SVM	1116	152
				R-CNN+SVM	1104	1679

3.3 The YOLO Based Methods

After working on the deep learning-based methods, researchers started finding out the YOLO based model for the object detection. This section briefly describes some of the work of different research using the YOLO and modified YOLO in different scenarios of object detection.

A real time object detection algorithm was proposed by Lu, Shengyu, et al. [35], it was based on the YOLO. In the proposed technique the images are preprocessed to remove the background, then objects are detected using fast YOLO. The YOLO is improved on the bases of GoogLeNet, a convolution operation is replaced, which results in reduction of the parameters. That is why it is called fast YOLO, as it takes less time and has less parameters.

Lin, J. P., et al. [37] proposed a method to count on the traffic flow, with the use of YOLO. There are mainly three steps of the architecture, detection, buffer, and counter. First the bounding box are created using detection, vehicle coordinates are stored with help of buffers, then there is a counter which counts the vehicles.

An optimized YOLO is proposed by Tao, J., [41], they fine-tuned the YOLO by replacing the fully connected layers of YOLO with average pool layer. The new network outperforms the region-based approaches for example R-CNN, it achieves the results 1.18 times faster than traditional YOLO. To further improve the network, it is also combined with R-FCN. The testing results on night gives improved results after doing some preprocessing for night images.

Putra, M. H., [43] proposed modifies YOLO in improvement in intelligent cars and advance driver assistance system. They used 7 convolution layers in YOLO, to reduce the grid size for small size car and person detection.

Sang, J., et al., [42] proposed a method which resolves the issues of low detection accuracy, slow speed, and vehicle type detection. They used YOLO-v2 vehicle based, which performing the k means clustering in training dataset, on the bounding boxes of the vehicles, selecting different sizes as anchors boxes. The different sizes are then normalized to calculate exact loss in term of length and width. For achieving good results in term of feature extraction, fusion strategy with multi-layer features is used. They results were very promising when tested on BIT vehicle validation dataset, with 94.78% mean average precision. It is also tested on other datasets, as Comp Cars, to prove that the proposed method is applicable to any vehicle detection systems.

Miao, Yan, et al., [34] approached a technique that detects the vehicle at nighttime. YOLO-v3 is pretrained on enhanced dataset and detection algorithm such as, faster R-CNN and SSD is being used to detect vehicles. The proposed method achieves the precision of 93.66%, which in comparison of faster R-CNN and SDD was 6.14 and 3.21% higher than respectively.

Viewing the work of other researchers, a video surveillance-based vehicle tracking system was proposed by Al-qaness, M. A., et al., [39], which combines three methods, tracking based on images, neural network and YOLO-v3. The proposed network is then tested real time on road traffic sequence. The dataset was collected through surveillance cameras, like Open Image, Pascal VOC, COCO dataset, the dataset is refined by using YOLO to identify the object such as car, truck, public transport etc., for training purpose. The dataset is then passed through a detection algorithm, which helps in improvement of traffic maintenance.

Huang, B., [40] used a tiny YOLO-v3 algorithm for target detection. The method detects the vehicle and identify the license plate number of the vehicle. They combined the YOLO-v3 with Birch algorithm to improve the acquisition results. For the improvement in real time performance, multi scale prediction of three scale detection is combined with two scale detection. The results proved that the improved YOLO-v3 tiny structure performs very well on the base of mean-average-precision, intersection over union and speed giving the results of 5.99%, 17.52% and 48.4%, respectively.

Cepni, S., et al., [44], studied different methods for object detection such as traditional CNN methods and the deep learning models such as YOLO and the family of YOLO-v3, v3-spp, v3 tiny etc. The comparative study is the tested-on COCO dataset which gives promising results on YOLO v3-spp with average of 84.88% and precision value of 72.02%.

With the continuous work and improvement on YOLO and object detection Peng, H. et al. [36] used YOLO v4 for more accurate and improved results. Apollo Scape automatic driving data

set is used in this method, which is first manually labeled and formed a database, which then passed from a YOLO v4 to extract the features. The results are compared with the available method as YOLOv4 tiny which has mean average precision of 3.2%, Faster-RCNN with mAP of 3.1% and YOLOv3 with mAP of 6.9%, but the proposed method gives mAP of 92.1%.

Wang, C., et al.,[38] proposed a high-level method to detect the smoky vehicles, they collected the dataset on the spot, captured mean time photos to create the database of semi-trailers, cars, vans, and different vehicles. The images are then augmented to achieve the improved and generalized results. The augmentation technique used in this proposed method is Cutout, which results in difficult training but its extents the mode ability. The model used is MobileNetv3-small which replaces the YOLO-v5s feature extraction network. At first, the dataset is trained without augmentation and then after augmentation the detection accuracy is increased by 8.5%.

Huang R. rt al., [45], worked on YOLO-LITE, it run the real time object detection model, on the portable devices such as laptop, mobile phone which do not have a working graphical processing unit. Researcher first trained the model on two different datasets, PASCAL VOC and COCO dataset giving the results of 33.81% and 12.26% mAP on each, respectively. The experiment shows that YOLO-LITE gives very promising results on a non-GPU computer when run on 21 FPS, since it is a very small system. The results are 3.8 time faster that of the previous networks such as SSD mobilenetv1. YOLOV2, YOLO-LITE was designed to create a smaller, faster, and more efficient model increasing the accessibility of real-time object detection to a variety of devices. Also, it has been shown that there could be a question on the use of batch normalization in smaller shallow networks.

The extensive summary of the literature review is shown in the Table no. 3.3

Table 3. 3: Literature Review YOLO Based Methods

Reference	Year	Method(s)	Database	Accuracy
Tao, J., [41],	2017	OYOLO (Optimized YOLO), and OYOLO and R-FCN	KITTI data set + road captured images	OYOLO 80.1% OYOLO + RFCN 83.4%
Sang, J., et al.,[42]	2018	YOLO v2	Comp Cars test dataset,	mAP is 94.78%
Lin, J. P., et al. [37]	2018	YOLO, Buffer and counter	Side entrance, and backdoor exit for to National Central University	Inbound accuracy is 100% for morning, afternoon and evening.

				Outbound accuracy is 100% except of night which is 80%
Rachel Huang [45]	2018	YOLO-LITE	PASCAL VOC dataset	mAP of 33.81%
			the COCO dataset	mAP of 12.26%
Putra, M. H., [43]	2019	Modified YOLO which uses 7 convolutional layers.	INRIA dataset	YOLO_7 x 7 mAP 37.9
Lu, Shengyu, et al. [35]	2019	Improved YOLO network by replacing the original convolution by a small convolution operation, based on GoogleNet. Fast YOLO	vehicle monitoring videos from smart cities from the Xiamen municipal transportation bureau. Videos consist of several 416x416 color image	Fast YOLO algorithm can recognize 45 frames per second,
Miao, Yan, et al. [34]	2020	YOLO v3, Fine tuning on YOLO v3	880 nighttime Images were collected using Logitech C920 camera. 1760 total images after preprocessing	Average Precision: 93.66%
Cepni, S., et al.[44],	2020	YOLO-v3, YOLO-v3-spp and YOLO-v3-tiny models	1. Video obtained from UAV (van Es, 2017) with 1280x720 resolution 2. Video obtained with terrestrial resolution of 1080x1920 were used	YOLOv3-spp average IoU: 84,88% precision value: 72,02%.
Peng, H. et al.[36]	2021	YOLOv4 deep learning model	Apollo Scape automatic driving data set	mean average precision (mAP): 92.1%
Wang, C., et al.,[38]	2021	YOLO v5 improved by Mobilenet	self-built dataset, expanded to 6102 images by preprocessing	12.5 FPS
M. A., et al[39],	2021	YOLO v3	Pascal VOC, Open Image, and COCO datasets	The maximum vehicle counted on daytime 98%.
Huang, B.,[40]	2021	Improved YOLOv3-tiny algorithm	Natural scene	mean-average-precision, intersection over union and speed is improved by by 5.99%, 17.52% and 48.4%,

It has been cleared that most of the research worked on wide based dataset, along with YOLO and different families of YOLO. Other than these research's, many work has been done on YOLO and the modified YOLO, some may use the combination of YOLO with other object detection algorithms for real time, nighttime, or video-based object detection. We have come across some gaps, discussed in the section below which led us to this research.

3.4 Research Gaps

Aforedescribed are few of the nice efforts that aim to detect vehicles under various conditions. Most of these works report experiments on PASCAL, VOC, and COCO dataset, which are not from Pakistan. One of our aims is to locate objects, such as cars, motorcycles, and pedestrians on Pakistani roads. However, our study reveals that conventional methods have less accuracy as compared to deep learning algorithms. Therefore, the methods, such as YOLO and its families have fewer complex architectures and do not have many layers involved. Therefore, in our work we utilize the architecture of the YOLO-v5 as described in the next chapter

CHAPTER 4

DATASETS AND METHODOLOGY

In this chapter first the dataset gathered for the vehicle detection is discussed followed by the proposed methodology.

4.1 Datasets

Since there are some datasets available for vehicle detection previously, a lot of work has been done on them by different researchers. But we are not using these publicly available datasets directly. We gathered our own dataset to carry out training and experiments to employ transfer learning. So, we have collected the dataset through different views and in different road and traffic conditions of local environment. Some of the publicly available datasets have been discussed below followed by the dataset gathered and used in this research.

4.1.1 COCO

COCO 2017 datasets [46] have different subtypes of datasets, such as object detection task, dense Pose task, panoptic segmentation task. Each of the task has different classes, COCO objection detection task is the relevant to our research. In COCO object detection task, it has been used for the detection of car, person, or anything like animal or human. there are more than 200,000 images for train test and validation set along with 50 categories, shown in Figure 4.1. The annotation is COCO is stored as JSON file containing info, licenses, category, images, and annotations.

Figure 4. 1: COCO dataset sample images

4.1.2 PASCAL VOC

There are 20 object categories in PASCAL VOC abbreviated as PASCAL Visual Object Classes, such as vehicle, animals, households, and others. Also, the images come along with three different types of annotation, bounding box, object class and pixel level segmentation annotations. The PASCAL VOC dataset is split into three subsets: 1,464 images for training, 1,449 images for validation and a private testing set. PASCAL VOC stores annotations as XML file, each images have different file. Some sample images from PASCAL dataset is shown below in Figure 4.2. [47]

Figure 4. 2: PASCAL VOC dataset images

4.1.3 Database

All the datasets mentioned above are from different categories and has a lot of classes, which we have not used because we want to employ transfer learning for model fine tuning. So, we acquired the dataset from real time images and videos. Our dataset contains 2400 images assigned to 2 categories of traffic scenes with 1800 images, belonging to category 1 and 600 images, belonging to category 2. Each category has three classes, which are car, motorcycle, and person. In addition, we have also gathered a video dataset, which contains same aforesaid three classes. The database is shown below in Figure 4.3.

Figure 4. 3: Sample Images of Traffic dataset

4.2 Proposed Methodology

Our proposed methodology consists of three main modules. The first module is a dataset acquisition module in which we require the data relevant to our problem. In the second module we basically perform training. In the training module we perform transfer learning. Transfer learning is a technique where pre-trained model weights are used as a starting point and fine tune these weights on acquired data. From this module a fine tunned weight is obtained which is then used to test the images in the next module. This module takes the test images and detect the objects with in it and classify the detected objects as one of the 3 categories. These three modules are described in detail below. The flow diagram of our proposed technique is shown below in Figure 4.4 which clearly demonstrate all the three blocks of the model, as well how the model is trained and then used for classifying and localize the objects. The sections below discuss modules of the proposed technique in detail.

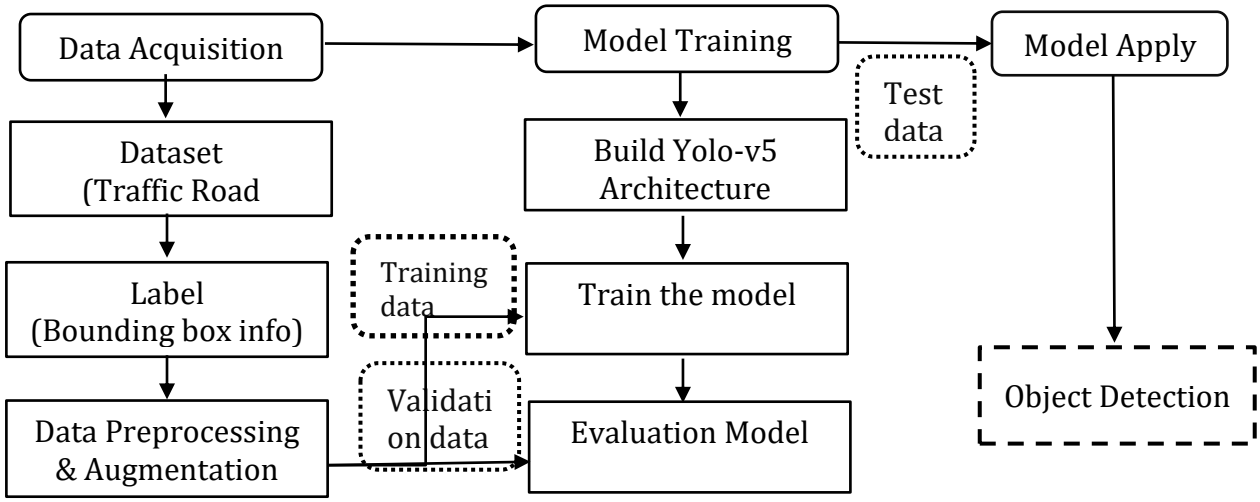


Figure 4. 4: Flow of the proposed method

4.2.1 Image Data Acquisition

To gather images, we face different challenges on the roads. For instance, in Pakistan, we come across the multi-class objects on the roads, such as massive traffic jams and overlapped vehicles. Therefore, we have gathered the dataset on two different situations. One is the image dataset at high density traffic scenes, and the other is the images from the low density traffic scene in which there is only one class in each image, with no overlapping. For better training, the images of low and high density dataset have been placed separately. To manage data systematically, we name it *as* Low density traffic scenes and High density traffic scenes as briefly described below.

4.2.1.1 Low density traffic scenes:

This dataset has been collected from the daily real-time traffic scenes, such as parking-lots, roads with less crowd, streets, and the places where there is less crowd of people. The purpose of gathering this dataset is to train the model on each class separately. We collect a total of 600 images from all three classes of cars, motorcycles, and pedestrians. The sample images from each class have been shown in the Figure 4.5.

Figure 4. 5: Sample Images of Low density traffic scenes

4.2.1.2 High density traffic scenes:

This dataset has been gathered from the crowded places, such as the parking-lots, shopping malls, highways, and the scenes near traffic signs. We collect a total of 1800 images for all three classes. The sample images have been shown below in Figure 4.6. We also gathered dataset considering challenging factors such as, low and high illuminations, occlusions, and Group of objects irrespective of size, shape, or color as shown in Figure 4.7 and 4.8. The statistics of both the dataset along with each class annotations are described in Table 4.1.

Figure 4. 6: Sample Images of high density traffic scenes

Figure 4. 7: Traffic scenes with occlusions

Figure 4. 8: Traffic scenes with low illumination

Table 4. 1: Dataset Images

Dataset		Crowded dataset	Less crowded dataset
Source images		1800	600
Annotations		15618	903
Class balance	**Car**	8457	655
	Motorcycle	4136	136
	Person	3025	112

4.2.1.3 Video dataset

Along with the images, we also gathered a set of video dataset from the location of crossway bridges. Few sample images of video dataset are shown in Figure 4.9.

Figure 4. 9: Video Frames (Video dataset)

4.2.2 Data Annotation

Data annotation is mainly the process of labelling the classes of the datasets for object detection through different tools for the purpose of supervised machine learning. Data annotation is an important step for the good training of the CNN model along with to get promising results, it should be done before feeding the dataset to system. Data annotation overall improves the accuracy of the output. The whole purpose of the data annotation process to label the classes in the dataset, and to assign them the respective class. Data annotation has different types such as semantic annotation in which different concepts are being labelled, such as to label a name from a text. Another type of data annotation is text classification, it extracts general tags from the text which is unstructured. The type of data annotation used in this research is Image and video annotation, it draws the bounding box to focus on different objects. The objects are then

labeled according to the respective class. There are different tools available online for data annotations.

For the image dataset, the labels have same numbers as of images. In this research we have used “Label Image Tool” for labelling and annotating the image dataset. The label image tool is easy to use and consumes less time. The image dataset of both categories has been uploaded to the Label Image tool, which reads the images and assign a bounding box for each object present in the image. For high density traffic scenes there are many bounding boxes on a single image. But for low density or a single traffic scene there is only one bounding box. These bounding boxes then define the label as the respective class. Each of the image is labeled according to the bounding boxes in labelImg tool as shown in Figure 4.10. The annotated images along with their annotated .txt file in shown in Figure 4.11. The first value of .txt file shows the class labels the other four values are coordinates point and bounding box width and height. The overall dataset is then divided into three classes, car, person, and motorcycle.

For the video dataset, the annotation is somehow lengthy, to do it quickly we used Dark label tool, it consumes less time in comparison of label image tool for the annotation of video dataset. The Dark image tool divides the uploaded video dataset into frames, such as frames of 10 seconds into 360 frames. These frames are then interpolated, in interpolation the first frames draw a bounding box around an object, and the last frame draws the bounding box around same object so all the objects in between the 10 seconds has been annotated and labelled according to the respective class. The interpolated frames and then saved in a .tex file, each frame with same method and in results the labelled video dataset is ready as shown in Figure 4.12.

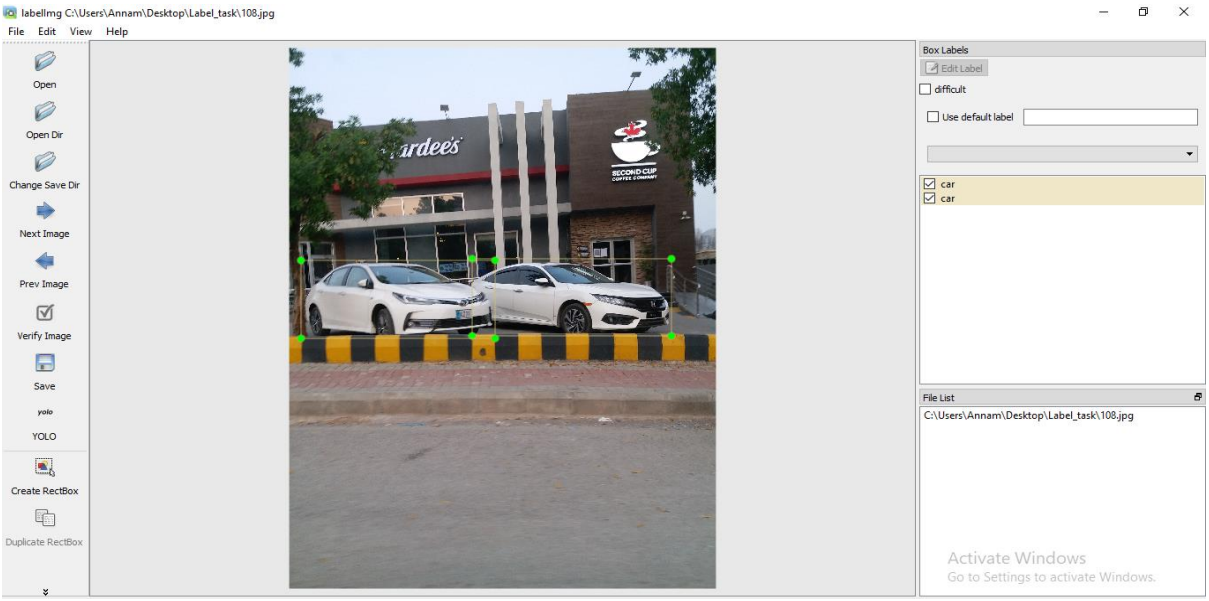
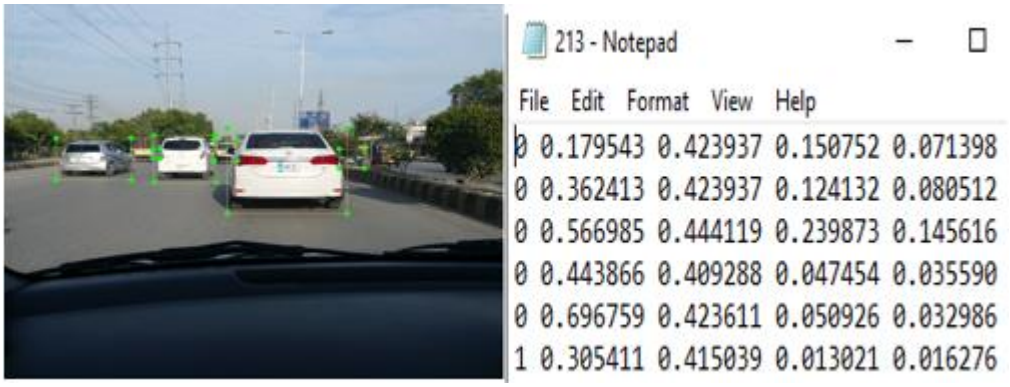
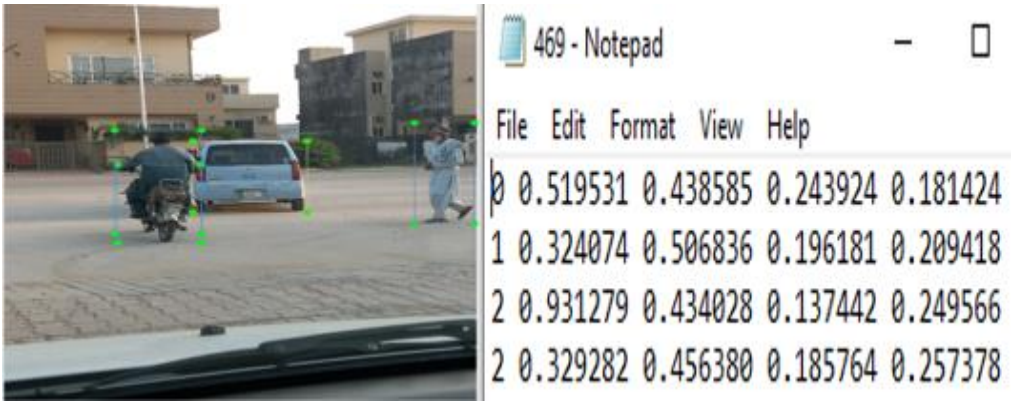


Figure 4. 10: Image Annotation Labelling tool



(a)



(b)

Figure 4. 11: Annotated Images: (a). Annotation of car and motorcycle (b). Annotation of car, motorcycle, and person

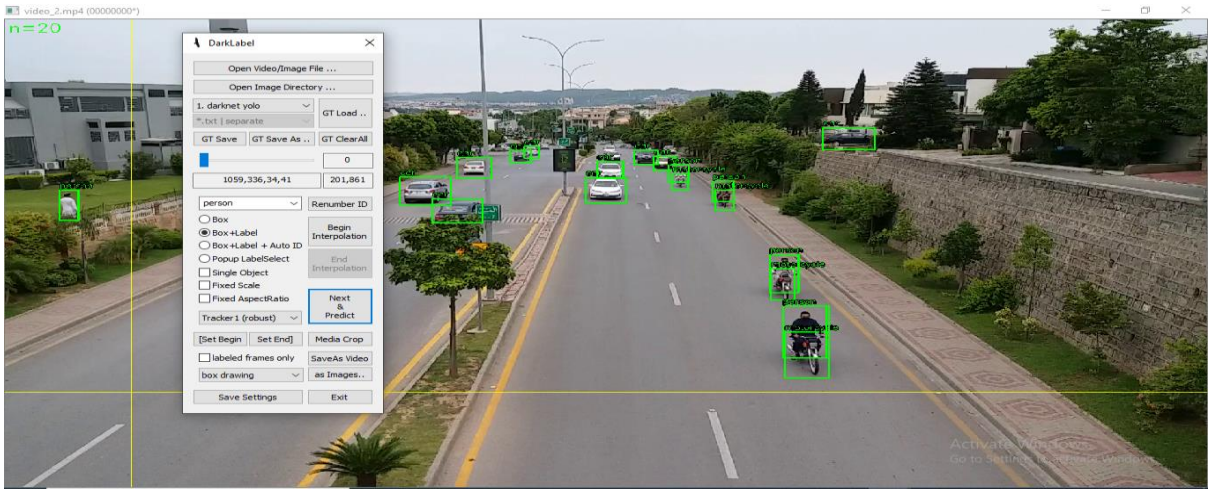


Figure 4. 12: Video Annotation in Dark label tool

4.2.3 Data preprocessing

To improve the data quality for better results, data preprocessing is the building block of deep learning. The real-world datasets might be noisy, or they could be inconsistent, some things may be missing, or could be uneven classed. We have preprocessed our dataset by two steps, one is to make the same size of each image of both categories and video dataset. We make the dataset of 416×416 pixels resolution of each image and video. Then the dataset is split into train, test, and validation set.

4.2.4 Data Augmentation

Training a model on small number of images could result in overfitting [48]. It results in poor generalization, although the training results are good, but the testing accuracy keeps dropping and the model classify the samples into one class, in short, the training accuracy is high, but the validation accuracy drops down. To overcome this issue data augmentation has been considered a good option, to enhance the dataset using different techniques for better results. Data augmentation is a technique which modifies the data by different techniques and increase the samples of the dataset. There are different augmentation techniques, some of them are mentioned below:

• Rescaling: Image could be rescaled on new dimensions
• Cropping: Image is cropped according to specific dimension
• Shifting: The image pixels could be shifted right or left
• Flipping: Image is either flipped Vertically or horizontally
• brightness changing: the contrast or brightness could be decreased or enhanced.
• Zooming: Image could be zoomed in or out, using scaling factor
• Saturation: The color range of the images could be altered

The traffic dataset has been then augmented using cropping, saturation and brightness changing technique as shown in Table 4.2.

Cropping: The image dataset of both categories and the video dataset has been cropped between 0% minimum zoom and 30% maximum zoom.

Saturation: The color range of image dataset of both categories and the video dataset has been changed. Images are saturated between -25% and +25%.

Brightness changing: The brightness of the images of image dataset of both categories and the video dataset has been changed. Darken and brighten the image between -25% and +25%.

After applying the data augmentation technique to the image and video dataset, it is ready to use in a model for object detection.

Table 4.2: Augmentation of Dataset

Data Augmentation	
Crop	0% Minimum zoom, 30% Maximum zoom
Saturation	Between -25% to +25%
Brightness	Between -25% to +25%

4.2.5 YOLOv5

YOLOv5 is far most the latest model for object detection in the YOLO family, released in June 2020 by Ultralytics. It has further four types, YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x. These types differ in size and inference time. The size ranges between 14MB to 168MB.
The Figure 4.13 [49] shows the model comparison .

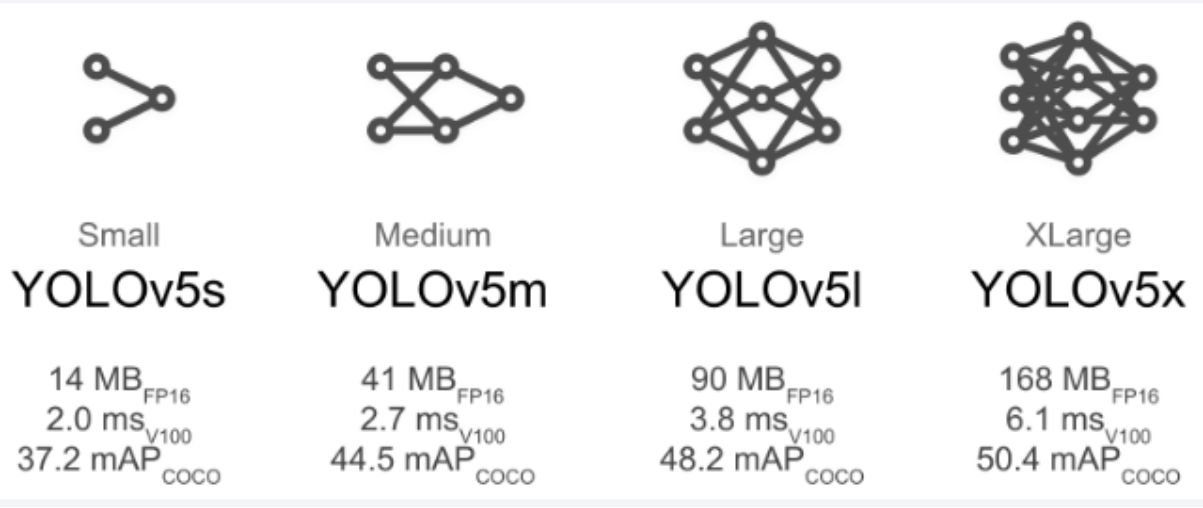


Figure 4. 13: YOLO Family

YOLOv5 has out passed other traditional object detect algorithms. We have already discussed YOLO v4 and YOLOv3, now we will have a brief overview on YOLOv5. It was released in 2020 by Glenn Jocher. YOLOv5 is easy to use as in comparison of other YOLO algorithms. There are three main architectural blocks, discussed below:

Backbone: In YOLO v5 the CSP-Cross Stage Partial Networks are used as a backbone to extract important features from the given input image.

In large backbones, CSPDarknet53 is used as backbone to solves the repetitive gradient information while the integrates gradient changed to feature map which helps in reducing the speed of inference, increases in accuracy, and the model size is reduces by decreasing the parameters.

Spatial pyramid pooling (SPP) is a pooling layer that is used to remove the fixed size constraint of the network.

Neck: The feature pyramid is built with PANet for feature aggregation. The features are then passed to head.

(PANet) is used as neck in order to boost the information flow. A new feature pyramid network (FPN) has been adopted by PANet that includes several bottom ups and top-down layers. Through with the propagation of low-level features in the model has been approved. PANet improves the localization in lower layers, which enhances the localization accuracy of the object.

To upsampling the previous layer fusion, upsample is used in the nearest node. Concat is a slicing layer and is used to slice the previous layer.

Head: In this block the predictions are generated with help of anchor boxes, which results in detection of object.

The Figure 4.14 below shows the YOLOv5 architecture [49].

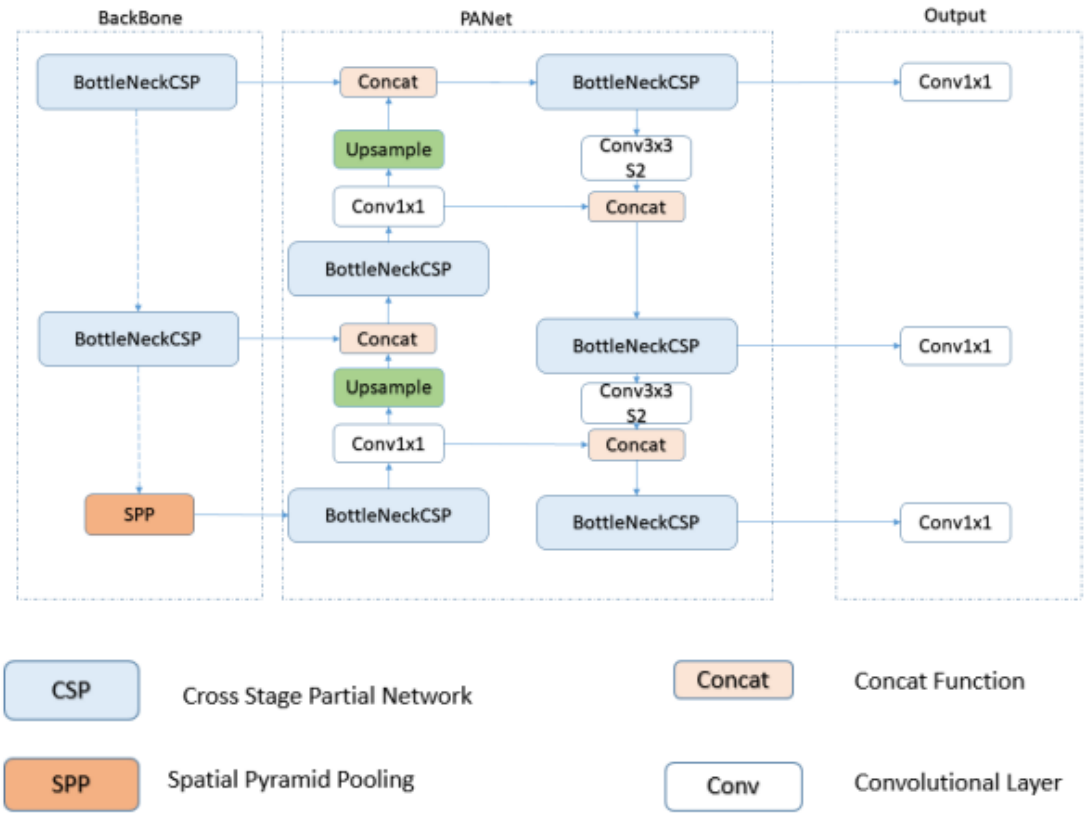


Figure 4. 14: YOLOV5 Architecture

Activation Function:

These are the architectural blocks, YOLOv5 also use other methods in training by using different activation functions, choice of activation functions is an important step in any deep neural network. Some of the activation functions are LeakyReLu, ReLu, swish, sigmoid etc.

In case of YOLOv5 the author decided to use two activation function mentioned below:

- Leaky ReLu: This activation function is used in hidden layers.
- Sigmoid activation: This activation is used in final detection layer.

Optimization Function:

Optimizers mentioned below used in YOLOv5

- ADAM
- SGD

By default, SGD is used as optimizer for training. However, it could also be changed to ADAM in command line.

Loss Function/ Cost Function:

Binary cross entropy is used as loss function in YOLOv5.

Comparison with YOLOv4:

YOLOv5 is easy to use in comparison of YOLOv4, in term of installation only a .txt file needs to be executed while in YOLOv4 a whole setup needs to run. In terms of storage size, it stores the weights in PyTorch format minimum of 27mb file, in comparison of YOLOv4 which stores weights in at least 250mbs size.

- Training steps for YOLOv5 includes:
- Setting up environment
- Setting up data and directories
- Setting up configuration of YAML files.
- Training the model for object detection.
- Testing with custom YOLOv5.

The data should be formatted accordingly, training and validation images in different folders. The labels should be in a .txt file. As for bounding boxes, it should be listed as one bounding box per line.

4.2.6 Transfer learning

Most of the machine learning tasks now a days use transfer learning, a method where the trained weights could be reused for another training as shown in Figure 4.15. It is very common approach in computer vision and machine learning, it saves time and make the computation easy. It is an optimized method. In transfer learning a pre trained is model is used on another but related problem. Such as training a deep neural network on a large dataset, we get the trained weights on a large dataset [50]. It takes a lot of time. and these weights can be used for a new dataset which has same features as the trained dataset. It's the choice of the researcher, whether he trains the final layer because of small dataset or he needs to train the whole model for a large dataset.

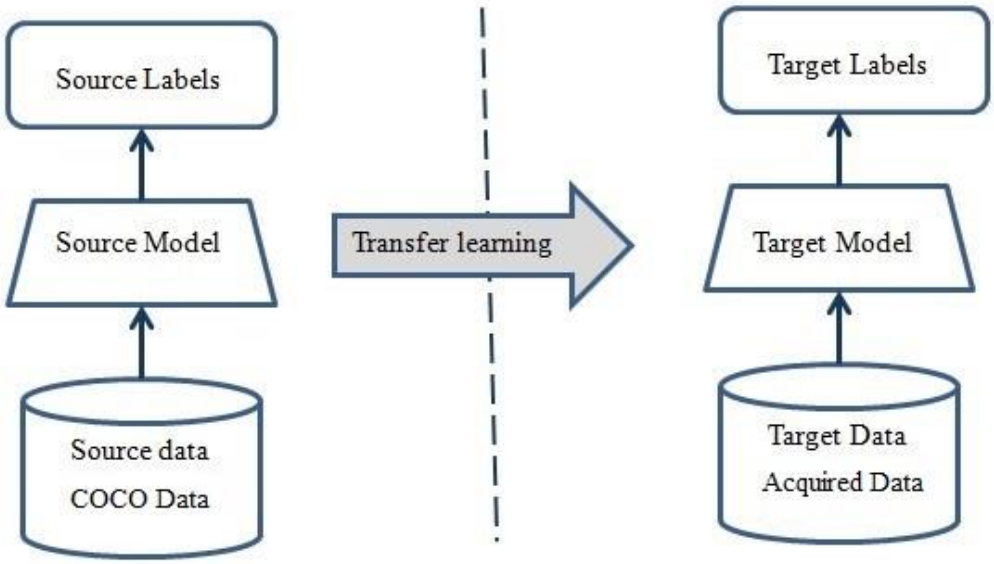


Figure 4. 15: Transfer Learning

In computer vision, transfer learning is defined as the use of pre-trained models. Pre-trained models are those that have been trained on huge standard datasets (such as COCO) and have solved a problem like the one we're trying to solve. Due to the high expense of training such models, it is common practice to import and employ models from pre-trained (YOLOv5) to our acquired dataset.

Transfer learning process

The complete transfer learning process from practical point of view is summarized as follows:

1. First from a wide range of available models choose a pre trained model. Choose model that is most related to the problem that needs to be solved. Like if you are using Keras access InceptionV3 [33], VGG and YOLOv5.
2. After that remove the classifier that was originally used and adds classifier that fits your task.
3. In the end needs to fine tune the model, for that there are four different options:

a. If the dataset set is different from the pre-trained model's dataset and are large. Train the model from the scratch because complete model training required large dataset.

b. If the dataset is large and like the dataset on which model was trained previously. Then it is sufficient to train the top layers of convolutional base and the classifier, it will save time and huge effort of training.

c. If the dataset is small and different from the dataset on which model was pre trained.

Then there is need to find a balance between layers to freeze and to train. Model will get over fit if you will go deep and it will not learn anything if you stay shallow. In this case strategy mentioned in point 2 will help, that is train the top layers of convolutional base and the classifier.

d. If the dataset is small and a problem that is like your problem was solved by the pre trained model. Then only last fully connected layer needs to be removed. As a fixed feature extractor, run the pre-trained model. After that, to train the new classifier use the resulting features.

Transfer learning is very beneficial because it speeds up the process of training model by reutilizing the modules or pieces of models that are already developed as shown in Figure 4.16. It also accelerates the results.

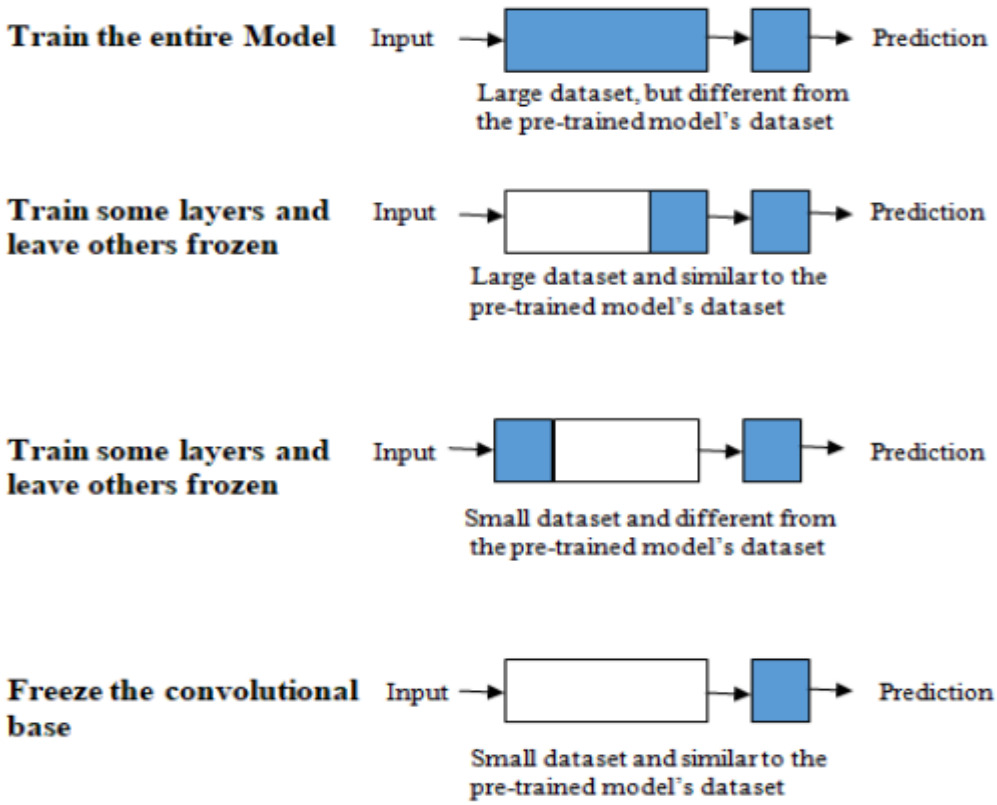


Figure 4. 16: Fine tuning

4.2.7 Training on Proposed dataset:

After the data acquisition, data annotations and data pre-processing, the training and validation dataset has been passed to YOLOv5s, for object detection. YOLOv5 has been implemented through Google Colab Pro. In Google Colab pro, the dependencies and environment of YOLOv5 has been installed for object detection. After the installation of YOLOv5 environment, the model configuration and architecture has been defined. In our training we have used the YOLOv5s. It could be defined using one line of code "custom_YOLOv5s.yaml";.

For training, we need to select the batch size, epochs, and size of the images. The path of training, testing, and validation dataset has been given. We have not trained our custom images and video dataset from scratch. If we must train our model from scratch, we must initialize it with some random weights. However, we have used pre-trained COCO weights for our model training as it saves a lot of time and makes computations easy. Using pre-trained YOLO-v5 model, we get the best weights after transfer learning. Moreover, we have used default layers and anchors because we are using the initial weights of COCO dataset. Furthermore, we have also used COCO as a benchmark to train our custom dataset. Meantime, we have also varied the batch sizes as, 5, 10, and 20. We have also changed the epochs to 100, 300, and 500. The value of confidence could also alter. After training and by changing parameters, we obtain the best results, as soon will be explained in chapter 5. After the training, we used the best weights to detect objects on the dataset. Finally, we get the values of predicted labels and the test images with the bounding boxes with confidence values.

CHAPTER 5

EXPERIMENTAL RESULTS

In this chapter, a brief overview of experimentation details and results of this thesis are discussed. In this chapter we evaluate the experiment done and its result on high density and low-density images dataset and traffic video dataset. A brief overview both images dataset and video dataset are given and then hardware requirements and evaluation parameters are discussed. The result of classification and detection are shown in the form of tables and some graphs. The results are also depicted in visual image form.

5.1 Databases

In previous chapter the used dataset has been explained in detail. We have used our own dataset in our research for the purpose of training and testing. Which has been collected from traffic data of Rawalpindi/Islamabad Pakistan. The images and videos have been captured using SAMSUNG GALAXY C7 SM-C7000 in different lights, background and so on. The videos are mostly from the crossway bridge. The data acquisition, annotation, and preprocessing has been discussed in detail in previous chapter. For model training and simulation, we need to split the whole dataset into the train set, validation set, and test set. The split ratio is 7:2:1, the images dataset of both categories and the video dataset has been splitted into 70% train set, 20% validation set, and 10% test set. The training, validation, and test set of each data is given in Table 4.2.

Table 4. 2: Dataset

Dataset	Classes	Class name	Training	Validation	Testing
High density traffic scenes	3	car, motorcycle, person	3685	356	177
Low density traffic scenes	3	car, motorcycle, person	1260	120	60
COCO 2017	80	car, motorcycle, person, dog, table, horse etc	118287	5000	40760

5.2 Hardware Requirements

We simulate our proposed object detection algorithm on Google Colab pro. It is the platform provided to create python code and scripts online, mainly for the machine learning and data analysis. The Google Colab pro provides the large memory and disk space for efficient data analysis. It provides faster GPUs, the runtime is enhanced, and gives more virtual memory. The hardware requirement for the proposed methodology is 2 GB of RAM, the GPU of K80, T4, and P100. The CPU requirement is 2xvCPU. The table 5.1 shows the hardware requirement. In Google Colab pro, the dependencies and environment of the YOLO-v5 has been installed for the detection of object. After the YOLO-v5 environment installation, the model configuration and architecture has been defined. In our training, we have used the YOLO-v5s.

Table 5. 1: Google Colab Pro Hardware Configuration

Google Colab Pro Hardware Configuration	
GPU	K80, T40 and P100
CPU	2xvCPU
Memory	24GB
Price	$9.99/month

5.3 Performance Measures

As for now, precision, recall and mean average precision are used as evaluation criteria.

- Precision is a measure of, for how many time the model could correct correctly.
- The ratio of true positive with respect to the sum of true positive and false negative is called precision. The formula is written below in eq (1)

$$Precision = \frac{True\ Positive\ Cases}{Total\ Positive\ Predictions} \quad (1)$$

- Recall is a measure of does the model guessed right what it is supposed to be guessed.
- Recall is the ratio of the number of actual positives to the total variety of matching (relevant) objects. The formula is written below in eq (2)

$$Recall = \frac{True\ Positive\ Cases}{Total\ Cases} \quad (2)$$

- Average mean precision, the average value of precision is computer for value of recall over 0 to 1.

- mAP mainly applied in object detection algorithms for instance, SSD, Faster R-CNN.

5.4 Experimental Analysis

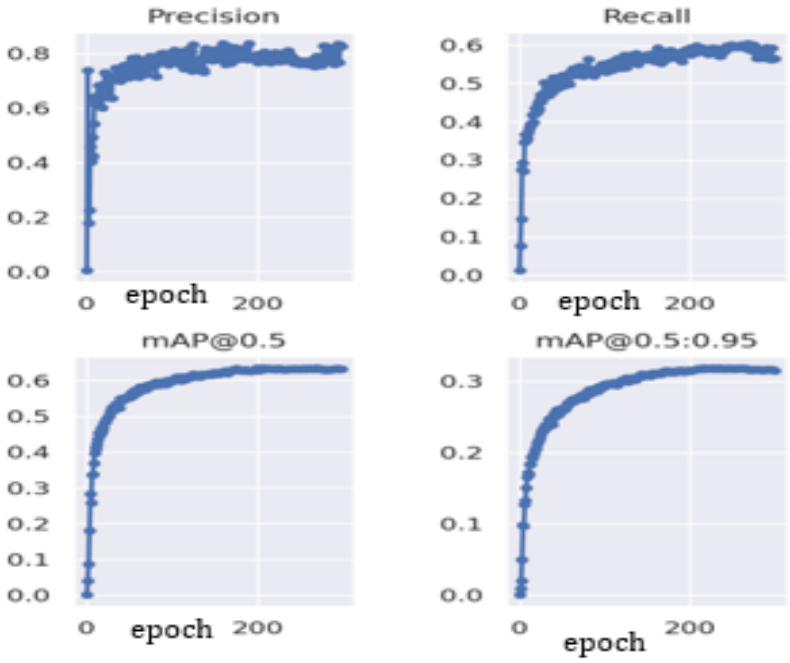
We have performed our training experiment with different parameters, the environment used was Google Colab pro, as discussed earlier. In this section the results with different batch size and epochs have been discussed along with the comparison of results with different parameters. We will discuss the results with respect to each category of our dataset.

5.4.1 High density traffic scenes:

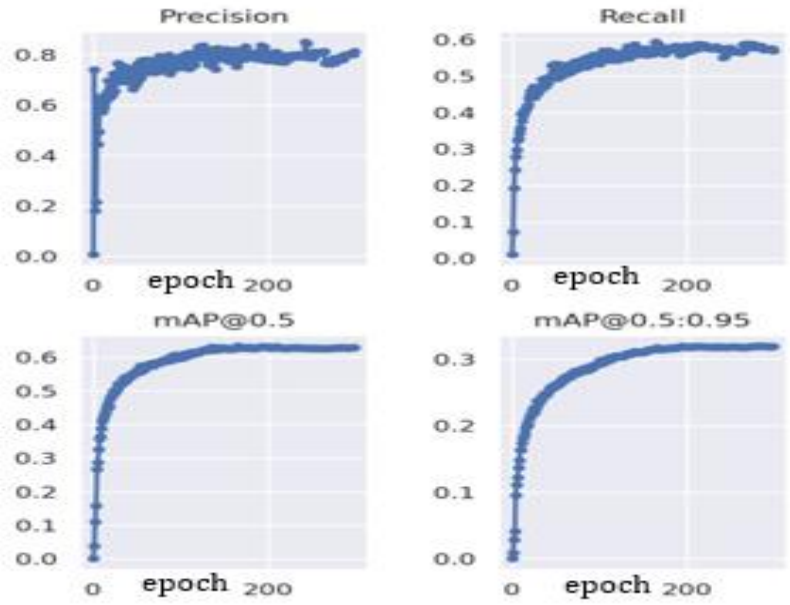
We have 1800 images, 15618 annotations and three classes. The total images after augmentations are 3685 training, 356 validation and 177 test images. We have performed our training experiment by changing the values of batch size. The training results are provided in table 5.1 on each category of high density traffic data. After the training, we used the fine-tuned weights for the detection on objects on the dataset. We get the values of predicted labels and the test images with the bounding boxes with confidence values. As can be seen in Table 5.1 that for batch size of 20 the highest precision of 0.832 is obtained for all classes. At each value of batch size, the graph of precision, recall and mAP is shown in Fig. 5.1. Also, the superimposed precision recall curve for each parameter value is shown in Fig. 5.2.

Table 5. 2: Training Summary of high density traffic scenes

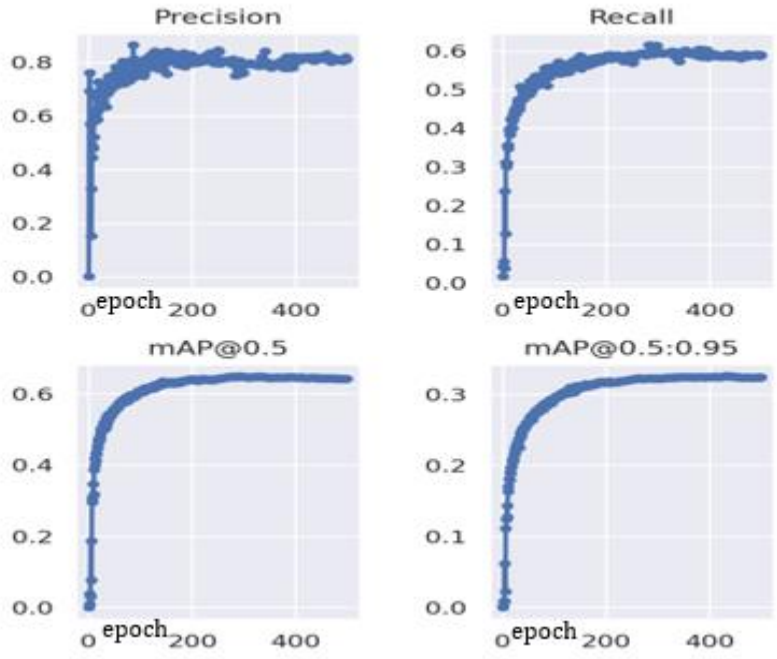
Epochs	Batch Size	Classes	P %	R %	map@0.5 %	map@0.5:.95 %
300	5	All	0.827	0.565	0.631	0.315
		Car	0.852	0.725	0.787	0.493
		motorcycle	0.853	0.586	0.652	0.289
		Person	0.776	0.384	0.455	0.164
300	10	All	0.813	0.57	0.629	0.319
		Car	0.842	0.727	0.785	0.489
		motorcycle	0.848	0.591	0.654	0.3
		Person	0.75	0.391	0.447	0.168
500	20	All	0.832	0.575	0.627	0.329
		Car	0.853	0.749	0.796	0.501
		motorcycle	0.856	0.575	0.623	0.316
		Person	0.787	0.402	0.46	0.17


Precision
Recall
mAP@0.5
mAP@0.5:0.95
epoch
0
200
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.8

(a)

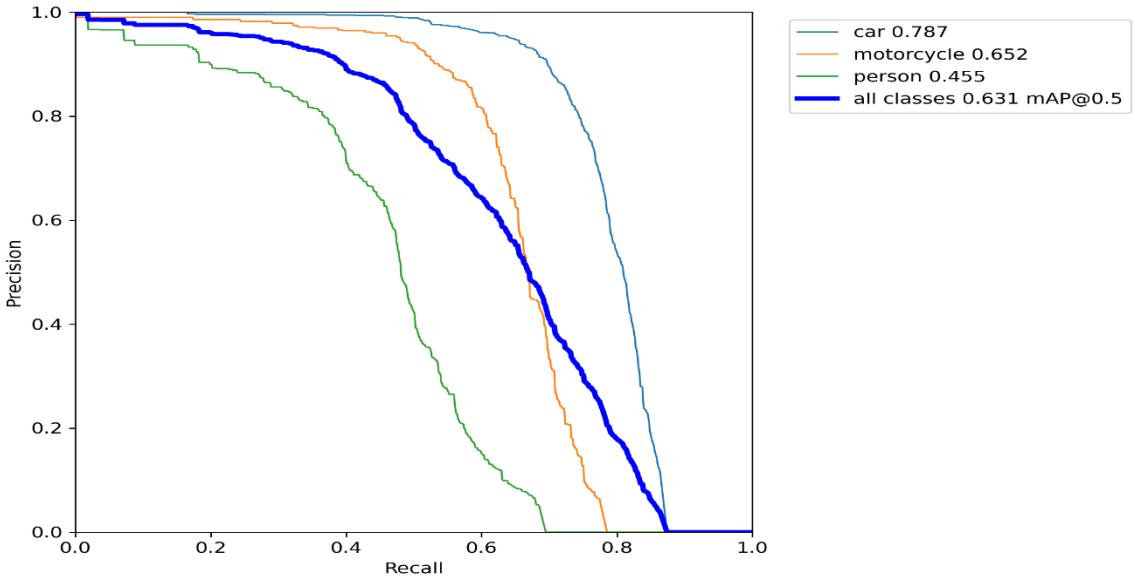

Precision
Recall
mAP@0.5
mAP@0.5:0.95
epoch
0
200
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.8

(b)

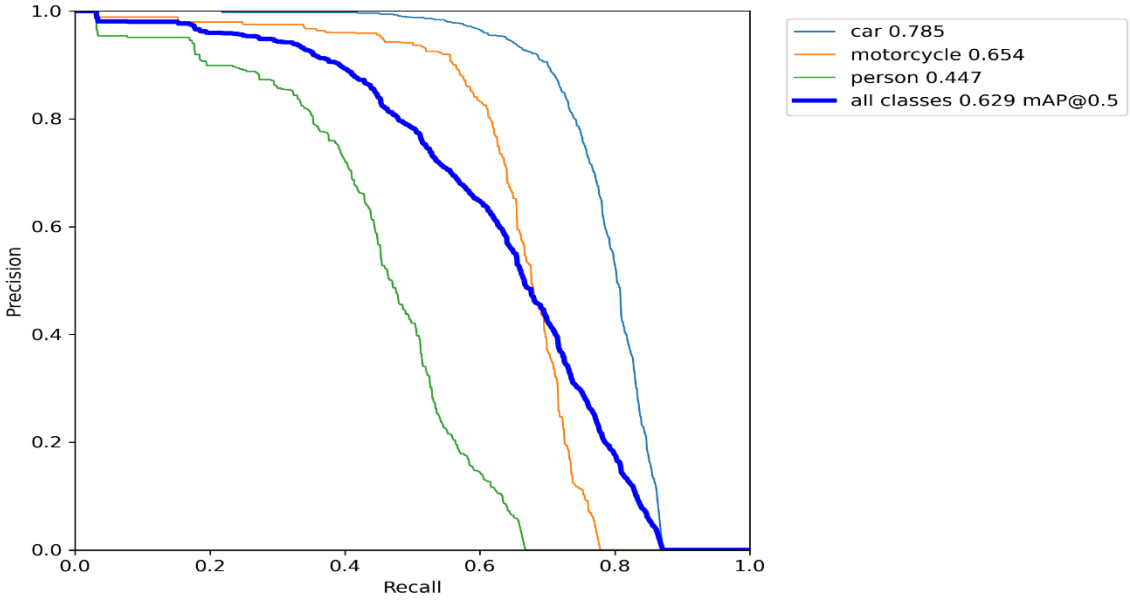


(c)

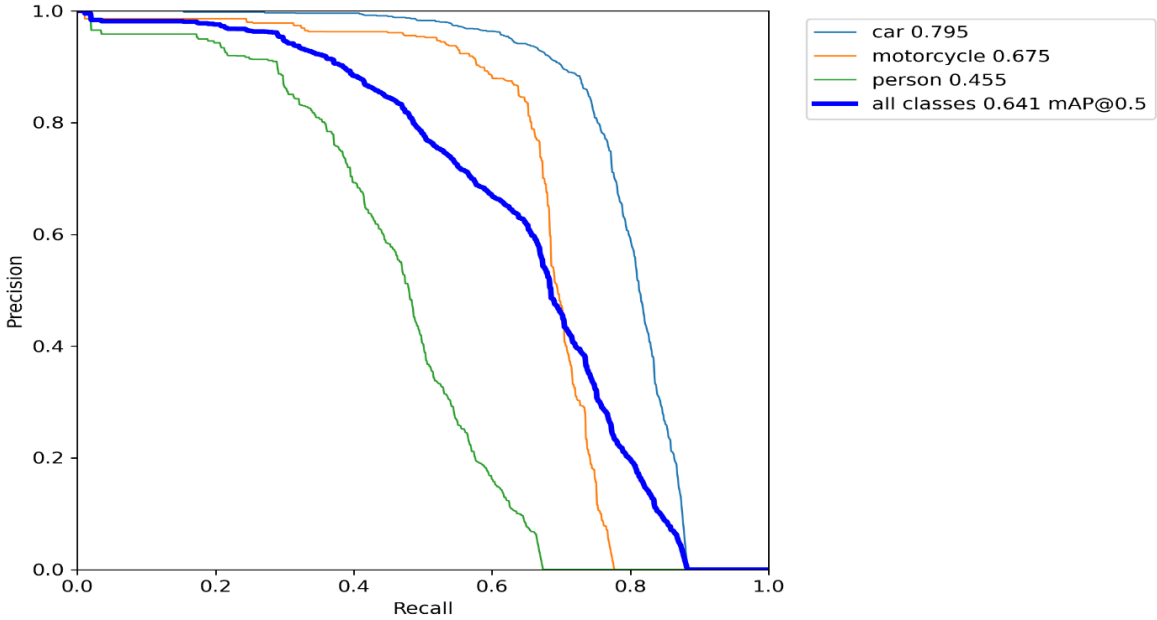
Figure 5. 1: Training results on high density traffic data: (a). Batch size 5, Epochs 300, (b). Batch size 10, Epochs 300, (c). Batch size 20, Epochs 500


car 0.787
motorcycle 0.652
person 0.455
all classes 0.631 mAP@0.5
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0

(a)


car 0.785
motorcycle 0.654
person 0.447
all classes 0.629 mAP@0.5
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0

(b)



(c)

Figure 5. 2: Precision/Recall curve on high density traffic data with: (a). Batch size 5, Epochs 300, (b). Batch size 10, Epochs 300, (c). Batch size 20, Epochs 500

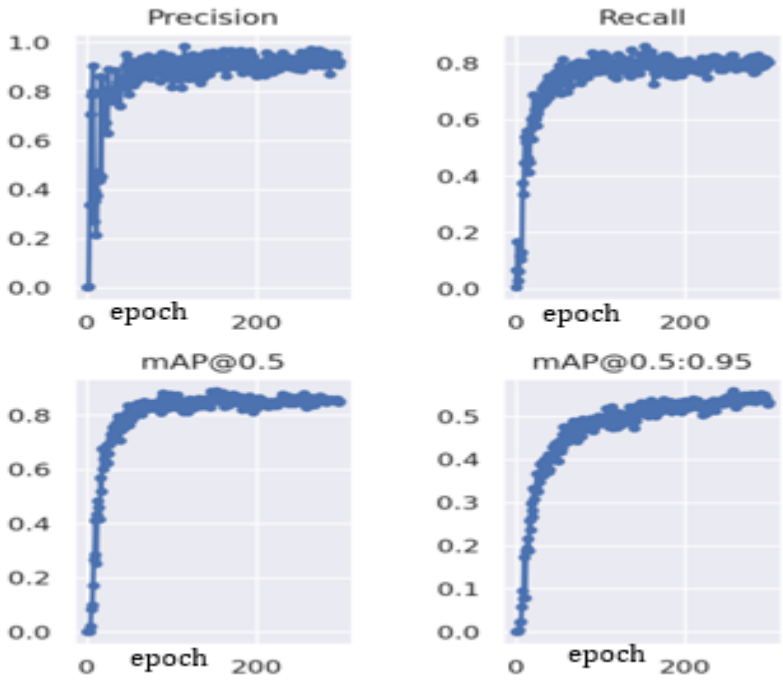
5.4.2 Low density traffic scenes:

We have 600 images, 903 annotations and three classes. The total images after augmentations are 1260 training,120 validation and 60 test images. We have performed our experiment by varying the values of batch size and epochs. The training results are provided in table 5.2 on each category of low density traffic data. As can be seen in Table 5.2 that for batch size of 20 with epoch set to 500, the highest precision of 0.983 is obtained for all classes. At each value of batch size and epochs the graph of precision, recall and mAP is shown in Fig. 5.3. Also, the precision recall curve for each parameter value is shown in Fig. 5.4.

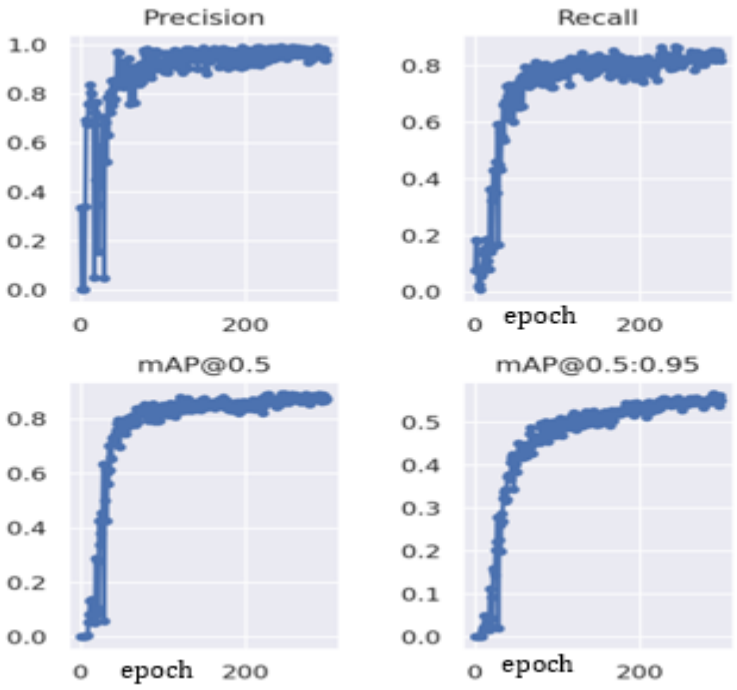
Table 5. 3: Training Summary of Low density traffic scenes

Epochs	**Batch Size**	**Classes**	**P %**	**R %**	**map@0.5 %**	**map@0.5:.95 %**
300	5	All	0.923	0.804	0.851	0.532
		Car	0.945	0.929	0.987	0.784
		motorcycle	0.948	0.875	0.9	0.508
		Person	0.875	0.609	0.667	0.304
300	30	All	0.959	0.817	0.872	0.551

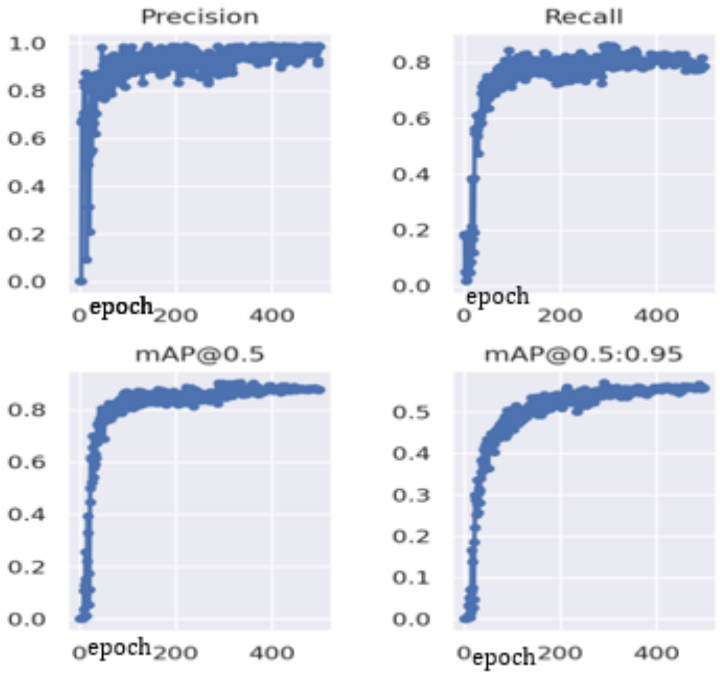
		Car	0.952	0.924	0.988	0.79
		motorcycle	0.988	0.875	0.924	0.582
		Person	0.937	0.652	0.704	0.334
500	20	All	0.983	0.788	0.877	0.557
		Car	0.968	0.93	0.988	0.789
		motorcycle	0.985	0.781	0.879	0.537
		Person	0.997	0.652	0.764	0.344



(a)

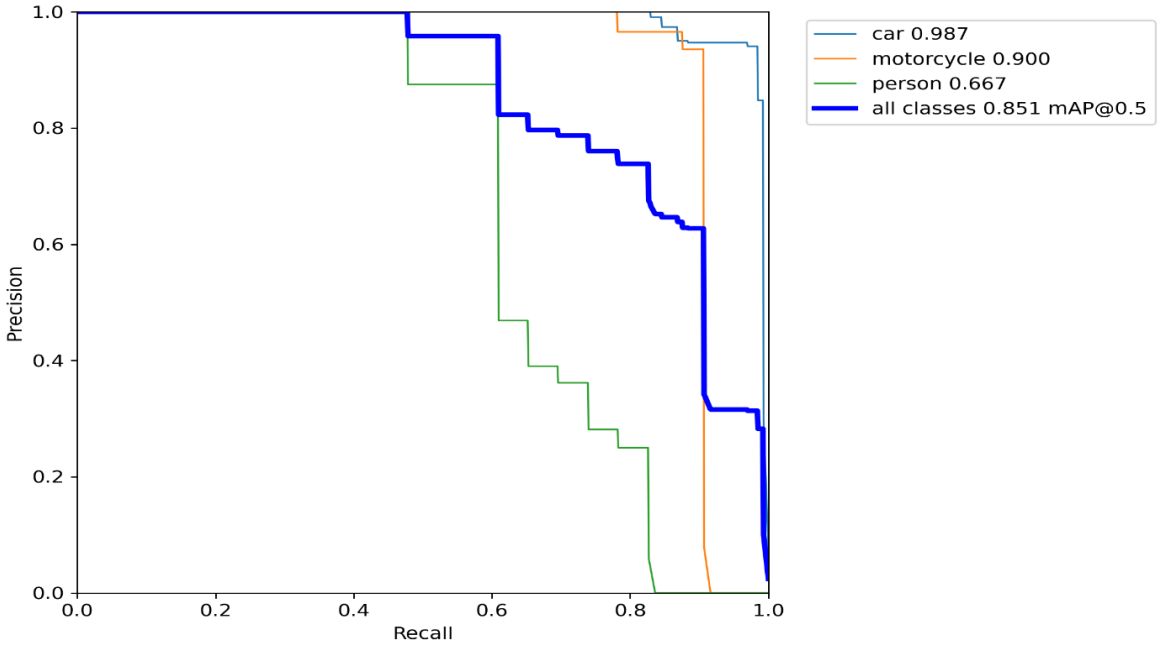


(b)

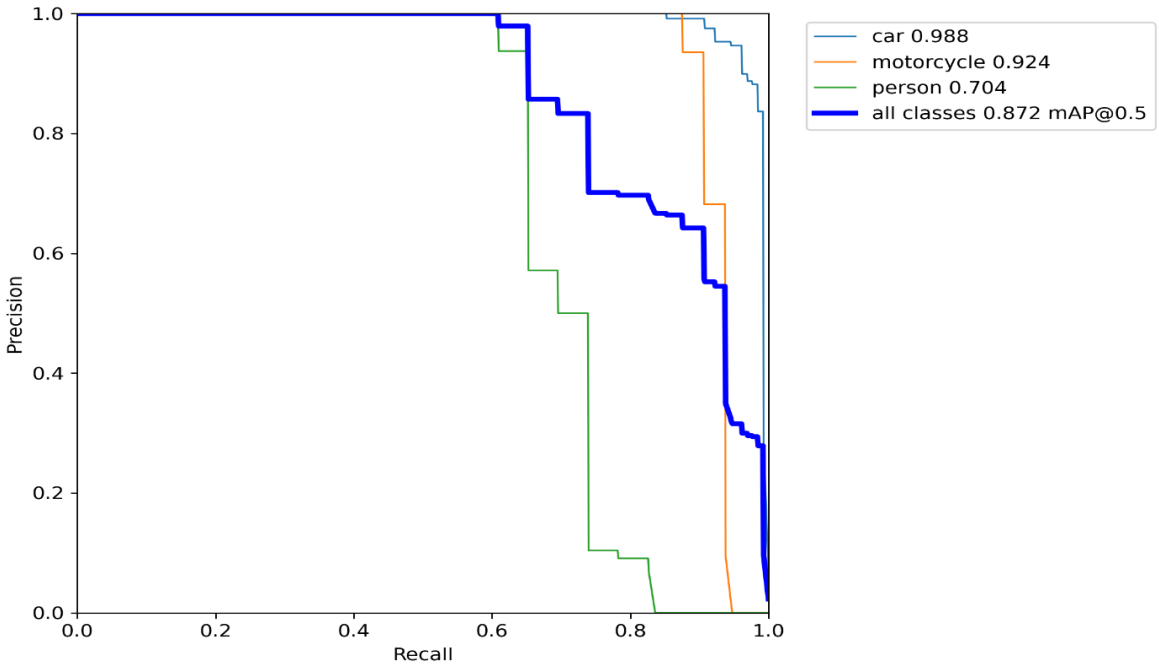


(c)

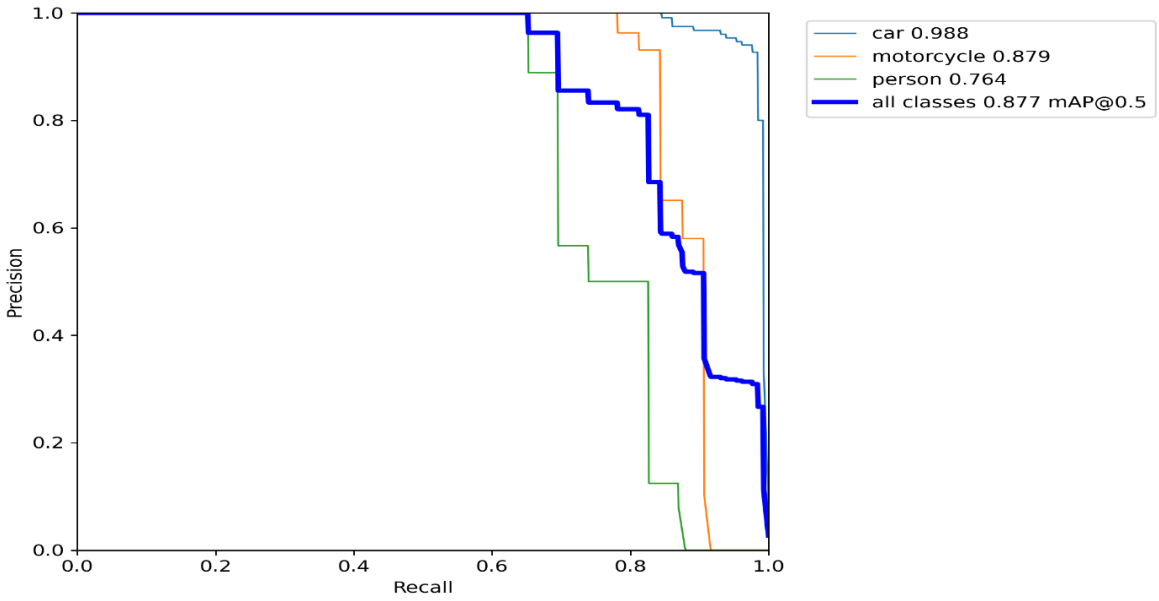
Figure 5. 3:Training results on low density traffic data: (a). Batch size 5, Epochs 300, (b). Batch size 30, Epochs 300, (c). Batch size 20, Epochs 500


car 0.987
motorcycle 0.900
person 0.667
all classes 0.851 mAP@0.5
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0

(a)


car 0.988
motorcycle 0.924
person 0.704
all classes 0.872 mAP@0.5
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0

(b)



(c)

Figure 5. 4: Precision/Recall curve on low density traffic data with: (a). Batch size 5, Epochs 300, (b). Batch size 30, Epochs 300, (c). Batch size 20, Epochs 500

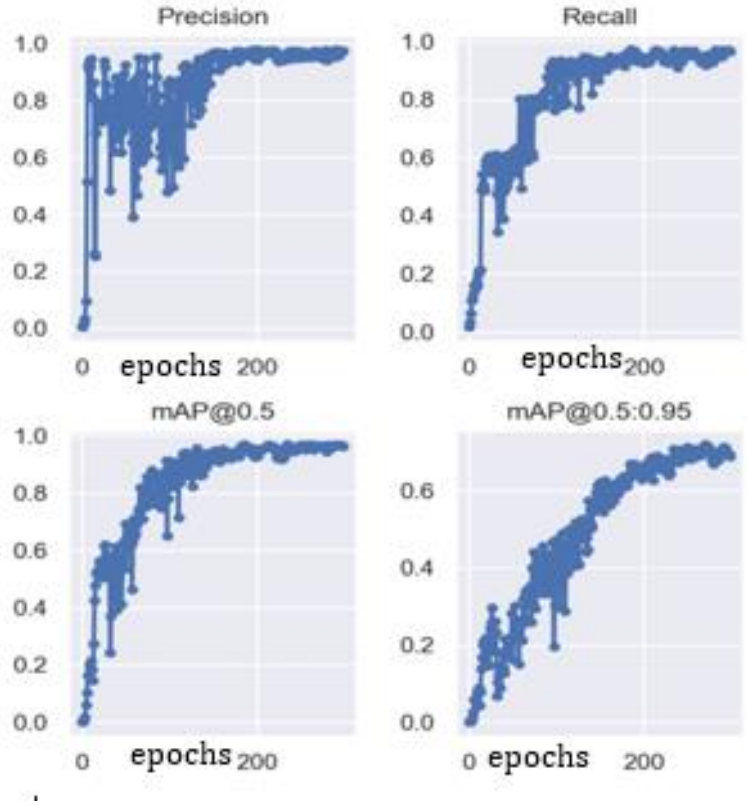
5.4.3 Video dataset:

We have performed experiment on 4 videos with three classes. We have performed our experiment by batch size 20 and epochs 300, as shown in table 5.3.

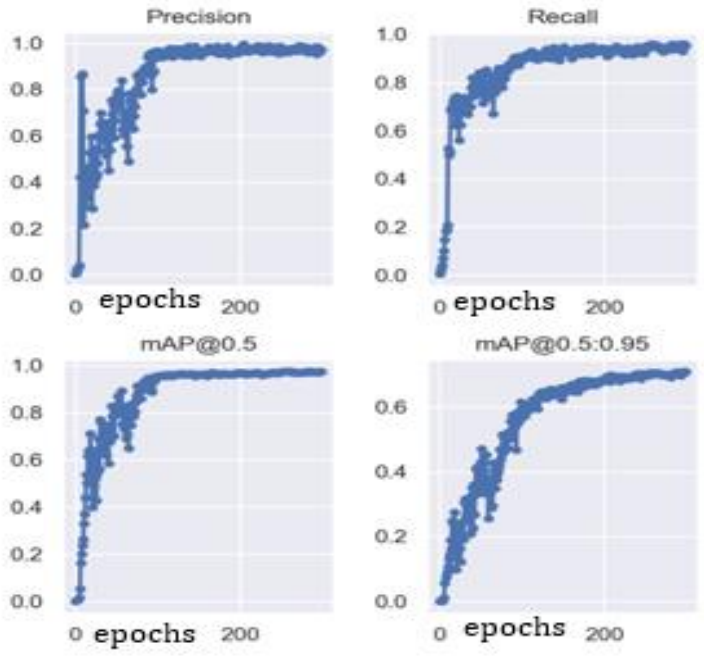
Table 5. 4: Training Summary on Video data

Dataset	Epochs	Batch Size	Classes	P %	R %	map@0.5 %	map@0.5:.95 %
Video1	300	20	All	0.973	0.965	0.964	0.687
			Car	0.977	0.946	0.978	0.798
			motorcycle	0.95	0.95	0.918	0.515
			Person	0.994	1	0.996	0.749
Video2	300	20	All	0.971	0.954	0.973	0.709
			Car	0.96	0.983	0.995	0.779
			motorcycle	0.964	0.947	0.964	0.676
			Person	0.991	0.93	0.961	0.67
Video3	300	20	All	0.958	0.942	0.966	0.642
			Car	0.984	0.994	0.997	0.8

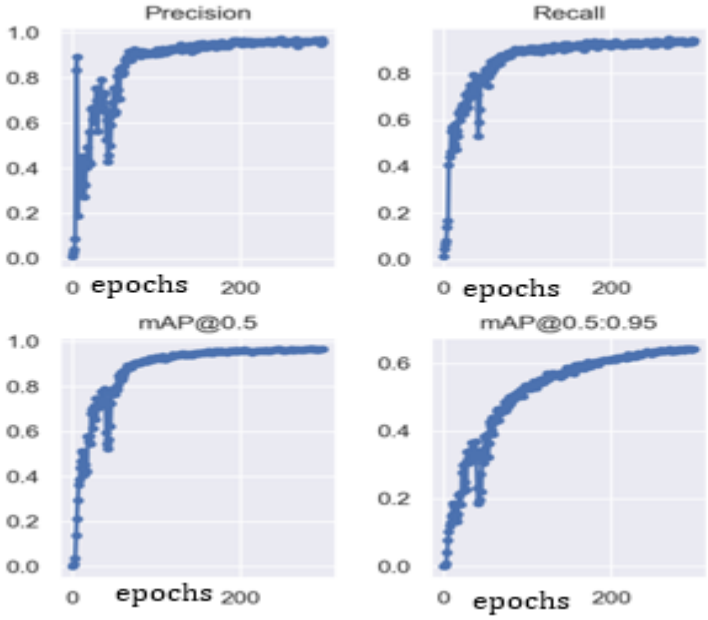
			motorcycle	0.978	0.968	0.986	0.638
			Person	0.912	0.863	0.917	0.489
Video4	300	20	All	0.945	0.939	0.958	0.636
			Car	0.954	0.983	0.992	0.707
			motorcycle	0.953	0.937	0.949	0.623
			Person	0.929	0.898	0.933	0.577



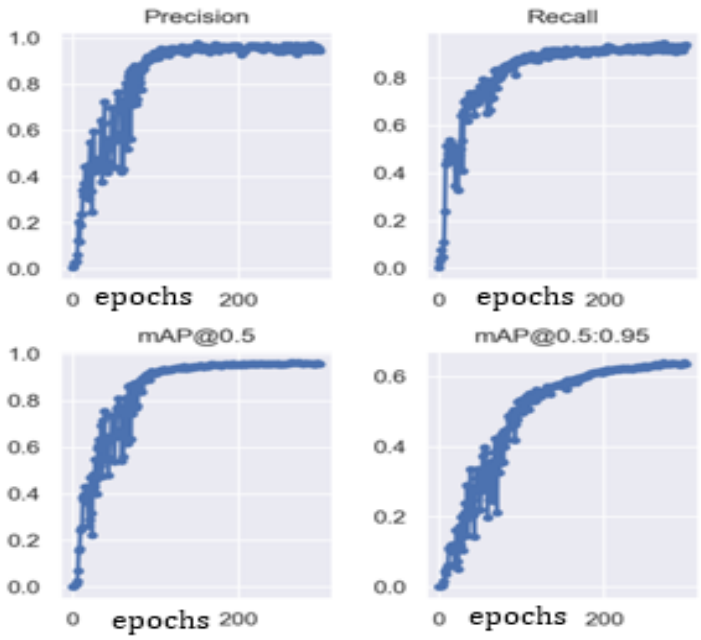
(a)


Precision
Recall
mAP@0.5
mAP@0.5:0.95
epochs
0
200

(b)

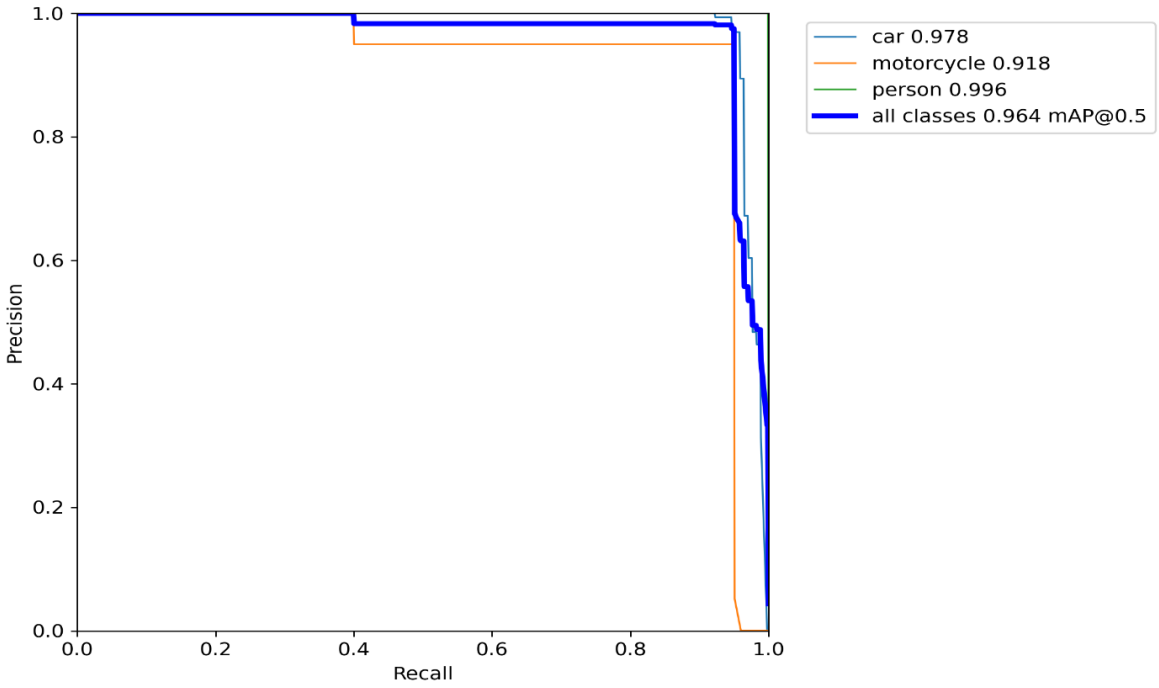

Precision
Recall
mAP@0.5
mAP@0.5:0.95
epochs
0
200

(c)

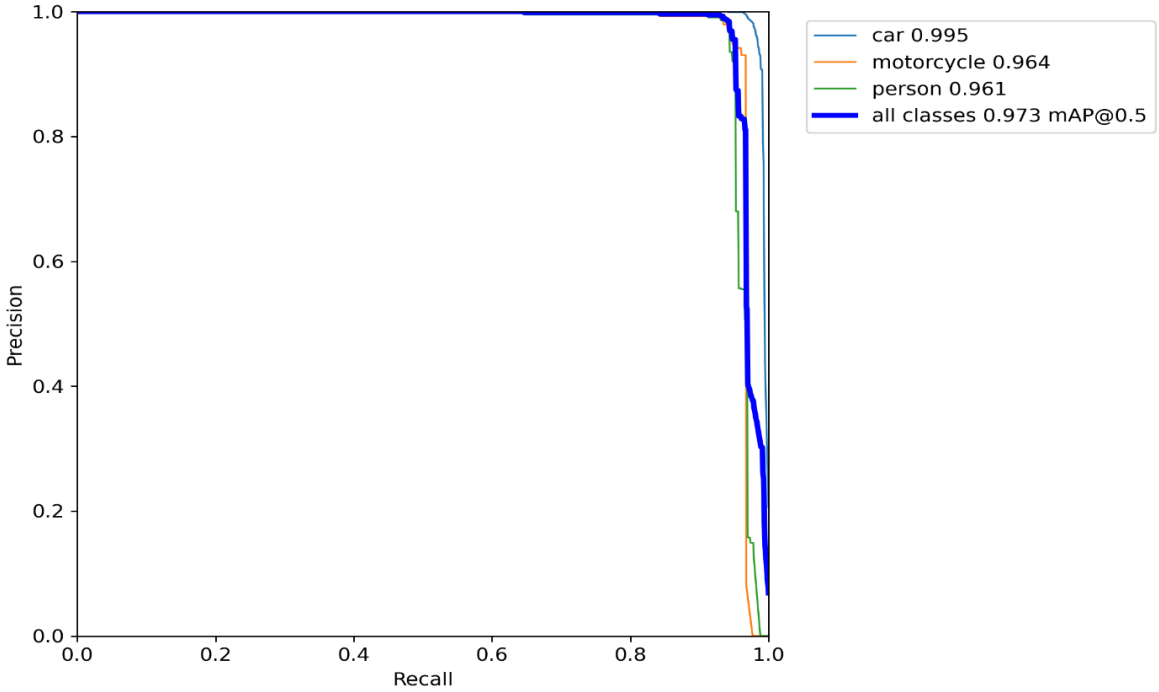


(d)

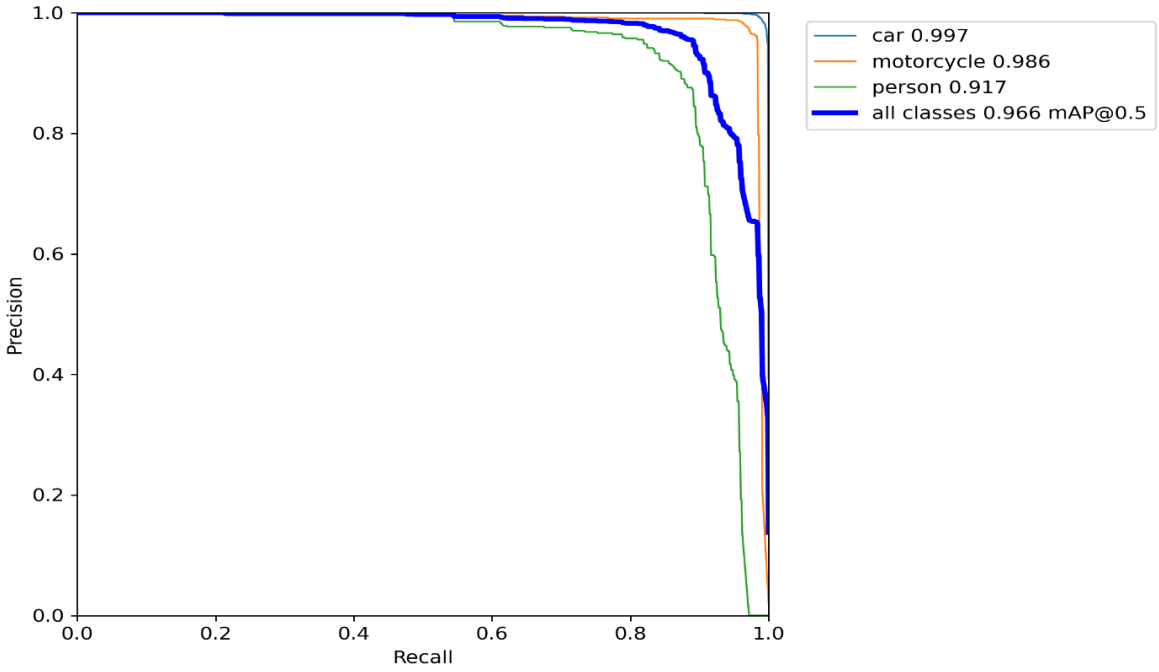
Figure 5. 5: Training results on Video traffic data: (a). Video 1, (b). Video 2, (c). Video



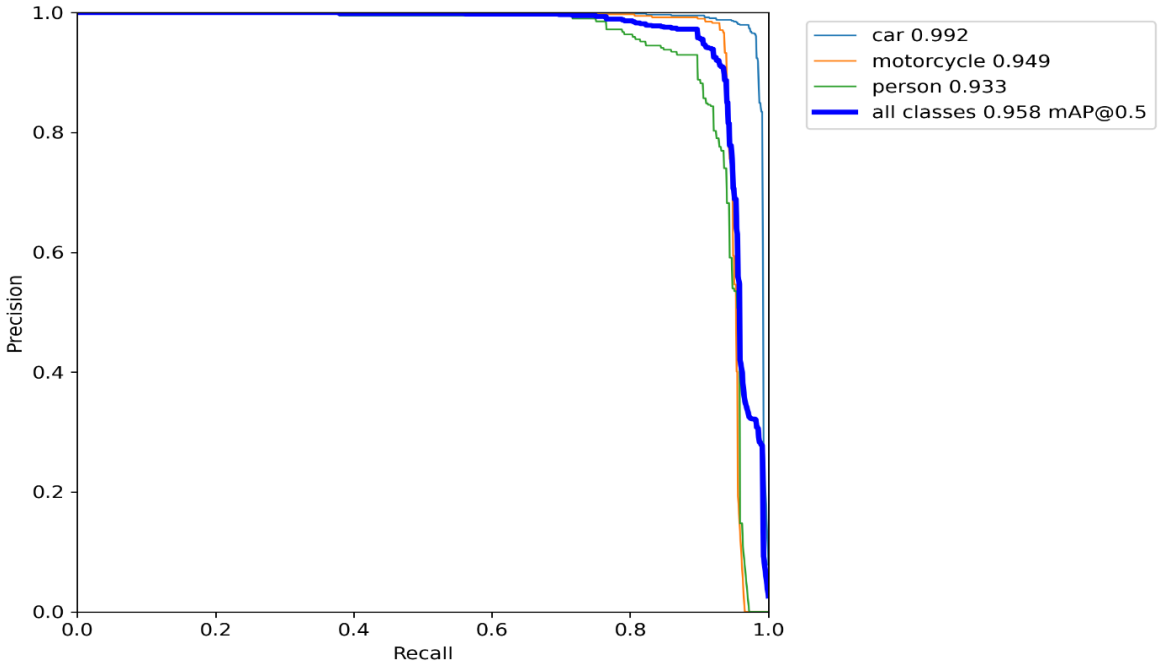
(a)


car 0.995
motorcycle 0.964
person 0.961
all classes 0.973 mAP@0.5
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0

(b)


car 0.997
motorcycle 0.986
person 0.917
all classes 0.966 mAP@0.5
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0

(c)



(d)

Figure 5. 6: Precision/Recall curve (a). Video 1 (b). Video 2 (c). Video 3 (d) Video 4

5.5 Object detection Analysis

To evaluate the qualitative performance of model, the model is tested with test images taken from dashboard camera of car. Fig. 5.7 shows the detailed detection results. From Fig. 5.7, important observations are described below.

- For high density traffic as shown in top row of Fig. 5.7(a), the proposed vehicle detection algorithm accurately detects all the vehicles of present study. For the rear and side views as shown by 1st and 2nd image of Fig. 5.7(a), the proposed method accurately locates the position of vehicles. Similarly, as shown in 4th image of Fig. 5.7(a), motorcycle and all the persons sitting on it are accurately detected.
- For low density traffic as shown in Fig. 5.7(b), the proposed vehicle detection algorithm accurately detects all the vehicles that appear in the image. Specially, the 5th image in the Fig 5.7(b), is a challenging image, in which resolution of the motorcycle is quite low. Yet the proposed vehicle detector accurately locates the vehicles.
- For partially occluded cases as shown in 1st and 2nd images in Fig. 5.7(c), the proposed algorithms perfectly detect all the cars despite being only partially visible or occluded. Moreover, the 3rd and 4th image of Fig. 5.7(c) show that cars and motorcycle in low illumination are precisely detected.

- For video traffic data as shown in Fig. 5.7(d), the proposed algorithm is capable to detect all overlapped objects. In this case, it can be observed that most of the images have been captured from long distance and resolution of the objects that appear in the image is also small. Yet the detector accurately detects and classifies the different vehicles.

(a)

(b)

(c)

(d)

Figure 5. 7: Detection results on: (a). High density traffic, (b). Low density traffic, (c). Partially occluded and low illumination, and (d). Video traffic data

CHAPTER 6

CONCLUSION & FUTURE WORK

6.1 Conclusion

This paper presented an accurate, fast, and robust object detection method, which is based on the YOLO-v5 in difficult traffic scenes. We developed our own dataset of three well-known objects, which are car, motorcycle, and person and also fine-tuned the weights of COCO dataset. To develop a robust object detection algorithm, we performed transfer learning, which is an optimized method. The proposed object detection method gives the highest accuracy of 0.832 for high density traffic data and 0.983 for low density traffic data on challenging Pakistani highways. Moreover, the proposed model gives highest accuracy of 0.971 for video dataset.

This model performs very well in detecting small and dense objects even in complicated and dense traffic places. This technique significantly elevated the accuracy and operational efficiency. In addition, the detection technique proposed in this research can additionally be relevant to a large number of real time applications, however the only premise is that a giant quantity of data is required for training of detection model.

6.2 Contribution

Our main contributions are listed below.

- We collect our own object dataset in Pakistani territory. The developed dataset contains images and videos of high and low density traffic patterns. To the best of our knowledge, limited dataset is available for challenging environment of Pakistani roads and highways.
- We propose a robust supervised vehicle detection method that can detect multiple objects that appear in an image. Moreover, the proposed method is also capable to detect various objects in large groups irrespective of the resolution of objects. Furthermore, the proposed method performs exceptionally well under partial occlusion and low illuminations conditions.
- Our proposed method is fast and yields good accuracy with minimum false positives. Moreover, the proposed algorithm is user friendly and is computationally efficient. We are optimistic that proposed algorithm will be useful for traffic control and surveillance staff.

6.3 Future Work

The proposed system is quite efficient for detection and classification at traffic scenes at different scenarios, also it provided efficient results. In future dataset can be extend by adding more images and adding more classes like traffic signals etc. this system can be trained and modified for the detection of objects at any kind of situation, such as night time, low light or could be combined with the license plate recognition to make a surveillance system. In the future work precision of the algorithm could be improved with training on bigger and more diverse dataset that cover different weather and lighting conditions.

REFERENCES

[1]. Z. Mahmood, O. Haneef, N. Muhammad, and S. Khattak, “Towards a Fully Automated Car Parking System,” *IET Intelligent Transport Systems*, vol. 13, no. 2, 2018, pp. 293−302.

[2]. Z. Mahmood, *et. al*, “Towards Automatic License Detection,” *Sensors*, vol. 22, no. 3, 2022, pp. 1–19.

[3]. Z. Mahmood, N. Bibi, M. Usman, U. Khan, and N. Muhammad, “Mobile Cloud based Framework for Sports Applications,” *Multidimensional Systems and Signal Processing*, vol. 30, no. 4, 2019, pp. 1991−2019.

[4]. M. Madhusri, S. Banerjee, and S. S. Chaudhuri, “Faster R-CNN and YOLO based Vehicle detection: a survey,” *5th International Conference on Computing Methodologies and Communication (ICCMC)*, 2021, pp. 1442–1447.

[5]. R. Joseph, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” *In Proceedings of Intl. Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[6]. Hamsa *et. al*, “Automatic Vehicle Detection from Aerial Images using Cascaded Support Vector Machine and Gaussian Mixture Model,” *Intl; Conf on Signal Process and Inf; Security (ICSPIS)*, 2018, pp.1–4.

[7]. M. Mikaty and T. Stathaki, “Detection of Cars in High Resolution Aerial Images of Complex Urban Environments”, *IEEE Trans. Geoscience and Remote Sensing*, vol. 55, no. 10, 2017, pp. 5913-5924.

[8]. C. Sun, A. Shrivastava, S. Singh and A. Gupta, "Revisiting Unreasonable Effectiveness of Data in Deep Learning Era," *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 843-852, doi: 10.1109/ICCV.2017.97.

[9]. B. Wu and R. Nevatia, "Simultaneous Object Detection and Segmentation by Boosting Local Shape Feature based Classifier," *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1-8, doi: 10.1109/CVPR.2007.383042.

[10]. Liu, L., Ouyang, W., Wang, X. *et al.* Deep Learning for Generic Object Detection: A Survey. *Int J Comput Vis* **128,** 261–318 (2020). https://doi.org/10.1007/s11263-019-01247-4

[11]. Hernández-Orallo, J. (2017). Evaluation in artificial intelligence: from task-oriented to ability-oriented measurement. Artificial Intelligence Review, 48(3), 397-447.

[12]. Q. Bai, S. Li, J. Yang, Q. Song, Z. Li and X. Zhang, "Object Detection Recognition and Robot Grasping Based on Machine Learning: A Survey," in *IEEE Access*, vol. 8, pp. 181855-181879, 2020, doi: 10.1109/ACCESS.2020.3028740.

[13]. CNN vs. RNN vs. ANN – Analyzing 3 Types of Neural Networks in Deep Learning" https://www.analyticsvidhya.com/blog/2020/02/cnn-vs-rnn-vs-mlp-analyzing-3-types-of-neural-networks-in-deep-learning/"

[14]. Basic Neural Network at "https://towardsdatascience.com/step-by-step-guide-to-building-your-own-neural-network-from-scratch-df64b1c5ab6e [Accessed 10th Feb. 2022]

[15]. Montalbo, F. J. P., & Alon, A. S. (2021). Empirical Analysis of a Fine-Tuned Deep Convolutional Model in Classifying and Detecting Malaria Parasites from Blood Smears. KSII Transactions on Internet and Information Systems (TIIS), 15(1), 147-165..

[16]. Van Herk, M. (1992). A fast algorithm for local minimum and maximum filters on rectangular and octagonal kernels. Pattern Recognition Letters, 13(7), 517-521.

[17]. Object detection "https://towardsdatascience.com/object-detection-simplified-e07aa3830954" [Accessed 14th March, 2022]

[18]. YOLO timeline https://www.v7labs.com/blog/yolo-object-detection, [Accessed 12th Feb. 2022]

[19]. J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6517-6525, doi: 10.1109/CVPR.2017.690.

[20]. Redmon, J. and Farhadi, A., 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

[21]. Bochkovskiy, A., Wang, C.Y. and Liao, H.Y.M., 2020. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.

[22]. Use Yolo v5 Object Detection Algorithm for Custom Object Detection. www.analyticsvidhya.com. [Accessed: March 5, 2022]

[23]. Hamsa *et. al*, "Automatic Vehicle Detection from Aerial Images using Cascaded Support Vector Machine and Gaussian Mixture Model," *Intl; Conf on Signal Process and Inf; Security (ICSPIS)*, 2018, pp.1–4.

[24]. M. Mikaty and T. Stathaki, "Detection of Cars in HighResolution Aerial Images of Complex Urban Environments", *IEEE Trans. Geoscience and Remote Sensing*, vol. 55, no. 10, 2017, pp. 5913-5924.

[25]. Ç. Arı and S. Aksoy, "Detection of Compound Structures Using a Gaussian Mixture Model With Spectral and Spatial Constraints," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 10, 2014, pp. 6627–6638.

[26]. A. Hbaieb, J. Rezgui and L. Chaari, "Pedestrian Detection for Autonomous Driving within Cooperative Communication System," *Wireless Comm and Networki Conf (WCNC)*, 2019, pp. 1–6.

[27]. L. Xiong, W. Yue, Q. Xu, Z. Zhu and Z. Chen, "High Speed Front-Vehicle Detection Based on Video Multi-feature Fusion," *2020 IEEE 10th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, 2020, pp. 348–351.

[28]. T. Yawen and G. Jinxu, "Research on Vehicle Detection Technology Based on SIFT Feature," *8th International Conf on Electronics Info. and Emergency Communication (ICEIEC)*, 2018, pp. 274–278.

[29]. K. He and L. Zhang, "Vehicle Detection in Satellite Images with Deep Neural Networks and Vehicle Shape Features," *International Geoscience and Remote Sensing Symposium (IGARSS)*, 2021, pp. 4212–4215.

[30]. P. Saini, K. Bidhan and S. Malhotra, "A Detection System for Stolen Vehicles Using Vehicle Attributes With Deep Learning," *5th International Conference on Signal Processing, Computing and Control (ISPCC)*, 2019, pp. 251–254.

[31]. R. Kulkarni, S. Dhavalikar and S. Bangar, "Traffic Light Detection and Recognition for Self Driving Cars Using Deep Learning," *4th International Conference on Computing Communication Control and Automation (ICCUBEA)*, 2018, pp. 1–4.

[32]. Q. Tan, J. Ling, J. Hu, X. Qin and J. Hu, "Vehicle Detection in High Resolution Satellite Remote Sensing Images Based on Deep Learning," *IEEE Access*, vol. *8*, 2020, pp.153394–153402.

[33]. M. Sheng, C. Liu, Q. Zhang, L. Lou and Y. Zheng, "Vehicle Detection and Classification Using Convolutional Neural Networks," *7th Data Driven Control and Learning Sys Conf (DDCLS)*, 2018, pp. 581–587.

[34]. Y. Miao, F. Liu, T. Hou, L. Liu and Y. Liu, "A Nighttime Vehicle Detection Method Based on YOLO v3," *2020 Chinese Automation Congress (CAC)*, 2020, pp. 6617-6621, doi: 10.1109/CAC51589.2020.9326819.

[35]. L. Shengyu, B. Wang, H. Wang, L. Chen, M. Linjian, and X. Zhang, "A real-time object detection algorithm for video," *Computers & Electrical Engineering*, vol. 77, 2019, pp. 398–408.

[36]. P. Haolong, S. Guo, and X. Zuo. "A Vehicle Detection Method Based on YOLOV4 Model," *2nd International Conference on Artificial Intelligence and Information Systems*, 2021, pp. 1–4.

[37]. J. Lin and M. Sun, "A YOLO-Based Traffic Counting System," *2018 Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, 2018, pp. 82-85, doi: 10.1109/TAAI.2018.00027.

[38]. C. Wang, H. Wang, F. Yu and W. Xia, "A High-Precision Fast Smoky Vehicle Detection Method Based on Improved Yolov5 Network," *2021 IEEE International Conference on Artificial Intelligence and Industrial Design (AIID)*, 2021, pp. 255-259, doi: 10.1109/AIID51893.2021.9456462.

[39]. Al-qaness, M. A., Abbasi, A. A., Fan, H., Ibrahim, R. A., Alsamhi, S. H., & Hawbani, A. (2021). An improved YOLO-based road traffic monitoring system. *Computing*, *103*(2), 211-230.

[40]. H. Bingqiang, H. Lin, Z. Hu, X. Xiang, and J. Yao, "An improved YOLOv3-tiny algorithm for vehicle detection in natural scenes," *IET Cyber-Systems and Robotics*, no. 3, 2021, pp. 256–264.

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

[42]. Sang, J., Wu, Z., Guo, P., Hu, H., Xiang, H., Zhang, Q., & Cai, B. (2018). An improved YOLOv2 for vehicle detection. *Sensors*, *18*(12), 4272.

[43]. Putra, M. H., Yussof, Z. M., Lim, K. C., & Salim, S. I. (2018). Convolutional neural network for person and car detection using YOLO framework. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, *10*(1-7), 67-71.

[44]. Cepni, S., Atik, M. E., & Duran, Z. (2020). Vehicle detection using different deep learning algorithms from image sequence. *Baltic Journal of Modern Computing*, *8*(2), 347-358.

[45]. Huang, R., Pedoeem, J., & Chen, C. (2018, December). YOLO-LITE: a real-time object detection algorithm optimized for non-GPU computers. In 2018 IEEE International Conference on Big Data (Big Data) (pp. 2503-2510). IEEE.

[46]. COCO 2017 dataset " https://www.kaggle.com/awsaf49/coco-2017-dataset, " [Accessed: , 22nd Feb, 2022]

[47]. COCO and pascal dataset "https://towardsdatascience.com/coco-data-format-for-object-detection-a4c5eaf518c5, [Accessed 22nd February 2022]

[48]. Hawkins, D. M. (2004). The problem of overfitting. Journal of chemical information and computer sciences, 44(1), 1-12.

[49]. YOLOv5 Architecture https://machinelearningknowledge.ai/introduction-to-yolov5-object-detection-with-tutorial´, [Accessed 27th February 2022]

[50]. Yang, X. S. (2019). *Introduction to algorithms for data mining and machine learning*. Academic press.

[51]. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).

[52]. Transfer learning https://www.topbots.com/transfer-learning-in-nlp/', [Accessed 20th February 2022]