T.C.

ALTINBAS UNIVERSITY

Institute of Graduate Studies

Information Technologies

REAL-TIME DETECTION TO SOLVE TRAFFIC CONGESTION PROBLEMS

Haider Adil Kahleel Al-Zubaidy

Master of Science

Supervisor

Prof. Dr.Osman Nuri UÇAN

Istanbul, 2020

REAL-TIME DETECTION TO SOLVE TRAFFIC CONGESTION PROBLEMS

by

Haider Adil Kahleel Al-Zubaidy

Information Technologies

Submitted to the Institute of Graduate Studies

In partial fulfillment of the requirements for the degree of

Master of Science

ALTINBAŞ UNIVERSITY

2020

The thesis titled “REAL-TIME DETECTION TO SOLVE TRAFFIC CONGESTION PROBLEMS” prepared and presented by “Haider Adil Kahleel Al-Zubaidy” was accepted as a Master of Science Thesis in Information Technologies.

Prof. Osman Nuri UÇAN

Supervisor

Thesis Defense Jury Members:

Prof. Osman Nuri UÇAN	School of Engineering and Natural Sciences, Altınbaş University	_____
Asst. Prof. Doğu Çağdaş ATİLLA	School of Engineering and Natural Sciences, Altınbaş University	_____
Asst. Prof. İzzet Paruğ DURU	Vocational School of Higher Education, İstanbul Gedik University	_____

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Approval Date of Institute of Graduate Studies:
____/____/____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Haider Adil Kahleel AL-Zubaidy

DEDICATION

I dedicate this work to my Father. (Prof. Dr. Adil Kahleel AL-Zubaidy)

ACKNOWLEDGEMENTS

Even words cannot express my heartfelt and gratitude. I am pleased to thank my father and mother, who worked hard and watched over my upbringing and education since childhood. And I should pause to go back to years ago in the university with professors and classmates; who have supported me with much of the sciences and considerable efforts.

Specifically mentioning:

Prof. Dr.Osman Nuri UÇAN,

He did not skimp on me with his valuable advice, which helped me to complete this thesis.

Also, I would like to thank. My friend Mahmood A.Jumaah and all my other friends.

ABSTRACT

REAL-TIME DETECTION TO SOLVE TRAFFIC CONGESTION PROBLEMS

Al-Zubaidy, Haider Adil Kahleel ,

M.Sc., Information Technologies, Altınbaş University,

Supervisor: Prof. Dr.Osman Nuri UÇAN

Date: 11 / 2020

Pages: 87

Traffic congestions in the main roads have become a rising issue because of the inconsistent and ineffective infrastructure, the mismatch between the urbanization rate, and the increase of vehicles in the developed countries. Even when several technologies reduce traffic congestion, these have produced inconsistent and ineffective results, which calls for immediate attention in real-life traffic detection through advanced cameras and applications. In this thesis, the researcher shows a system for the detection of traffic congestion in real time by using surveillance cameras (CCTV) as a data source to determine the status of traffic congestion. The initial stage of the proposed system was the pre-processing step using illumination correction and noise reduction. In the processing stage, the video has been fragmented into several frames on the basis of frame for each second (fps) related to video (for instance, 30 fps); these frames represent the classifier's input. Two classifiers have been used. First, the Haar Cascade classifier based on digital image processing techniques. Second, a Deep Learning classifier based on machine learning techniques. These classifiers were utilized for the detection of the vehicles and counting the number of vehicles per time threshold using surveillance cameras. The final decision has been specified on a basis of number of the vehicles during specified time as well as on several stages to evaluate the condition (traffic congestion, non-traffic). Experimental results have been performed using several videos during different periods. For example, they were processing each second or every five seconds of the entire video duration. Haar cascade achieves an accuracy average

of 77.41%, F-score average of 80.06%, and the error average of 0.2%. Deep learning achieves an accuracy average of 97.11%, F-score average of 96.15%, and the error average of 0.025%. Therefore, the deep learning method has been relied upon, where the results show higher accuracy and less processing time than haar cascade. The experimental results are showing that the suggested system might be adequately estimating traffic congestion in real time at low costs in comparison to other traffic congestion estimation systems.

Keywords: Traffic Congestion, Deep Learning, Haar Cascade, Vehicles Detection, Object Recognition, Machine Learning.

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURE

Pages

LIST OF ABBREVIATIONS

AI:	:	Artificial Intelligence
ANN	:	Artificial Neural Network
BFLOP	:	Billion Floating-Point Operations
CCTV	:	Closed-Circuit Television
CNN	:	Convolutional Neural Network
CUDA	:	(Compute Unified Device Architecture)
DCCN	:	Data Communication and Computer Networking
DCNN	:	Deep Neural Bypass Network
DIP	:	Digital Image Processing
DNN	:	Deep Neural Network
FN	:	False Negative
FNR	:	The False-Negative Rate
FP	:	False Positive
FPN	:	Feature Pyramid Network
FPR	:	The False-Positive Rate
FPS	:	Frames Per Second
GMM	:	Gaussian Mixture Model
GPU	:	Graphics Processing Unit
GSM	:	Global System for Mobile Communications
HOG	:	Histograms of Oriented Gradients
MCMC	:	Markov Monte Carlo
NN	:	Neural Network
OBU	:	On-Board Unit
OF-MCMC	:	Optical Flow Markov Monte Carlo
OMNeT++	:	Objective Modular Network Testbed in C++
R-FCN	:	Region-Based Fully Convolutional Networks
RFID	:	Radio-Frequency Identification
RPN	:	Region Proposal Network
SIFT	:	Scale-Invariant Feature Transform
SN	:	Sensitivity
SP	:	Specificity
SRGB	:	Standard Red Green Blue
SRGB	:	Standard RGB
SSD	:	Single Shot Multibox Detector
SUMO	:	Simulation of Urban Mobility
SVM	:	Support Vector Machine
TN	:	True Negative
TP	:	True Positive
UAVs	:	Unmanned Aerial Vehicle
V2V	:	Vehicle-to-Vehicle Communications

VANETs	:	Vehicular Ad Hoc Networks
VOC	:	Visual Object Classes
YOLO	:	You Only Look Once

1. INTRODUCTION

Traffic congestion is considering a real problem and on urban roads networks has become increasingly problematic since the 1950s. It is bring about major economic loss, increase drive times and aggravate the pollution. Traffic congestion can be defined as an annoyance with which the majority of drivers were used to; it is consuming fuel, money and time. For example, a typical auto passenger into the United States spent extra 34 hrs in his car as well as wasting fourteen fuel gallons due to congestion [1]. Governments provide large amounts of money to try to identify and encounter traffic congestion, but this seems difficult because of the complex nature of traffic congestion [2]. These negative impacts can be reduced after defining the main drivers of highway traffic congestion [3]. Currently, highway congestion is the national focus. Traffic congestion seriously affecting the natural travel of person and restricting the society's economic development, this this is an issue which should be quickly resolved. Due to the significance and prevalence of traffic control systems, the vehicles' detection was difficult problem because it involves criminalizing a large group of diverse objects from an unknown background [4]. This calls for an immediate attention to reduce traffic congestion through the use of information technology, where cameras and other technologies can be used to detect traffic jams through the use of certain algorithms. The advantage of the technologies related to image and video processing for detecting highway congestions is highly important [5].

The availability of advanced technologies have played a critical role in improving operations and services, but these have not yet been optimally leveraged and capitalized upon to reduce traffic congestions [6]. Even when countries have access to resources, real time detection of vehicular movement has not been prioritized, nor is there any particular algorithm to ensure the possibility of reducing or eliminating traffic congestion in major roads. This is an appalling case in developed countries, and has resulted in the frustration and dissatisfaction of the general public in worst case scenario. With this existing flaw in the system and lack of advanced technologies being used to solve traffic jams, this research has specifically looked into the real time detection aspect as a way to solve the existing issues in major roads [7]. This can be beneficial for both developing and developed countries, and may help in

increasing the satisfaction and happiness level of general population because of reduced or non-existent traffic jams.

1.1 WHAT IS TRAFFIC CONGESTION

Traffic congestion on highways is all the problems that have caught the attention of many over the past years. It is a phenomenon that occurs daily and we often encounter congestion even if there are no clear causes such as traffic accidents, work areas, and weather, which affect traffic. With economic development, this study noticed an increment in the number of vehicles on large-scale, thus the traffic congestion is considered as recurring problem which has various bad impacts on society [8, 9].

Nowadays, the main cities are majorly depending on transportation systems, for their economic survival as well as environmental and social sustainability. In addition, the transportation systems have direct effect on all economy sectors and large growth in traffic volume in large cities, which include a high number of vehicle owner. One of the main factors for traffic congestion, see Figure1.1. Now and in the future, there will be a worse situation if it is not planned in the long term [3].

Figure 1.1: The diffrenece between normal and congasted roads.

1.2 WHAT IS REAL-TIME TRACKING

The real time tracking systems are tracking at least one object. Imaging devices include a camera, a frame, and a processor. This program consists of a low-level image capture

program and a tracking algorithm. The system tracks objects based upon the object's color, movement, or shape in the image. The method then calculates the target's most likely location using real-time tracking technology [10]. Objective detecting is significant in any vision-based monitoring system in a real-time [11]. Real time detection as well as tracking related to objects was huge, lively, yet complex and inconclusive computer vision field. Also, the technology developments in cameras affecting real time achievement related to the detection and tracking of objects. With regard to real world object tracking systems, the system should have high robustness for tackling the changing environments with real time constraint, also the limited memory and processing resources [12].

1.2.1 Camera System

The camera system includes a large number of high-resolution cameras, each of the high-resolution cameras are linked to a set of image sensing areas that were configured for the first time to detect motion based on the sensed images, identify a group of image-sensing areas based on movement, and work this linked high-resolution cameras. With the specified set of image sensing regions [13]. Using one of the type of cameras used in highways (CCTV Camera (Closed-circuit television)) as shown in Figure1.2. It is used to monitor traffic flows primarily for traffic management purposes. In the result, the output of camera system will be a video file that used as input for DIP (digital image processing) approach to localize (detect) cars in the main road.

Figure 1.2: The CCVT Camera on the highway.

1.2.2 Vehicle Tracking

Vehicle tracking can be defined as one of the considerable methods of computer vision for detecting traffic jams[14].One of the important components of tracking systems is configuring path or configuration, where we use the motion segmentation step [15]. The process coordinator is creating distinctive process with regard to all identified vehicles. Also, it applies initial parameters with regard to size and position of the car which have been specified via vehicles detector as shown in Figure1.3[16].To determine the side position of the moving vehicle about a visually distinct line extending all the way the vehicle is going. The camera periodically scans the road including the visually marked line. The signals representing consecutive scans are stored and then linked to determine the side movement of the car between the scans [17].

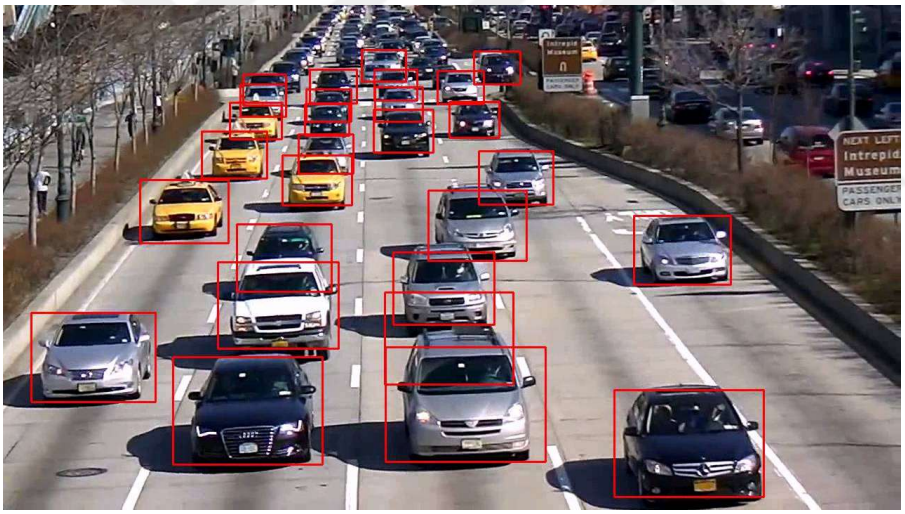

Figure 1.3: Vehicle track using camera.

1.3 PROBLEM STATEMENT

The main problem taken by this study is the problem of highway congestion, which is a real problem that has economic dimensions, psychological and health problems in addition to some risks of traffic accidents. In this study we will answer the main question, which is how to reduce congestion by using modern techniques of machine learning.

1.4 OBJECTIVE AND AIMES

The aim of the presented study is suggesting algorithm for solving traffic congestion and detect it on highways using surveillance cameras. The benefits of this algorithm are used to detect vehicles from the video recorded from the cameras and calculate the traffic congestion rate, which facilitates knowing the areas of congestion and notify the vehicle drivers in order to reduce the speed or take alternative roads. In addition, the proposed algorithm will test and monitor the highways in terms of real time detection to solve traffic congestion.

1.5 STRUCTURE OF THESIS

Chapter one presents a general introduction of the study and explain what is the real time tracking system. Finally, it describes the main problem and objectives.

Chapter Two presents a comprehensive review of literature of the congestion traffic.

Chapter Three presents the used materials and methodology in details.

Chapter Four presenting experimental results as well as the way the researcher evaluate the suggested algorithm.

Chapter Five presenting the study conclusions and suggestion to future works.

2. LITERATURE REVIEW

This chapter will review the most recent method that presented by other researchers and discuss the difference between them.

2.1 TRAFFIC CONGESTION DETECTION METHODS

Traffic road congestions have become a challenge because of the number of increasing vehicles on the road, which has been in line with findings of Baker (2018), where the author indicated that there are approximately 1.3 vehicles on the road. The number was expected to increase to 2 billion by 2040 [18]. Traffic congestion in 21st century, is not just related to developing countries, but is generally seen in developed countries as well. It is characterized by longer trip times and slower speed; henceforth resulting on increased vehicular queuing. Even when countries are experiencing exponential increment in the number of vehicles, also the roads' conditions remain the same, which Roy et al. acknowledged to have led to increased traffic congestions [19]. This calls for an immediate attention to reduce traffic congestion through the use of information technology, where cameras and other technologies can be used to detect traffic jams through the use of certain algorithms.

2.1.1 Surveillance Videos on the Basis of Macro Optical Flow Velocity

Traffic jam is collected information using monitoring videos as shown in Figure 2.1. Traffic flow velocity, which is defined as "the overall optical flow velocity," was directly estimated from video pixel traffic. In addition, the progress of calculation involves 2 stages: Initially, feature points' extraction, satisfying 3 conditions: main direction consistency, strong motion and angle. Second, calculation of optical flow vectors that are related to the feature points utilizing LK (Lucas–Kanade approach). Furthermore, the average value was specified as total optical flow speed. Based on continuous time nature as well as other properties, the traffic congestion was directly and automatically detected [18].

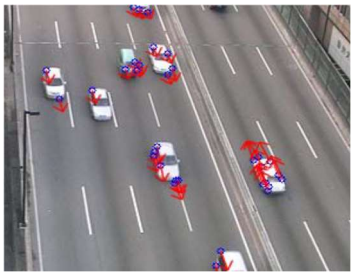

Figure 2.1: Optical flow vectors of key features points [18].

2.1.2 Image Processing

The method of congestion identification is on the basis of image processing, while the velocity of the propagation that has been related to congestion wave was acquired via transforming 3-D spatial temporal direction re-construction data (with velocity in z axis) to spatiotemporal velocity image as well as carrying out 2-D self-correlation. However, congestion template was developed for correlating with the original image for achieving congestion identification as well as temporal positioning, as can be seen in the Figure 2.2, such theme is comparing the impact of differences in on-board unit (OBU) sample rate on the results of congestion detection, the intensity estimation and congestion identification might be performing excellently as OBO sample rate isn't extremely low. Therefore, the method of self-correlation has high robustness compared to conventional Hough transformation. [20].

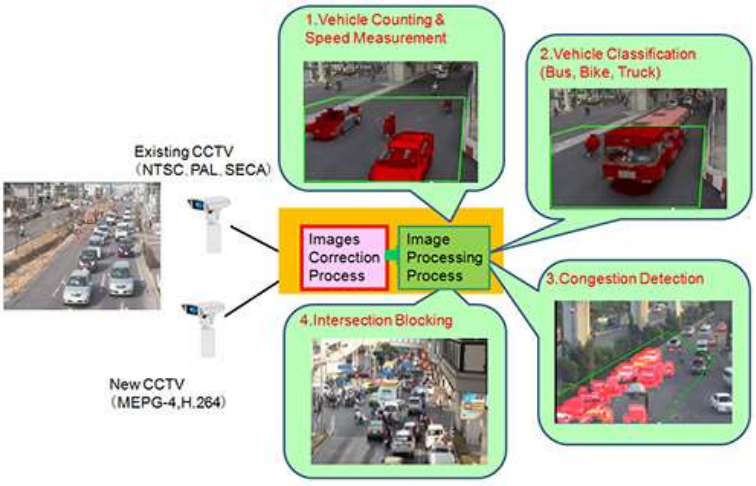

Figure 2.2: Image processing for the traffic management.

2.1.3 YOLO (You Only Look Once)

The YOLO framework was utilized for calculating the traffic flow. Also, there are 3 blocks in each system structure, involving the detector which is creating the vehicles' bounding box, the buffer storing the coordinates of vehicles, and the counter which count the vehicles, as can be seen in Figure 2.3. The proposed system only requires the use of simple distance calculations for the purpose of achieving the vehicle's calculations. Also, by adding checkpoints, the system can mitigate the false detection results, while videos from various angles and locations were utilized for validating and analyzing the suggested system's overall efficiency. The results have indicated that the system is achieving elevated counting accuracy in environment with adequate ambient lighting [21].

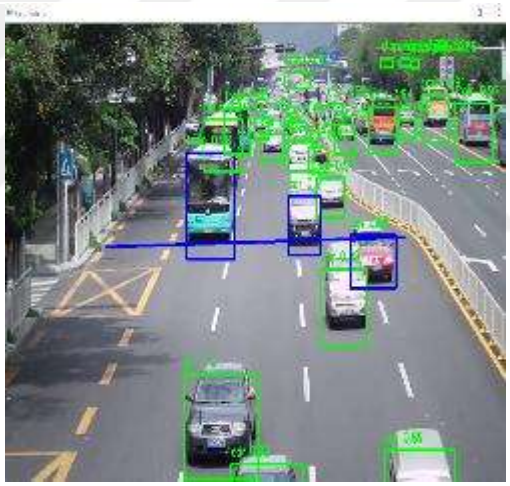

Figure 2.3: Traffic congestion detection using Yolo algorithm.

2.1.4 Optical Remote Sensing Imagery

Figure 2.4 is showing the detection of traffic congestion in the time series which are related to the optical digital camera photos. Also, it is used to derive various parameters of traffic like congestion length, start and end of congestion, average vehicle speed, density of vehicle, or other traffic control applications. In addition, the approach is based on identifying the vehicles in road segment through identifying the changes between 2 images with short time intervals and advanced information like parameters of the road, sizes of vehicles, database of the road, and simple linear movement model which is on the basis of spacings between vehicles—the evaluated velocity properties of experimental data that have

been acquired via airborne optical sensor. Furthermore, 3K camera system is excellently corresponding to reference measurements [22].

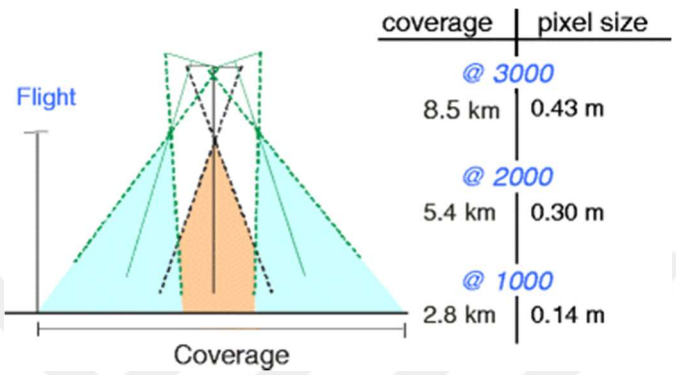

Figure 2.4: Geometry of the image acquisition for the 3K camera system. The angle of tilt which is related to the sidewards looking cameras is approximately 35° [22].

2.1.5 Vehicle-to-Vehicle Communications

An approach utilizing the V2V communications for detecting and characterizing the traffic congestion, as seen in the Figure 2.5, the technique involves and approach for compensating the impact of radio propagation on precise estimation which is related to traffic density and responsible for gradual market introduction regarding the cooperative vehicular communications estimated in large scale highway cases utilizing i-TETRIS simulation platform, also its performance was put to comparison with V2V as well as infrastructure based detection solutions. In addition, the acquired results are showing that they might be effectively and precisely detecting the congestion conditions. Furthermore, it might be precisely characterizing these situations, especially the intensity and length of traffic jam, with no requirement to use infrastructure nodes, while the V2V communications might be representing cost effective and efficient solutions for the authorities of the road management in detecting and characterizing conditions of the road traffic congestion [19].

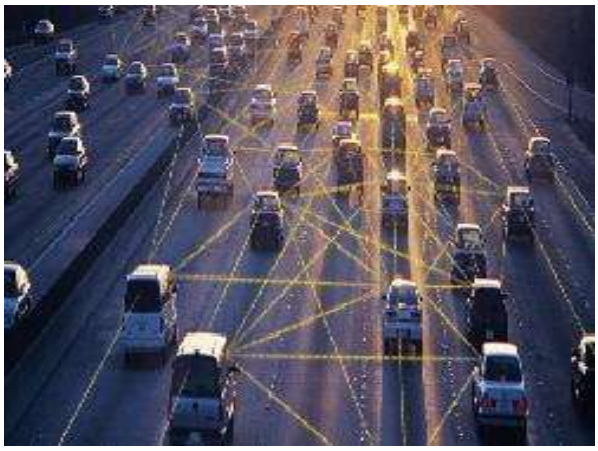

Figure 2.5: V2V communication in road.

2.2 CAR TRACKING METHODS

Car tracking is considered as important research area. The systems of car tracking must have the ability to track sudden variations in the state of streets. Also, the automatic car detection was of high importance in monitoring traffic as well as control access applications making it relevant recently due to the more and more security awareness [23]. Following image preprocessing as well as passing it to monitor the recognition pattern of the video, the objects' coordinates in each one of the frames might be extracted easily. After that, the coordinates related to extracted object might be filtered for obtaining the needed vehicle coordinates. For the purpose of achieving the function of counting vehicles, there is a requirement for determining the relation of vehicles in various tires, whether they are representing the same vehicle. Even though the counting of vehicles might be done utilizing tracking algorithm, short-period related to recognition failure might result in incorrect tracking [21]. Many computer systems are suggested for detecting cars through various features.

2.2.1 Image Processing

The image related to vehicles might be specified as digital image, that was binary data. The image was utilized as primary input, as can be seen in Figure 2.6. The image, when taken from the natural environment, is raw and uncoordinated. So, must process data and extracting the related information from images for effective processing. The constant requirement for extracting data from the images resulted various areas (object recognition, computer vision and image processing) in computing industry. In addition, the digital image processing involves the approaches of digital image processing through a computer, while

the digital image includes limited number of elements, each one of them with certain value and location, such elements are referred to as pixels or image elements, while the image processing includes image representation, compression, as well as many other complex processes which might be achieved on the image data. Furthermore, the operations which come under the image processing have been image enhancements like edge enhancement, noise, sharpening and brightness [23].

Figure 2.6: Vehicle detection using image processing.

2.2.2 Haar-Cascade Classifier

The developed system includes three main steps. First, the initial vehicle targets are created with the Haar-Cascade compiler. Only candidates who pass all stages are classified as positive and rejected candidates at any stage are classified as negative. The advantage is that the majority of the first candidates are actually negative images, which usually cannot pass the first few stages. Consequently, this early rejection greatly reduces the total time of computing.

Second, the target verification is used, which rely depends on a vehicle's lighting feature. As there can be many false positives in the output of images in the first phase of the system, which comes from the limitations of the training set to reduce the false alarm rate, the algorithm does the aforementioned car lighting feature. Regardless of the shape, texture or color of the car, they all share a common feature they all have red lights in the back, assuming that most false positives discovered in the first step do not have such a feature, and

the result can, therefore, be refined results, and in the end, Results are improved by Kalman tracking of objects [24]. As can be seen in Figure 2.7.

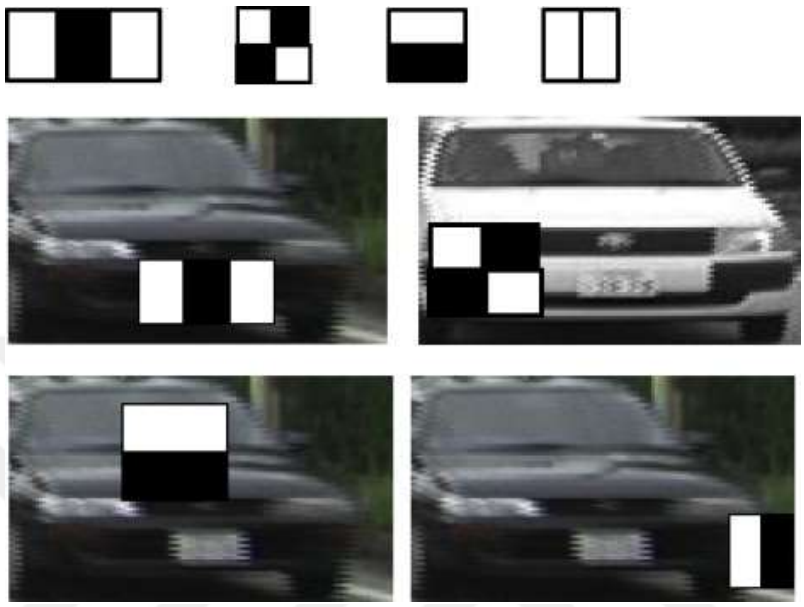

Figure 2.7: Vehicle detection using haar cascade classifier.

2.2.3 YOLO

YOLO framework using the existing detection and classification approach that called (YOLOv2) to improve calculation as well as processing speed, also at identical time effectively for identifying objects in the video recordings. In addition, the sorting algorithm is creating bounding box for each one of the objects which they were trained in and creating a comment describing certain class of objects. YOLO based detection and classification YOLOv2 applies GPU for increasing the speed of operations [25].

YOLO framework, as shown in Figure 2.8 using in calculate traffic flow like the following: The system structure includes 3 sections, a detector creating vehicles' bounding box, a buffer storing vehicle's coordinates, and the counter count the vehicles. The system only requires using simple distance calculation for achieving the vehicle's calculation. Besides, by adding checkpoints, the system can mitigate false detection results. In addition, the videos form various angles and locations were utilized for proving and analyzing the system overall efficiency and analysis, and the results have shown that the system has achieved high environmental accuracy with sufficient ambient lighting [21].

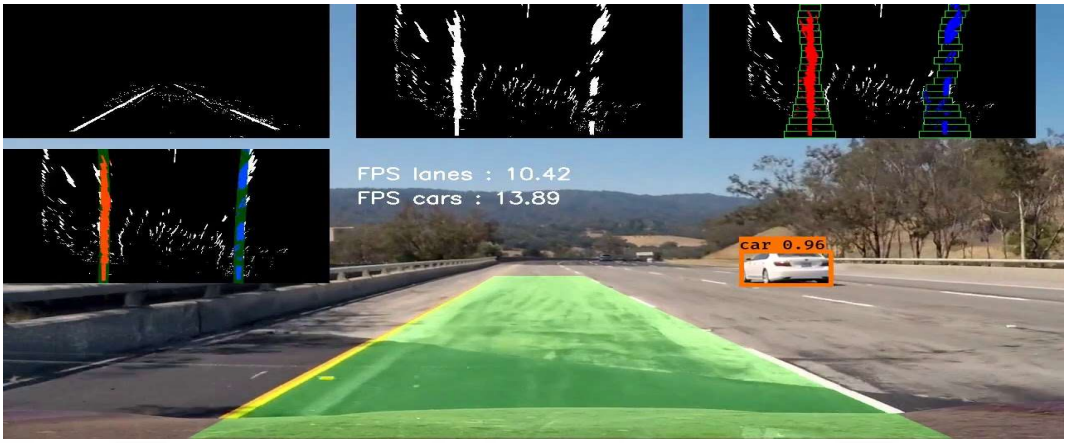

Figure 2.8: Vehicle detection using YOLO.

2.2.4 Contour

Tracking the main object using the contour is of high importance in computer vision as can be seen in Figure 2.9. The problem involves tracing the boundary contour of a moving and deformed object in a series of images. Usually, the problem is solved as follows. First, the object's circumference is obtained in the first frame. Once the approximate contour of the required structure is available on the first image of the sequence. Second, the system automatically defines the lines on subsequent images at the video rate. Third, the object is detected and rounded by bounding box with label. The contour basically is useful in many fields such as traffic recognition, automated monitoring, human-computer interaction, traffic control, and vehicle navigation [26].

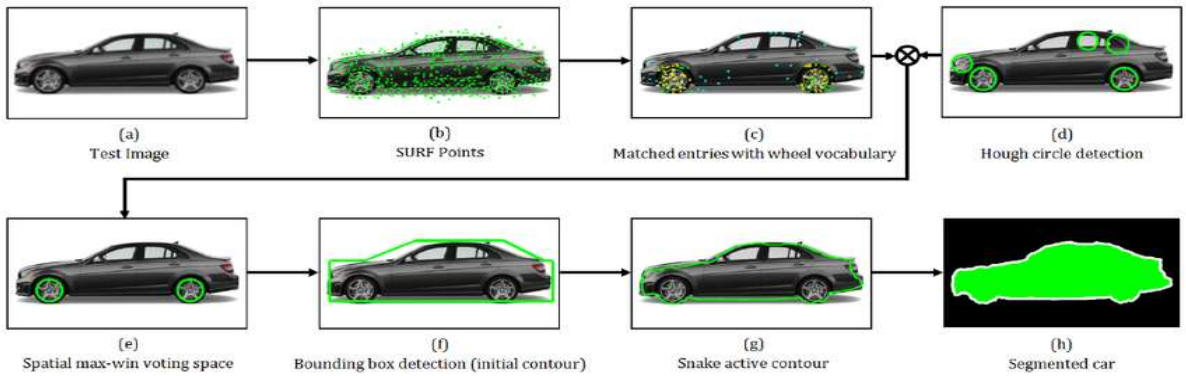

Figure 2.9: Vehicle segmentation and detection process using contour.

2.2.5 Gaussian Mixture Model (GMM)

A new background GMM and shadow elimination approach were utilized for dealing with the abrupt changes of the lighting and vibration of the camera as can be seen in Figure2.10. The Kalman filter performs the tracking of the vehicle for the purpose of enabling the classification by the majority choosing over a number of the sequential frames, and an approach of level set was utilized for refining forefront blob. The extensive experimentations with the real-world data were performed for the evaluation of the execution of the system. The optimal display results of the training of the an SVM with the use if a combination of the silhouette of the vehicle as well as the intensity-based features of the pyramid HOG that have been obtained after the subtraction of the background, the classification of the forefront blobs with the majority vote [27].

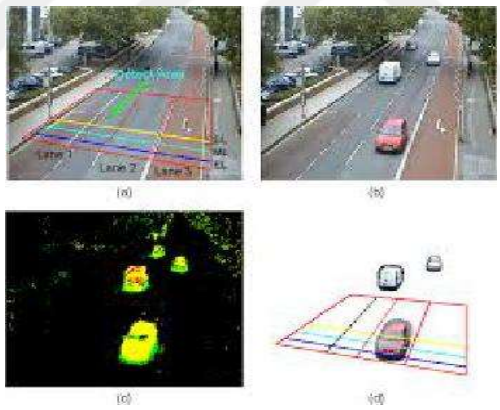

Figure 2.10: Vehicle detection using Gaussian Mixture Model (GMM).

2.2.6 SIFT (Scale-Invariant Feature Transform)

It is a photo descriptor for image-based matching that has been presented with D. Lowe (1999, 2004). It is used in conjunction with related image descriptors for several purposes in the computer-associated vision by matching the point between the various views of a 3D scene and identifying the display-based objects. SIFT descriptor is consistent in rotations, image field metering shifts, translations, strong to moderate shifts, and lighting differences.

Empirically, SIFT descriptor proved quite beneficial in practice for the strong image matching as well as object recognition in the conditions of the real-world [28].

Complete vehicle tracking and detection methods based upon the computer vision. Vehicle detection which has been based upon the SIFT matching. With the SIFT features discovered and matched, the engineering relationships between both images are appreciated. After that, the former image has been aligned with current image in order to identify moving vehicles through the analysis of image of the difference between the 2 alignment images. Vehicles are tracked as well, based upon SIFT feature matching. To reduce time consumption and maintain a greater accuracy of tracking, the candidate vehicle that has been detected in current image has been matched with the sample of the vehicle in tracking set of samples, containing all vehicles that have been detected in preceding images. Additionally, vehicle input and output management are achieved based upon the SIFT feature matched with an effective update approach for tracking sample set [29].

2.2.7 Markov Monte Carlo (MCMC)

MCMC basically referred to as the MCMC Optical Flow (OF-MCMC) is a proposed sampling tracking approach for the tracking of the vehicles. initially, the approach of the optical flow was utilized for getting the moving car's direction in the primary tires that mwy result in solving the issue of changing the volume, more than that, the optic flow approach may get the car's moving speed that substitutes the automatic slope of the second degree of the movement model due to the characteristic change the parameter. Secondly, when calculating whether or not a single particle is accepted, the distance factor is taken under consideration, which may reduce interference of the similar compounds [30]. As shown in Figure2.11.

Figure 2.11: The 1st row is tracking results of OF-MCMC approach. The 2nd row is tracking results of the AR2-MCMC approach [30].

2.3 LITERATURE REVIEW

1. Weiqi proposed in 2011 the inclusion of a VISSIM model for the microscopic traffic simulation program and theoretical basis. Based on intersecting converging transport shapes, complex and common problems presented a comprehensive analysis to investigate the current traffic situation and systematically introduced the single-link signal control timing designs and the theory of the traffic flow. The approach of the research is a simulation of resistance which is carried out with the use of VISSIM program, considering that it is utilized for the real-time detections and meets practical use requirements. The research and conclusion work to enhance the scheme of the traffic design. Practice has shown that the simulation of the small traffic is one of the active ways of solving the traffic issues. simultaneously, applications of the traffic simulation have gotten numerous applications in addition to development of the transportation and

computer technology theories and resulted in a sufficient economic and social benefits [31].

2. In 2011, Kaushik introduced an intelligent traffic jam system which has been referred to as the Traffic Monitor for monitoring and measuring the road traffic jams with the use of an investigation vehicle. The concept of the inspection vehicle recently appeared to collect real-time traffic data. The system has presented a simple platform for the analysis of patterns of traffic and congestion. Traffic Monitor is a fast, inexpensive and easy to monitor traffic jam monitoring and measurement system, combining active RFID (which has been based upon the ISM 802.15.4 and 2.40GHz ISM range) and Global System for Mobile Communications (GSM) technologies. The congestion detection algorithm is based on calculating the speed of vehicles along the way and the average parking time when crossing the road [32].

3. Madhavi. In 2012 a real-time traffic signal system were proposed using morphological edge detection and fuzzy logic. The main problem that has been facing every country is the increasing number of cars worldwide, especially in large urban regions. Hence the need to simulate and improve algorithms of traffic control for the purpose of more sufficiently accommodating such increased demand. One of the ways for overcoming the traffic issues in the big cities is to develop a smart traffic control system that measures road traffic intensity. In this paper, the authors have presented the techniques that solve this traffic problem. Discuss the morphological approach of edge detection for the purpose of monitoring real-time traffic and then preventing the logic. When comparing the two approaches, it has been found that the fuzzy logic is easily implemented from the approach of the morphology due to the fact that the morphology approach is quite long, even in the case where the approach of the edge detection doesn't work sufficiently throughout the night, the edges of some of the compounds will not be doable for the detection due to the darkness at night, but the mysterious logic does not It only calculates the number of vehicles that do not handle edges, but also has given more precise results. If they see the cost factor, the morphological approach is less expensive than obscure due to the fact that the adjustment approach requires only a high-quality camera and does not

feel it is less expensive. Mysterious logic provides the ability to implement the real-life rules that are similar to how humans think, so mysterious logic undoubtedly gives better results. It deals as well with the number of the vehicles achieving better results, but the modification method depends on the traffic density that produces approximate results [33].

4. In 2013, Fahri proposed a solution to the growing traffic jam issue in the urban regions pricing systems were used. In this paper, they have suggested a fundamentally various road pricing plan for the prevention and the reduction of the traffic jam in major cities. Contrary to allocating a small space to the costs of congestion in the city, they suggested using the road pricing system for the whole city. Which is why, the system of road pricing has the ability of controlling traffic flow in the whole city's traffic network. Moreover, the road rates are dynamically adjusted based on the instant traffic density levels for every one of the city roads to quickly and effectively control traffic flow and prevent traffic jam. In addition to that, they changed road rates according to previous road usage statistics by predicting potential congestion. Simulation results for their road pricing algorithm confirm that traffic jam has been prevented across the whole traffic network, and that traffic density has gone smoothly [34].

5. Konstantinos in 2013 focused in this research on surveying drone-based systems to monitor and manage traffic. Even though there have been numerous researches on this topic, UAVs (unmanned aerial vehicle) have proved to be a viable and faster substitute in monitoring and managing real-time traffic, providing the optimal solution to the issue, it can be argued that UAV may be quite useful for traffic control. Maneuverability and wireless network communications are among the most important points which have made unmanned aerial vehicles more beneficial, compared to other approaches that are being utilized presently. The first makes it able to track vehicles on ground, whereas the One second allows the transfer and reception of instructions, information, images and video to the Earth's base. The research focuses on the sensor types that may be utilized on board (vision, radar, or hybrid) as well as on the processing type (i.e. on-board or off-board) of data in the sensors of the vision [35].

6. In 2014, milos and Veselin demonstrated an algorithm VANETs (Vehicular ad hoc networks) designed for the purpose of enabling each one of the vehicles in the network to detected and determine the traffic jam level in a fully distributed manner, regardless of any supportive model and additional information like the traffic data from the local police. Based upon traffic jam notes for every one of the vehicles, and by adapting the broadcast interval, traffic information can be published to other vehicles. The algorithm senses as well each one of the vehicles to the level of congestion in ways that are spatially segregated from its current location on a number of the streets. Its durability maintains overall vehicle knowledge of congestion, in spite of the short-term variations in the movement of the vehicle. Due to the fact that the determination of the amount of congestion depends upon each one of the vehicles, the algorithm can operate even in the case where only 10% of the vehicles are enabled in network. Adaptive data collection and transference have been utilized in order to secure that vehicles don't transmit excessive traffic information. Vein simulations are performed based on OMNeT ++(Objective Modular Network Testbed in C++) network simulator and SUMO (Simulation of Urban MObility) vehicle traffic simulator [36].

7. In 2014, Ahmed introduced a system for the detection of the crowding in real-time with no prior knowledge nor human supervision. The system designed calculates vehicles on the road through the detection and adjustment of the vehicle's taillights from an image captured in real-time, while the system takes less time and may be implemented in any highway hypothesis. Experimental results have explained that a system has a high accuracy of the detection of up to 99% [37].

8. Real-Time Traffic Detection from Twitter Stream Analyses has been proposed by Eleanor in 2015. A real-time monitoring system for the traffic congestion detects of Twitter stream analyses. The system gets tweets from Twitter, processes tweets, by using text mining techniques. As associated with the traffic congestion or not. This system has been used on the Italian way network, which allowed for the detection of traffic congestion about in real-time. Used the SVMs as a ranking type, and accomplished a

precision rate of 95.75% through solving the problem (traffic versus non-traffic tweets) [38].

9. Lhtisham suggest in 2015 a smart E-Traffic management system, utilizing the image processing technologies in addition to an intelligent traffic control algorithm. Traffic recognition has been accomplished with the use of Vehicle Recognition Progressive workbook with the use of the OpenCV and Visual Studio C&C ++. The workbook has been trained in 700 positive samples and 1,140 negative ines. The results have explained that the detection accuracy of vehicles has about 93%. The number of vehicles on all intersection methods is utilized for the estimation of the traffic. Traffic accumulation has been then evaded or increased through passing extracted data to the algorithm of the traffic control. Shows control algorithm about 86% enhancement on a static lag controller in worst-case scenarios [39].

10. Alan and others proposed in 2016 a smart traffic system that has been referred to as the CHIMERA, improving the general spatial use of the road network as well as reducing the average traveling expenses by car by preventing the cars from getting into traffic. Simulation results display that the proposal has been more effective in predicting congestion and has the ability to correctly redirect vehicles, resulting in an appropriate load balance for traffic. Considered CHIMERA, a new solution to discover and monitor urban traffic congestion. The suggested solution has the aim of reducing the travelling time, reducing the distance and the time of the travel. Simulation results have shown the efficiency of the CHIMERA. Compared with the original vehicle navigation tracker program, CHIMERA reduces average flight time by 31%, average stop time at 70% and average travel distance by 8%. [8].

11. Shaif in 2017 presented this paper describing vehicle detection technology wiich may be utilized for the traffic control systems. The intelligent traffic monitoring system which is supplied by the electronic devices operates through the communication with the moving vehicles about traffic conditions, observing regulations and rules, and avoiding collisions amongst the cars. So the first step in that process is auto-detection. This system used the

Haar-like features in order to detect the car, that is, in general, utilized for the face detection. Haar's feature-based serial rating devices is an effective way to detect objects first presented by Viola and Jones. It is a machine-based technology that utilizes a combination of the negative and positive images for training purposes. Results have shown that this approach has been extremely effective and fast in the detection of the cars in the CCTV footage in real-time [40].

12. Chan-tong et al in 2017 proposed an economic system to detect real-time traffic congestion with the use of the on-line images that have been provided by local government. The proposed system consists mainly of the discovery of the vehicles which utilize the images on the Internet and estimating traffic congestion based upon estimated number of the vehicles. To detect vehicles, study a way to use signs on the road and try Haar-like features, which can be used for the detection of the vehicles using images on the Internet from various camera sites. For traffic congestion estimation, a coefficient of image correlation for the successive images has been utilized, in addition to a threshold for the number of the detected vehicles. Two different congestion levels have been taken under consideration, those are NORMAL and CONGESTED, even though the number of the congestion level may be extended easily. The experimental results have shown that the suggested system has the ability of the correct estimation of the real-time traffic congestion at low costs. In comparison with the conventional traffic congestion estimated systems, this system has provided a more cost-effective solution with potential commercial applications of the tourists and residents in Macau [41].

13. Revanth Ayala, introduced in 2018 the presented project aims at the development of a solution for the real-time traffic jam detection on the roads. The solution has captured images from the live broadcast of the traffic cameras that are located in a variety of the locations and operates a deep learning algorithm for the detection of whether or not that image is showing the traffic congestion. The use of a combination of those images and continuous verifications, the app performs the determination of when crowd starts and ends. For the purposes of the analyses, the solution has also recorded detected incident videos [42].

14. In 2018 Aleksa presented evidence of the latest YOLOv3 algorithm to discover traffic participants. They trained the grid on 5 object classes (which are trucks, cars, traffic lights, pedestrians, and lights) and has shown showed the efficiency of this approach in various driving conditions (such as clouds, bright sky, fog, snow and night). In this paper, the app introduces the YOLOv3 algorithm for real-time detection of traffic participants. NN weights have been initialized with the use of a pre-trained COCO data-set. The NN has been trained in the Berkeley Deep Drive data-set for specializing in discovering five categories of traffic participants. Wrong discoveries were discovered in a custom data-set which is consisting of images which represent a variety of the traffic cases in the area of Novi Sad [43].

15. Pranamesh et al in 2018 presented a study for the detection of the traffic congestion from camera images with the use of the deep CNNs. Where it is proven recent enhancements in the machine visibility algorithms have resulted in CCTV cameras as one of the significant sources of the data for the determination of the traffic congestions. This research has utilized two different approaches of the deep learning. YOLO and the D-CNN, for the detection of the traffic congestion from the images of the camera. SVM, which is a simple algorithm, has been utilized as a comparison for the determination of enhancements that have been obtained with the use of the deep learning algorithms. Ownership data from nearby radar sensors have been utilized in order to distinguish crowded images in the data set and to train models. YOLO and DCCN (Data Communication and Computer Networking) achieved an accuracy of 91.50% and 90.20%, respectively, while SVMs accuracy has been 85.20%. Distinctive curves have been utilized to the receiver to determine the sensibility of the models about different camera configurations, lighting conditions, and so on [44].

16. In 2019, Chan-Tong Lam proposed an inexpensive, standardized system for real-time traffic state detection with the use of the free images that are available online. Due to the fact that the vehicle detection quality from those on-line images may vary greatly, based upon the camera locations and weather conditions, it has been suggested using a general

object detection system with deep learning, such as YOLO v3, for the detection of the real-time traffic status of those images. For the purpose of the determination of traffic status with the use of the projected surrounding squares and the number of vehicles that have been obtained from YOLO-v3 detector, it has been suggested using cross-union (IOU) in order to combine the surrounding squares of multiple objects projected on the images captured in different images times, called mIOU. The corresponding estimated number of vehicles so as to evaluate the multiple traffic condition Levels. The experimental results found that the resolution and recall rate were capable of achieving 86% & 87%, respectively, according to the cameras' locations. It has also been shown that the suggested system utilizing mIOU and YOLO-v3 may be utilized to estimate the traffic state of low-resolution images, without requiring any prior training of networks for various settings. The suggested traffic detection technology applies many different applications [45].

3. METHODOLOGY

This chapter provides a summary of detection and classification methods. The study explains how to recognize vehicles and detect traffic jams in real-time. This work aims to solve traffic problems, rely on tracking vehicles, and calculating their number, using classifications that depend on their work in surveillance cameras to detect vehicles, thereby determining traffic congestion in real-time see Figure 3.1.

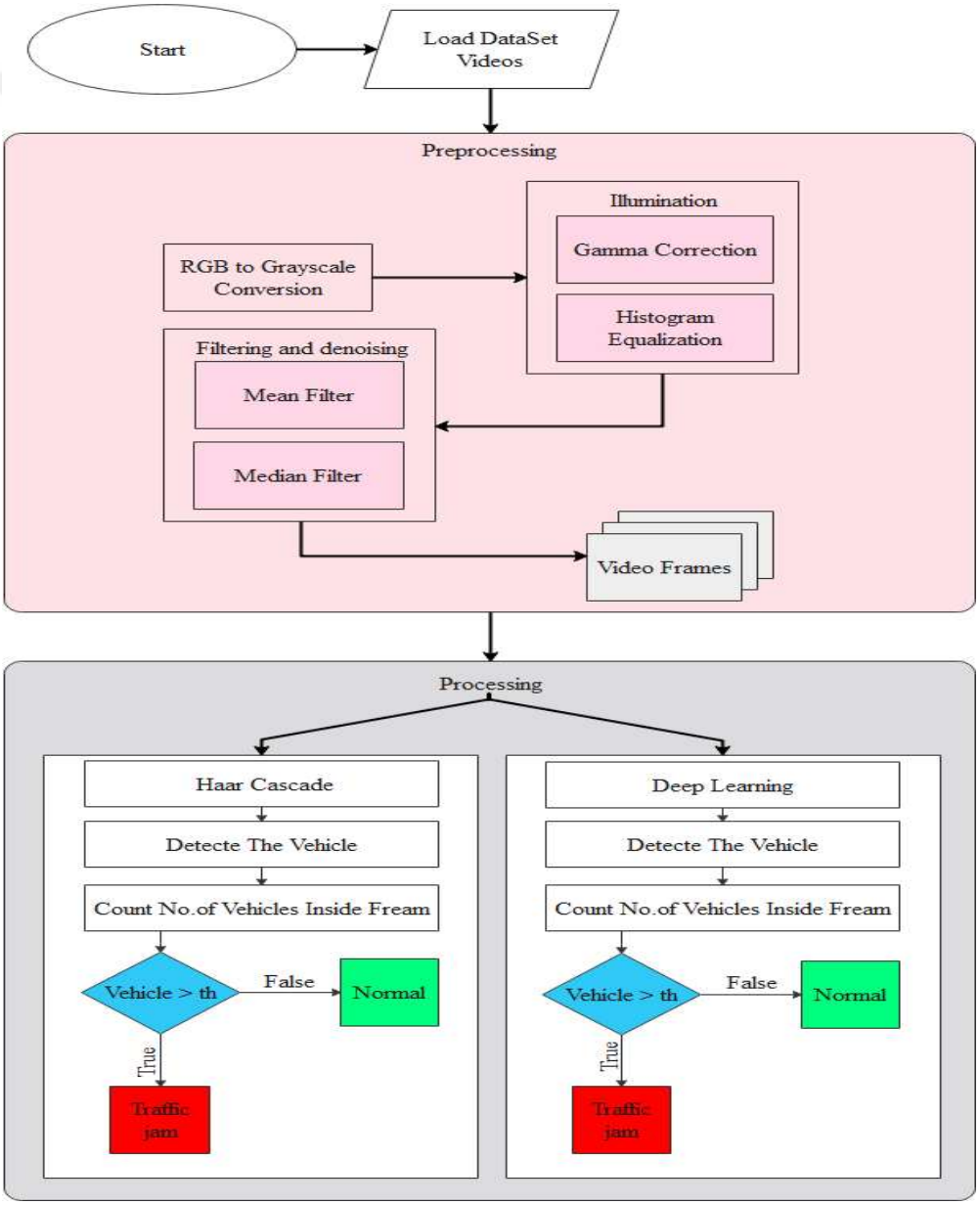

Figure 3.1: A general flow chart describing the main stages of the method used.

3.1 PREPROCESSING

In the present section, we will describe details of performing video pre-processing before performing the above-mentioned categorizations.

3.1.1 Histogram

Graphic frequency is an approximate description of numerical data distribution. It has been first submitted by Carl Pearson [46]. In order to create a graph, the 1st step is to "bin" (or "bucket") the range of values – which means, dividing the whole value range to a series of the time slots - and then compute the number of the values which fall into every one of the time intervals. Boxes have been usually defined as consecutive and non-interlaced variable dividers. Boxes (separators) should be adjacent, and usually (but not necessarily) of equal size.

The histogram of the image is a type from a histogram which works as a graphical description of tonal pattern in the digital image [47]. It draws the number from pixels for every one of the values. Through viewing the image histogram, we will judge the complete tonal distribution.

- The horizontal histogram axis describes tonal changes, whereas vertical one describes the total amount of the pixels [46].
- The left party of horizontal axis describes dark regions, the middle describes the values of the middle degree, and the right side describes the light regions. The vertical axis describes the area size (the entire number from pixels) captured in every one of those regions.
- Consequently, the graph of a very dark image will be having most from the data points on the graph's center and left.
- On the contrary, the graph of a very bright image with some dark regions and/or shadows will be having the majority of the data points on right and center of the graph.

When pre-processing images, we have created an image histogram which is edited. The histogram will plot the number of the image pixels (i.e. the vertical axis) with a certain tonal value or brightness (i.e. the horizontal axes). The digital editor algorithms permit users to have the ability of the visual adjustment of the value of brightness of every one of the pixels and dynamically show results in the case of making the adjustments [47]. The equation of the graph is one of the common examples of those algorithms. Which is why, enhancements in the brightness and disparity of the image may be obtained.

3.1.2 RGB to Grayscale Conversion

Converting colored image into grayscale, which is used for different purposes that include better bandwidth, better storage use, and post-processing as explain in Figure 3.2. Every pixel in the color image is tripled (R, G, B) as intensively as red, green, and blue. This work are used three ways to select a single value mainly Average, Weighted or luminosity, and lightness. Different color models are used for many applications for example, computer vision, image processing, and computer graphics.

Figure 3.2: Convert colored image mode into grayscale mode.

This work are used the Gray Weighted method to retrieve content-based images except that it takes a gray intensity image to restore the image. The grayscale image produces different shades of eight-bit pixels, ranging between 0 and 255 in contrast to the colored image. The color image separately sets eight-bit values red, green and blue levels. Likewise, every level has a distinct value of 0 to 256. The benefit of the use of a gray density image simplifies more information compared to the 3D color images and consumes less memory. Basic

information which is associated with the grayscale images like edges, regions, points, is not lost [48].

The formula for the conversion of the RGB color image into grayscale has been represented in the following equations:

$$WG = ((0.3\mathrm{R}) + (0.59\mathrm{G}) + (0.11\mathrm{B})) \tag{3.1}$$

Where WG is weighted gray that converted from the colored pixel, R represents red pixel, G represents green pixel, and B represents blue pixel [48].

$$AG = (R + G + B)/3 \tag{3.2}$$

Where AG is average gray that converted from the colored pixel, R represents red pixel, G represents green pixel, and B represents blue pixel [49].

3.1.3 Illumination

The first approach is to consider a variation in the mean value of the pixel of an image in the case of the variation in the lighting of the scene. The capability of devices in compensating for that is varied greatly between different camera models, lenses, and lighting. In the case where the device doesn't compensate for that, the mean image grey level moves to one end, either 0 or 255, according to the decrease or increase of the lighting. In either case, the image is noisier (arbitrary values are added to the grey level) and the amount of the contrast will decrease (smaller standard deviation of pixel values). The two conditions cause the detail in the image to be lost [50].

The second very important factor to consider is the ability of the camera to correctly encode high-contrast scenes. For example, the image's left side can show the sunlit ground simultaneously that the right side of image appears as a shadow of the wall. Only very good devices will be able to make up for this. If the compensation is poor, the image will end up showing all white on the left or all black on the right, making it very difficult to distinguish the objects for a portion of the image [50].

- Gamma correction

The gamma correction can be defined as a technology which is utilized to recompense for nonlinear width properties of the project. The Gamma explains the relation between the color value and its illumination on a given image. For the images that have been described within the RGB color extent for appearing correctly visually, they must generate a directly proportional output brightness (linearly linked) to the value of the input color.

This correction is accomplished through setting input values through a function of correction, which has been specially designed for the properties of projector. Often the function is performed using a lookup table of appointment and is typically used for a separate table for every component of RGB color components. Therefore, video cameras typically perform a non-linear illumination intensity layout of the output value, which provides a more consistent cognitive scale with finer increases of dark values. Which has been referred to as gamma feature as shown by the transfer function:

$$V = C_1 \Phi^{\gamma} + c2 \tag{3.3}$$

Here, Φ represents luminous flux (i.e. standard) and V represents the signal's voltage, c1 represents the sensitivity of the camera and c2 represents the displacement value. Fig. 3.2 shows distinct transfer patterns for 3 values of gamma, γ = (0.5, 0.75, and 2.20). Those values have been typical for the ones that have been utilized in image processing. Figure 3.3 illustrates a number of the corrected image examples with different gamma level values [50]

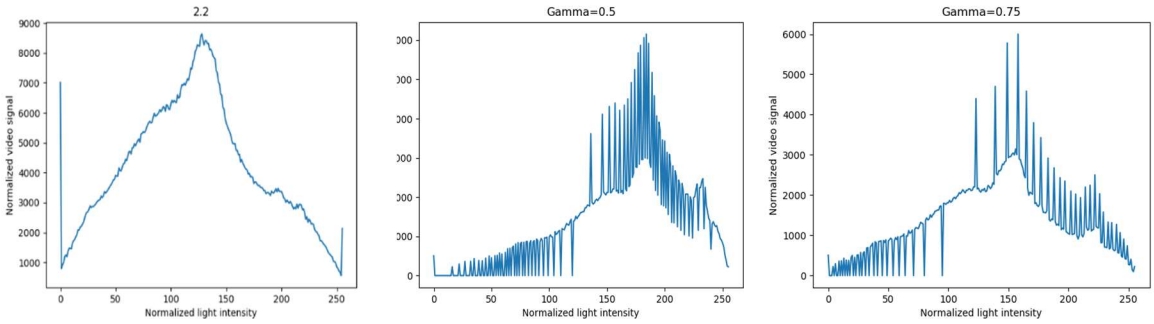

Figure 3.3: Explains histogram of Gamma curves for γ= (0.5, 0.75, 2.2).

The gamma value width is somewhat dependent on the system's exact properties, however, the nominal value was 2.2. The story is slightly more complex, due to the fact that there is a personal aspect of the human perceptions of the brightness (in other words, the lightness), which is affected by viewing environment. Luminous flux is often utilized for displaying video on a dark background. In order to implement an accurate subjective response in that situation, video signals are usually pre-compensated, and the luminous flux is treated as if it had a 2.20 gamma value. Which is why, the known value of gamma 2.2 has an assumed built-in dim view environment [Poy-98]. SRGB (Standard RGB) stands for the color values in the approximate gamma space 2.2 [51]. As shown in the Figure 3.4 with a comparison of the difference in the gamma value.

Figure 3.4: Demonstrate of gamma correction.

- Equalization of the Histogram

It is one of the methods of image processing to set the contrast with the use of an image Histogram. This method is the best way for improving an image. It provides a more sufficient image quality with no loss of any information as shown in Figure 3.5.

Figure 3.5: Examples of Histogram Equalization.

This approach often results in the increase of global contrast of numerous images, particularly in the case where usable data in an image has been represented with the values of the close contrast. With that adjustment, density may be distributed better on graph. Which allows regions with low local contrasts be capable of gaining higher contrast, as shown in Figure 3.6. Histogram equalizer effectively accomplish that method through the spreading of most common density values [52].

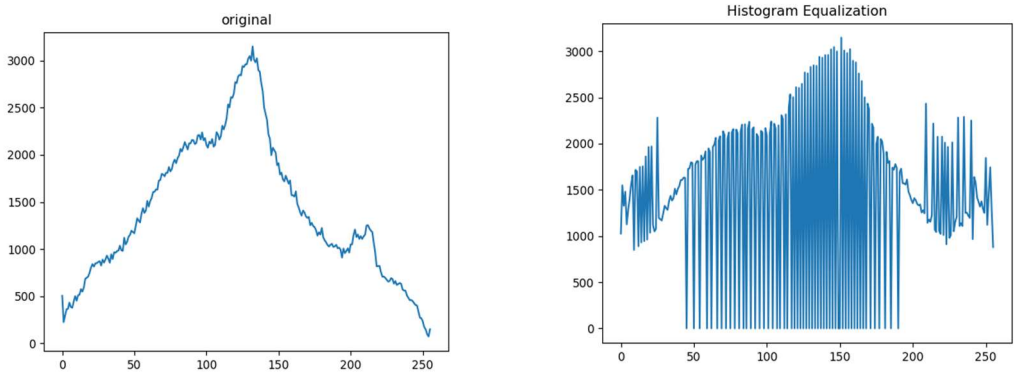

Figure 3.6: Explin the Histogram of Histogram Equalization.

Assume that $\boldsymbol{f}$ is a certain image that has been represented as a matrix of m_r x m_c of integer intensities of the pixels that range between **0** and $\mathbf{L} - 1$. $\mathbf{L}$ represents the number of the possible values of intensity, of ten 256. Assume that $\mathcal{P}$ denotes normalized histogram of $\boldsymbol{f}$ with a bin for every one of the possible intensities. Therefore:

$$\mathcal{P}n = \frac{number\ of\ pixels\ with\ intensity\ n}{total\ number\ of\ pixels} \quad n = 0,1 \dots \dots \mathcal{L} - 1. \tag{3.4}$$

Histogram equalized image g is defined b

$$\mathfrak{g}_{i,g} = floor((L-1\sum_{n=0}^{\mathfrak{f}_{i,j}} \mathcal{P}_n), \tag{3.5}$$

Where floor () sill round down to nearest value of the integer. Which can be considered equivalent to the transformation of pixel intensity values,K, off by function

$$T(K) = floor((L-1\sum_{n=0}^{k} \mathcal{P}_n). \tag{3.6}$$

The motivations behind such transformation has come from considering f & g intensity value as random continual variables X, Y on [0, L−1] with Y calculated from:

$$Y = T(X) = (L-1)\int_0^X px(x)dx\,, \tag{3.7}$$

Here, px represents probability density function of f .T represents cumulative distributive function of X multiplied by $(L-1)$. Assuming for the simplicity that it is invertible and differentiable. It may be shown then that T defined by $T(X)$ has been distributed uniformly on $[0, L-1]$, considering that $P_Y(y) = \frac{1}{L-1}$ in the Histogram equalization [53].

3.1.4 Filtering and Denoising

Conventional noise cancellation strategies have been based upon the theory that the images present a low pass property whereas the noise is covering a wide range of frequency values. It is possible to reduce noise energy through the processing of the image data with a low pass filter which ignores only optical signal's frequency range. More sufficient outcomes may be resulted through creating a matching filter as soon after filter strategy has modeled the original signal. None-the-less, these processes of filtering would have the ability of altering the features that have been calculated on this image; which is why, objects would not be found any longer, or errors of detection could be more common. Numerous of the approaches of object detection have been based upon the Histograms of Oriented Gradients (HOG), even though additional validations are ongoing with the use of various approaches of object detection. The experimental data have shown that the strategy of the de-noising may result in the improvement of performance of the algorithm's correct detection up to 30% at high levels of noise [54].

Filters' success has been based upon 2 fundamental characteristics: protection of the edges and sufficient attenuation of the noise with a robustness against the impulsive noise. Preserving the edges is important in the area of image processing because of the visual perception nature. None-the-less, the filtering is also removing extremely fine details and in some cases results in changing the structure of the signal. The fundamental reason is that this filter only utilizes information of sort order for input data within the window of the filter, and ignores the original time-order information. For the purpose of avoiding the damages to the "good" image pixels the scheme of switching has been presented. Therefore, it can serve as a preprocessing step to robustify many tracking-by-detection algorithms [55].

One of the most widespread operations of filtering is to smooth noise. The main principle of noise smoothing the utilization of the theorem of central limit for the reduction of noise variance. Rather than that, the average of the adjacent pixels is found. For the purpose of avoiding the output value from differing from input in image's uniform regions, a filter of noise smoothing is required.

$$\sum_{i,j=W} W[i,j] = 1 \tag{3.8}$$

Where W represents weights for the sake of this being possible, the weights' array $W_{i,j}$ has to be expressible as a product of the weights of columns and rows [56].

The majority of the images have the energy concentrated in lower frequency values, with high frequency values including the information which is associated with the edge and fine details (in which the values of the pixels change abruptly). Random (i.e. white) noise is of a uniform distribution of the frequency. Which is why, the low pass filter removes the majority of the noise at the same time as having the minimum influence upon the energy content of the image. Unfortunately, in attenuating the noise, the high-frequency range the image is reduced as well, which results in the blur [56].

- Mean Filtering

Filtering means an intuitive, simple and easily implemented approach of image smoothing, in other words, which reduces the level of the contrast between one pixel and beyond. It is usually utilized for the reduction of the noise in the pictures. As shown in Figure 3.7.

Figure 3.7: Remove noise from the image by the mean filtering.

The concept of mean filter is to fundamentally exchange every one of the pixel values in the image with a mean value (i.e. average) for its neighbours, which includes itself. Which has an impact of the elimination of pixel values whose perimeter does not represent. Average

filtration is often viewed as a torsion filter. Similar to other convolutions, it depends on the nucleus, representing the neighborhood's size and shape from which samples will be taken in the case of the calculation of the mean value. Usually, a square 3x3 core is utilized, as can be seen in Figure 3.8, even though larger cores (like 5x5 squares) may be utilized for more severe smoothing. (It should be noted that a small nucleus may be applied several times for the purpose of obtaining a similar impact but not similar to a single pass with large nucleus) [57].

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix} \qquad \begin{bmatrix}1 & 1 & \cdots 1\\ 1 & 1 & \cdots 1\\ \vdots & \vdots & \ddots \vdots\\ 1 & 1 & \cdots 1\end{bmatrix}$$

(a) (b)

Figure 3.8: (a) 3x3 mean filter; (b) general n x n mean filter.

- Median filter

The median filter can be defined as a non-linear signal processing approach which is beneficial for the suppression of noise. It has been proposed as one of the tools in the analysis of time series in 1971 by Tukey and has, after that, becomes useful as well in the image processing. The median filter is carried out through letting a window move over image points (i.e. sequence) and substituting the window center value with median from original values in the window as can be seen in Figure 3.9. Which results in an output image (i.e. sequence) that often is smoother compared to the original image [58].

Figure 3.9: Remove noise from the image by Media filter.

Taking under consideration a group of the random variables X=(X1, X2... XN), order statistics X (1) ≤ X (2) ≤⋯≤X (N) represents random variables, which are characterized by the sorting of Xi values in an ascending manner. The value of the median is then represented in the form of the equation below:

$$\text{median}(X) = \begin{cases} \text{X (K + 1)} = \text{X (m)}, \text{for N} = 2\text{K} + 1 \\ 1/2(\text{X (K)} + \text{X (K + 1)}), \text{for N} = 2\text{K}, \end{cases} \quad (3.9)$$

(2K+ 1) represents the rank of the median. The median has been viewed as a robust estimator of the distribution location parameter and discovered several applications in the de-noising and smoothing, in particular for the signals that have been contaminated through the impulsive noise [59].

3.1.5 Video Framing

That is the basis for video work. Whether it is a digital movie or an old-school movie, the video is a series of static images that, when viewed in sequence at a certain speed, give the appearance of movement. The frame rate is the speed at which these images are displayed, is usually expressed as "frames per second" or FPS. Each image represents a frame, so if a video is captured and played at 24 FPS, every second of the video displays 24 distinct still images.

The frame rate very affects the style and experiment of the video experience. Different frame rates result in different viewing expertise, and selecting a frame rate oftentimes means choosing between things like how real video is, whether it is planning to use technologies like slow movement or movement blur effects.

For example, videos are usually exposed at 24 fps, for this frame rate is related to how to see the system and creates a highly cinematic look, as shown in Figure 3.10. Live video or video that has a lot of motion, such as a sporting event or video game recording, often has a higher frame rate, with lots happening at one time and a high frame rate to keep the motion smooth and clear details [60].

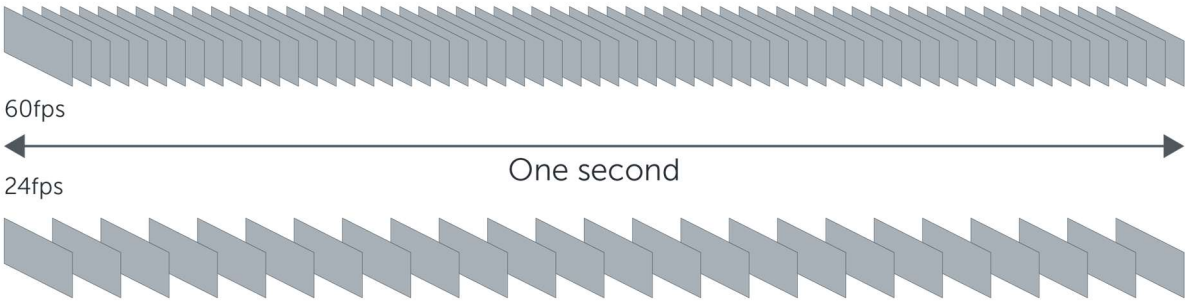

Figure 3.10: Explains video frame in one second.

There isn't any better rate of the frame. As indicated earlier, various frame rates provide various results, which is why, choosing the best method means using the option which optimally suits what one is attempting to create.

3.2 CLASSIFIERS

In this part, the project has presented the classifications that are worked and compared between them. The best qualified method has been chosen. This work has two types of classifications which is haar cascade and deep learning.

3.2.1 Haar Cascade

Haar's feature-based serial ranking devices are an effective way to reveal things that Paul Viola and Michael Jones suggested in there, "Haar cascade" in 2001 [61].

In this stage, vehicle detection has been found. Initially, the algorithm requires many positive (i.e. vehicle) images and negative (i.e. buildings and street) images for the training of workbook. Then it is required extracting features from it. Which is why, Haar features have been illustrated in Figure 3.11. They are similar to our convolutional nucleus. Every one of the features has been obtained through the subtraction of total pixels under white rectangle from total pixels under black rectangle.

It is a machine-based procedure in which the cascading function has been trained in many negative and positive images. It is utilized afterwards for the detection of the objects in different images.

- **Positive Images** - These images contains selective images which the algorithm has chosen to identify.
- **Negative images** – images that contain different features and the algorithm has excluded from other images, do not contain the object.

All preprocessing is applied to all positive and negative images. Before starting to apply the Haar features to the images for each feature, define the best threshold for classifying vehicles as positive and negative. Classification errors or errors are expected during training, determine the features with the lowest parameter of errors, and accurately classify the features of vehicles and non-vehicles.

The Haar cascade has four stages:

1-Haar-Like Feature

The concept of the Haar-like features has been based upon disclosing encryption features of some information considering the class that was to be discovered. They are neighboring rectangles in a specific position of the image. Figure 3.11 shown Haar-like feature types based upon the number of the neighboring rectangles. There are 3 condiment-like feature types. The first one is edge features, which are shown in Figure 3.11 by (a). The second type is line features (Figure 3.11 (b)). The last type of feature is the four rectangle features (Figure. 3.11 (c)) [62].

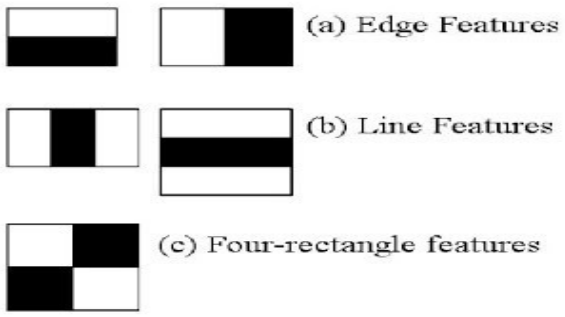

Figure 3.11: Rectangular feature [62].

The concept behind the Haar-like detection approach is simple. It is based upon precept of calculating the distinction between the total of black and white pixels. The fundamental

feature of the approach is the quick summation calculation with the use of the combined image. It has been referred to as the Haar-like due to the fact that it is based upon same Haar wavelet principle [62].

2-Integral image

The rectangle features may be quickly calculated with the use of an average image representation which is referred to as merged image [63]. An integrated image includes a representation of the small units of a specific image as can be seen in Figure3.12.

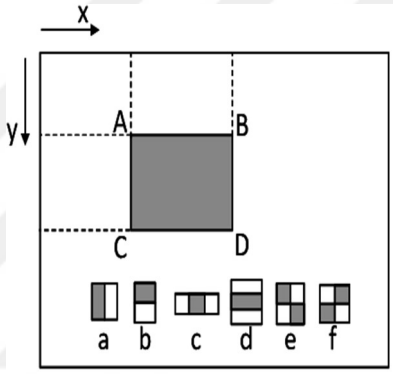

Figure3.12: Sample about that integral image and Haar cascade rectangle features (a-f) [64].

The integral image, which is referred to as the summed-area table as well, is a method for the quick and efficient computation of the summation of the values in the rectangle sub-set of the grid. It has been initially presented by Crow to the area of the computer graphics to be used in the mipmaps. Viola and Jones have used integral image for fast counting from the Haar-like features, as it will be detailed below [64]. The integral image has been composed as:

$$ii(x,y) = \sum_{x' \le x, y' \le y} i(x', y'), \tag{3.10}$$

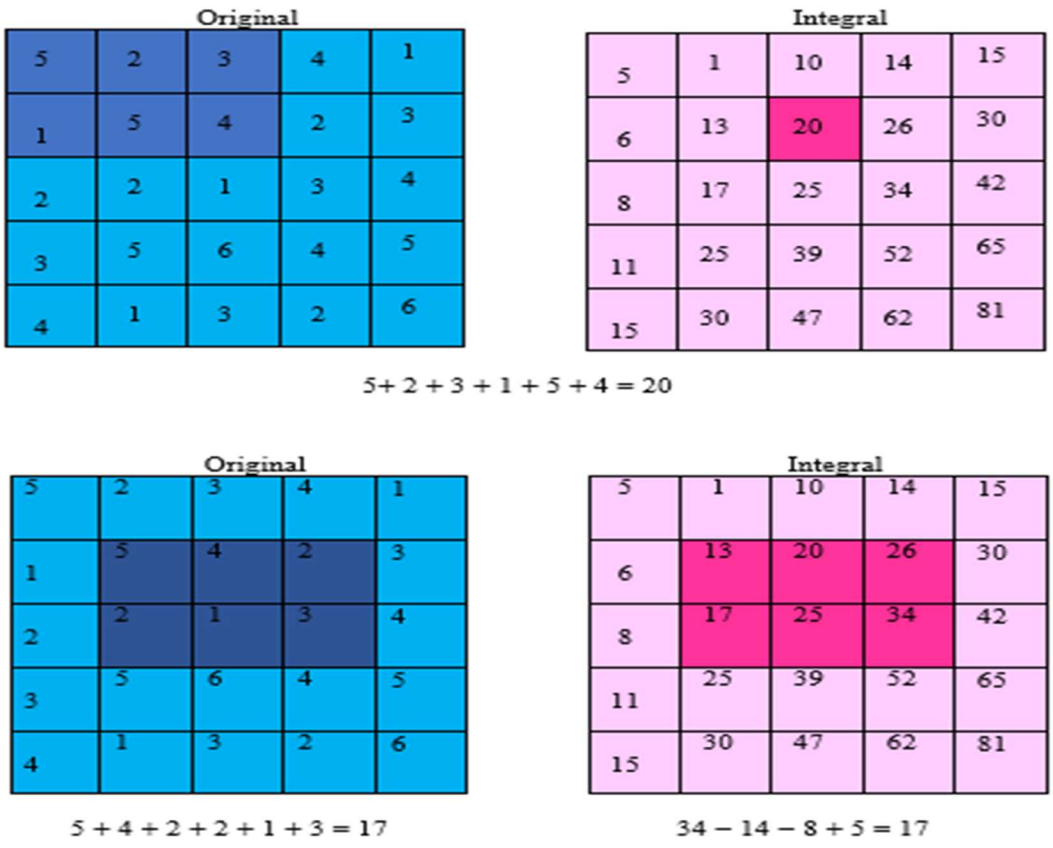

Figure 3.13: Integral image: illustration example.

The use of integral image for computing the summation of any rectangular region is of high efficiency, as can be seen from Figure 3.13. The summation from the pixels in rectangle region ABCD may be computed as:

$$\sum_{(x,y)\in ABCD} i(x,y) = ii(D) + ii(A) - ii(B) - ii(C) \tag{3.11}$$

Anywhere ii(x, y) is integral image at the pixel point (x, y), i(x', y') is the original image, and ABCD rectangle region [64].

The integrated image may be utilized for the calculation of the Haar-like rectangular features, as can be seen from Figure. 3.11 (a-f). Features have been characterized as the intensity (i.e. weighted) difference between 2 rectangles and 4. For example, in the feature

(a), the element's value is the difference in the average value of the pixel in white and grey rectangles. Due to the fact that the rectangles are sharing angles, calculating two rectangular features (A&B) needs 6 array references, the 3 rectangle features (c&d) need 8 array references, and the 4 rectangle features (e&f) require 9 array references [64].

3- AdaBoost Training

As soon as the value of the integrated image has been found, the images must be categorized according to the needed criteria. In the process of the vehicle detection, the object must be categorized as a vehicle or an object without a vehicle. A single algorithm when using such classification may not accurately categorize the objects. Classification would be more efficient in the case where multiple classifiers were combined and merged. One of these classifiers is the AdaBoost classifier, where many weak classifiers are combined for the purpose of forming a robust workbook [65].

In the original form, AdaBoost training is utilized in order to enhance the classification performance of a weak learning algorithm. Viola and Jones introduced a variety of AdaBoost to define a small group of the features and train the workbook. In previous studies Vila and Jones had more than 180000 rectangular features associated with every one of the image sub-windows, which is much greater than pixels. Although each feature can be calculated very efficiently, the calculation of the whole group is still expensive. The main challenge is how to define a small set of these features [66].

AdaBoost feature identifies poor classifiers throughout the analysis of the vehicle features for the elimination of negative entries. The term "reinforcer" indicates that the classifiers at every stage of succession are themselves complex and made from the basic works utilizing different reinforcement methods. AdaBoost assigns weight to every one of the training samples and determines the likelihood that the sample will be dropped into training set. The Viola-Jones algorithm counts a weak workbook as:

$$q(x, f, p\theta) = \begin{cases} 1 & pf(x) < p\theta \\ 0 & \text{otherwise} \end{cases} \tag{3.12}$$

$\boldsymbol{f}$ is the element value, $\boldsymbol{\theta}$ represents the threshold value and p represents polarity, which indicates the inequality direction [65].

After that, the weak classifiers are processed for the purpose of achieving a strong workbook while minimizing the decrease in the false-positive positive rate. Strong classifier counts.

$$H(x) = \begin{cases} 1 & \sum_{t=1}^{T} a_t q_t(x) < \gamma t \\ 0 & \text{otherwise} \end{cases} \quad (3.13)$$

Where $a_t = \log \frac{1}{\beta t}$ and Υt are taken for the purpose of ensuring that all of the positive training samples have been accurately categorized [65].

4- Cascade Classifiers

Cascade Workbook is a series of weak workbooks to categorize image areas efficiently. Its aim is increasing the efficiency of the detection of the objects and reduction of the computation time. As exhibited in Figure. 3.14, every one of the nodes in the string has been considered a low rated and candidate for 1 Haar feature. The AdaBoost algorithm assigns the weights to nodes, and the node with the maximum weight comes first.

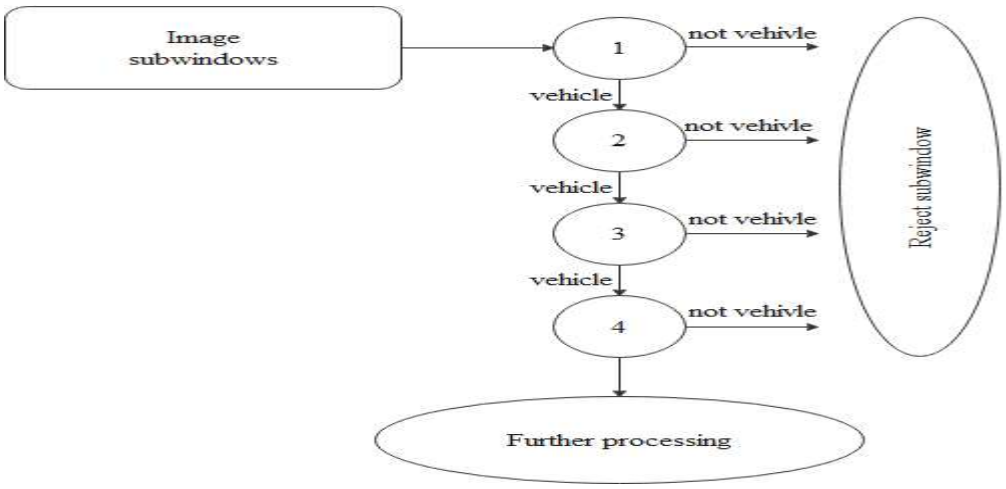

Figure 3.14: Cascade Classifier.

In the case where the filter fails to cross the image areas, this certain image sub-window is disposed of for additional processing. Then it is viewed as non-object. Which means that the processed image regions don't contain the object which requires being discovered. Which is critical to the workbook's efficiency, as almost all negative subfolders will be discarded in the 1st stage. However, when the image areas successfully pass filter, they move to next step that has a more complicated filter. Only the areas which have passed all filters with success match the object. This means that the areas of the image include the exposed object [62].

Every one of the workbook stages calls the region which is specified by sliding window's current position, either negative or positive. The positive means that an object has been discovered, and negative means that has not been any object found. In the case where a label has been negative, the classification of this region is completed, and detector moves the window to the following location. If the poster has been positive, the workbook passes the area to the following stage. The detector detects the object at the current position of the window when the last step categorizes the area positively.

The reason for the multi-stage classifier is the efficient and rapid rejection of non-existent sub-pumps. The next nodes in series in Figure 3.15 represent complicated works, in case of vehicle detections. The classifier has been utilized for the rejection of more false positives (i.e. the non-surface areas) for sub-windows [63].

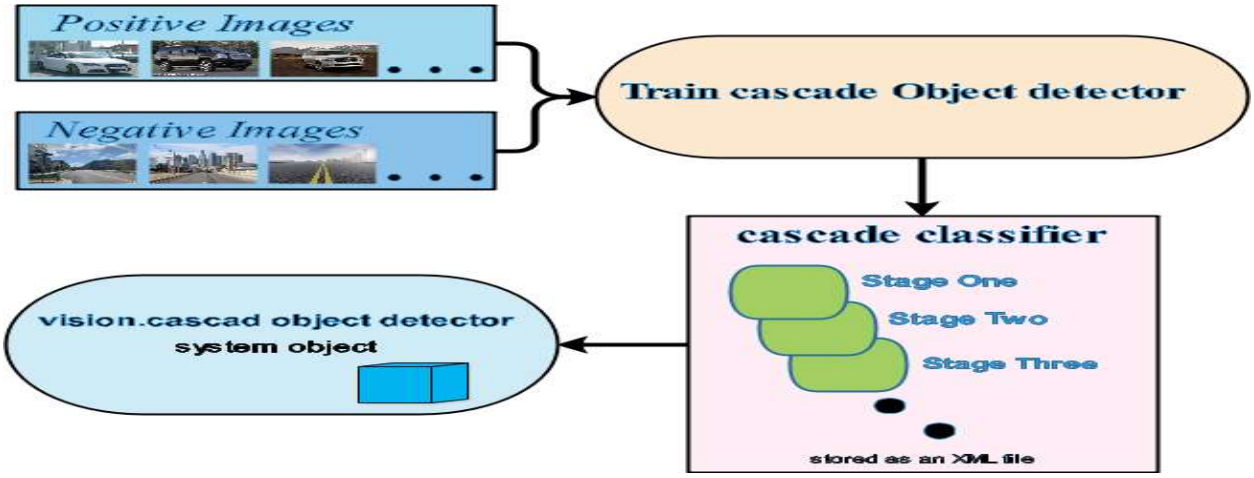

Figure 3.15: Example of Cascade Classifier.

Stages have been designed for rejecting the negative samples with maximum speed. It is assumed that most windows include no subject of interest. On the other hand, the true positive cases are uncommon and worth the time for the verification.

- True positivity occurs in the case where the positive sample has been accurately categorized.
- False positivity occurs in the case where the negative sample has been wrongly categorized as positive.
- False negativity occurs in the case where the positive sample has been wrongly categorized as negative.

For the purpose of working well, every one of the steps in a chain has to be having a low rate of the false-negatives. In the case where a stage is inaccurately describing a vehicle as negative, the classification is stopped, and the error can't be corrected. None-the-less, every one of the stages may be having a high rate of false-positives. Even in the case where the detector categorizes any incorrect vehicle as error, you can correct the error later. The addition of more stages results in the reduction of overall rate of false-positives, however, it also results in the reduction of overall rate of true positives.

Sequential workbook training needs a group of the positive samples and a group of the negative ones. One has to present a collection of positive images that have specific areas of interest to be used as complimentary samples. Image Labeler may be utilized for naming the objects of the interest with surrounding boxes. The Image Labeler will output a table to be used in the positive models. A group of the negative images have to be provided from which the function will automatically create the negative models, as shown in Figure 3.16. for the purpose of achieving adequate detection accuracy, the feature type, stage number as well as other functional parameters must be set [67].

Sequential workbook training needs a group of the positive samples and a group of the negative ones. One has to present a collection of positive images that have specific areas of interest to be used as complimentary samples. Image Labeler may be utilized for naming the objects of the interest with surrounding boxes. The Image Labeler will output a table to be

used in the positive models. A group of the negative images have to be provided from which the function will automatically create the negative models.

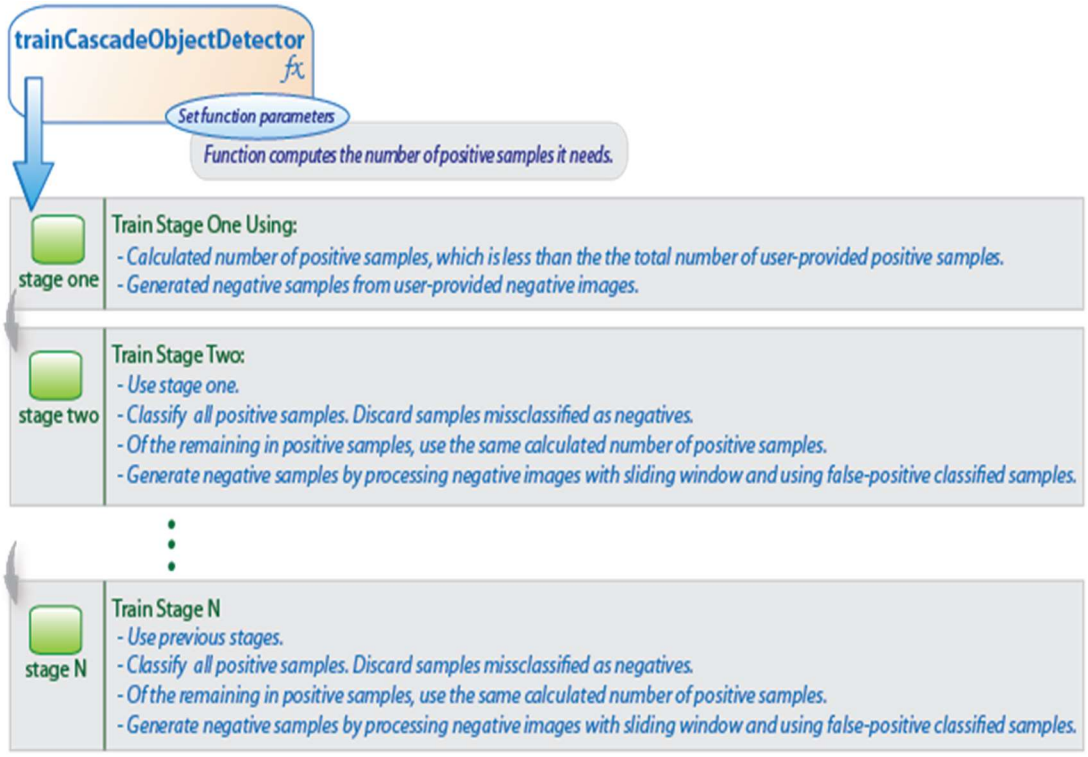

Figure 3.16: Train Cascade object detector [67].

3.2.2 Deep Learning

Deep learning can be defined as an AI function which simulates the work of the brain of the humans in data processing and generating patterns which will be used in the decision-making. Deep learning has been classified a sub-set of the machine learning in the field of AI which contains the networks that can uncensored learning from unstructured or unclassified data. which has been referred to as the deep neural network (DNN) or deep neural learning as well [68].

Deep learning is the term applied to many simple neural networks that are linked together. Deep learning has several layers, in which the 1st layer's output is the 2nd layer's input, the 2nd layer's output is the 3rd layer's input, etc. Neural network (NN) works by analyzing the relevant features of the input data, detecting patterns in that data, and then making

predictions about that data or similar data. The deeper the NN, the more layers it has, the more sophisticated the pattern it can distinguish [69].

The general object detection aims to locate and classify the objects present in any image and to mark those by rectangular perimeter boxes in order to show confidence in existence. The frameworks for detection methods for general objects can be fundamentally classified to 2 types. One of them follows a pipeline to reveal traditional objects, generates area proposals initially, and then categorized every presentation to various object classes [69].

Using deep convolutional networks (CNNs) reached impressive success in the vehicle detection. CNN has a very good capability for learning image features as well as the ability of performing a variety of the associated tasks, like the classification and the bounding box regression. The approach of the detection may, in general, be fragmented to 2 sections. Such 2-stage approach results in the generation of object filter box through various algorithms and after that, it categorizes the object with a CNN. The 1-stage approach creates no filter box, however, it transforms the problem of the positioning of bounding box to the object into a problem of regression for the processing directly [70].

The other is to directly reveal things as a problem of classification or regression and adopt a unified model for achieving the final result (locations and categories). Proposed region-based classifications fundamentally include.

The first algorithm class is suggestion of the region first. Which indicates that an object's potential areas are identified either with the conventional technologies of the computer vision (like the selective search) or the use of the RPN (i.e. the Region Proposal Network). As soon as small set of filter windows has been compiled, a certain number of the regression models may be formulated, and the models of classification for solving a problem of object detection state-of-art CNN based object detection models. This classification includes algorithms such as Faster R-CNN [71] R_FCN [72], and FPN-FRCN [73]. The approaches in such category are often referred to as the 2-stage approaches. In general, they are more precise, however, they are also more time consuming compared to 1-stage approach that will be provided below.

The 2nd class of the algorithms searches only for the objects in fixed locations with fixed sizes. Those sizes and locations have been selected strategically so that the majority of the scenarios have been covered. Those types of the algorithms often separate the native images to fixed-sized grid areas. Those algorithms are attempting at the prediction of a certain number of the objects with predefined shapes and sizes for each part. The algorithms which belong to that class have been referred to as the single-stage approaches. Some of the examples of those approaches are YOLO [74], Single Shot Multibox Detector (SSD) [75], and Retina-Net [76]. Algorithms in that class often work faster though they provide lower level of accuracy. This algorithm type is often utilized for applications which expect real-time detections.

- Faster R-CNN

R-CNN series, which have been presented in 2014 by R. Gircek and others, has been enhanced with the Fast R-CNN and then ultimately with fastest R-CNN. The differential method which makes the R-CNN faster is introducing the RPN, which is a fully trained and integrated CNN that predicts object boundaries and object markers in each list simultaneously. With the importance of RPN to the Raster CN-Faster, which remains a very good object detection framework which is available to the researchers, this study is focused upon the design of the RPN and the lockboxes concept and the non-maximal suppression [77].

R-CNN detector network consists more quickly of the feature extraction network that the CNN network usually pre-sets, similar to what we used for its predecessor, as shown the Figure 3.17. This is followed by two sub-networks that can be trained. The first is the Area Proposition Network, which, as can be implied from its name, is utilized for the creation of the proposals of the objects, and the 2nd one is utilized for the prediction of actual object class. Therefore, the main preference for the Faster R-CNN is RPN inserted back the final convolution layer. Which is trained for producing the proposals of the area in a direct manner with no external mechanisms, like the selective researches. After that, Return on the Investment pool, top seed, and a square reference which is equivalent to the Fast R-CNN have been used [71].

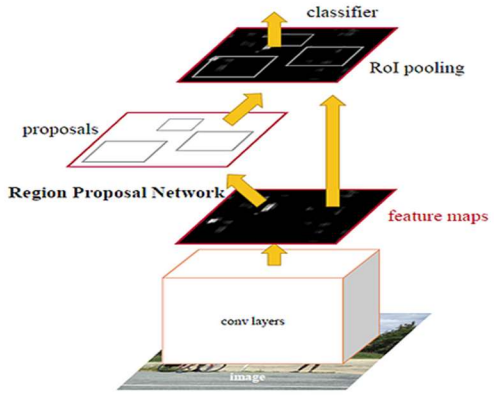

Figure 3.17: Unified Faster R-CNN Network [71].

- Region Proposal Networks (RPNs)

The RPNs first creates anchor boxes from wrapping feature map which has been produced by underlying net. The anchor has been centered on sliding window and is joined to an aspect ratio and a scale. For the purpose of matching the callback and processing speeds, 3 anchor squares measurements of 128, 256, & 512 are used and three 1: 1, 1: 2, and 2: 1 docking ratios for each anchor nine anchors at every one of the sliding positions. For a 14 x 14 convolution feature map, there have been totally 1764 anchors, as can be seen from Figure. 3.18.

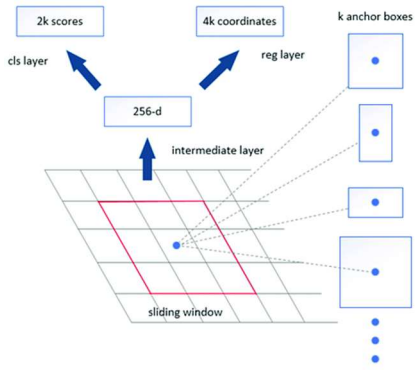

Figure 3.18: Anchor boxes which are produced by the RPN [78].

After that, the RPN will take all fixing boxes and output 2 distinct outputs for every anchor. The first is the score of the target, which means the possibility of the anchor being. The 2nd output is the slope of the bounding box to adjust the anchors for better fitting the object [78]. As exhibited in Figure3.18.

When the missing functions for all anchors are optimized here, the direction of negative examples is achieved. For this reason, a random 256 anchors' sample is shown in the image instead. Accordingly, if there are more than 128 positive samples, they are filled with stacked samples. Otherwise, it is full of negative examples. Additionally, after Fast R-CNN multitasking loss, the target function has been reduced to a minimum. This loss function is shown in the following equation [79].

$$L(\{p_i\},\{t_i\})\frac{1}{N_{cls}}\sum_i L_{cls(p_i,p_i^*)+\lambda\frac{1}{N_{reg}}\sum iPiL_{reg(t_i,t_i^*)}} \tag{3.14}$$

Where, i represents the anchor index in mini-batch and p_i represents predicted anchor's likelihood i being an object. The label of the ground-truth p_i^* will be 1 in the case where the anchor has been positive, and will be 0 otherwise, t_i represents a vector that stands for the four parameterized predicted bounding box coordinates, and t_i^* is that of ground-truth box which is related to positive anchor. The loss of the classification L_{cls} is log loss over 2 classes (not object versus object). For regression loss, $L_{reg}(t_i,t_i^*)=R_{reg}(t_i,t_i^*)$ has been utilized, where R represents the function of the robust loss (i.e. smoothL_1) has been defined in. A term $p_i^*L_{reg}$ indicates that the loss of the regression has only been activated for the positive anchors $p_i^*=1$ and has been disabled in the opposite case $p_i^*=0$ the cls and reg layers' outputs which consist respectively of (p_i) & (t_i) [71].

Both terms have been normalized with N_{cls} & N_{reg} then weighted with a balancing parameter λ. In this implementation (as in released code), cls term in (Eq3.14) has been normalized by mini-batch size $(in\ other\ words, N_{cls}=256)$ and reg term has been normalized with the number of the anchor locations$(in\ other\ words, N_{reg}\sim 2400)$. By

default λ is set to 10, which is why, cls as well as the reg terms have been roughly weighted equally [71].

- Region Proposal Networks and Fast RCNN

In the final step, authors have described the way of proposing a network of training for the generation of the regions with no consideration of ways for detecting the organisms according to area CNN utilizes those suggested frames. For the disclosure, the Fast R-CNN has been utilized. How to train them in same architecture of the network for training a multi-tasking model of the network with shared turn. Authors were aware of the fact that in the case where the model of the network is separately trained on 2 separate tasks, even in the case where its parameters and structure are identical, then the core of the kernel in the convolution layers will be changed in various directions, which will result in the incapability of sharing the weights of the network. As a result, the authors has utilized the training approaches which have been detailed in steps that can be seen from Figure 3.19 [80].

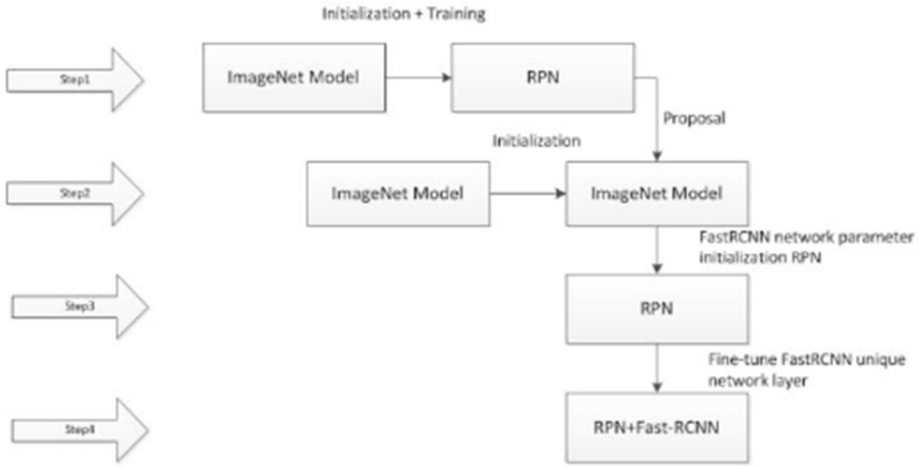

Figure 3.19: RPN and R-CNN Rapid Training Stream [80].

- Step1: configure with the model of the Image-Net, train RPN in an independent manner;
- Step2: Keep using the template configuration of Image Net, but utilizing a suggestion from the preceding RPN as input, Fast-RCNN training; therefore, the

parameters of every one of the layers of those two networks have not been shared whatsoever;

- Step3: Configure a new network of the RPN with the use of the network parameters of the Fast-RCNN for the Step2 however, set learning rate of these convolution layers between Fast-RCNN and RPN to zero, not update, update the layer of the RPN network, retrain, this time, both networks share every standard bypass layer;
- Step4: The layers of the network which remain constantly shared join the unique network layer of the Fast-RCNN as well, for the purpose of forming a network and keep training and setting unique network layer to the Fast-RCNN. Now, the system has realized the imagination of the target, the internal network for the prediction of the proposal and achieving the function of the detection.

- You Only Look Once (YOLO)

YOLO can be defined as a Real-time Object Detection, a very sufficient algorithm of object detection, which includes several of the newest ideas that emerge from the research community of the computer vision. The object detection has been described as a critical ability of the autonomous vehicles technology. It is one of the computer vision areas, exploding and working considerably better than it did just a few years back [81].

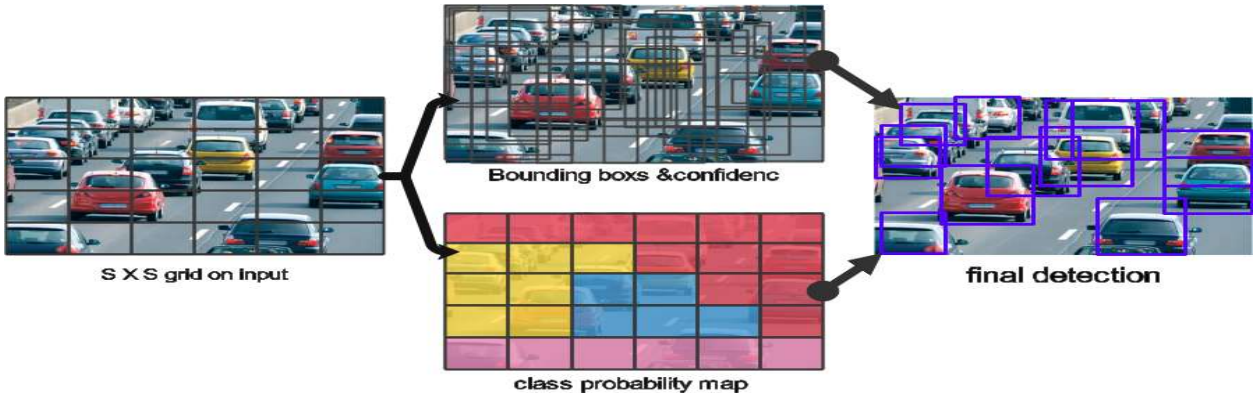

Figure 3.20: Main idea of YOLO.

YOLO uses a completely different method. It is a smart CNN that performs the real-time object detection. This algorithm applies a single NN to the entire image, after that, it splits the image to areas and can predict the surrounding squares and every area's probability. The projected probabilities weigh those surrounding boxes [74].

YOLO has gained such popularity due to the fact that it has achieved high accuracy with the possibility of real-time operation. The algorithm "looks only once" in the sense that it only needs one forward passing through NN for the purpose of making the predictions. Following the blocking of the limit (ensuring that the algorithm of the object detection can detect every one of the objects once only), it will then output the surrounding squares' recognized objects.

YOLO partitions the input image in the S × S grid and every network cell is effective for the prediction of the central object in the grid cell as shown in figure 3.20. All grid cell predicts the bounding squares of B and their corresponding confidence degrees. Officially, confidence degrees have been characterized as Pr (Object)*$\mathrm{IOU}_{\mathrm{pred}}^{\mathrm{truth}}$, indicating the probability of having (Pr (Object) ≥ 0) and showing confidence in their prediction ($\mathrm{IOU}_{\mathrm{pred}}^{truth}$). At the same time, irrespective of the number from box C, the likelihoods of the conditional class (Pr (Class_i|Object)) in every one of the grid cells has to also be expected. Note that the contribution is calculated only from the grid cell which contains the object.

$$\Pr(\mathrm{Object}) * \mathrm{IOU}_{\mathrm{pred}}^{\mathrm{truth}} * \Pr(\mathrm{Class}_i|\mathrm{Object}) = \mathrm{pr}(class_i) * \mathrm{IOU}_{\mathrm{pred}}^{truth} \tag{3.15}$$

Where the current probability of objects of the class in the square and the suitability between the expected fund and the element is taken into account. Throughout the training, the following loss function has been improved,

$$\begin{aligned}\sum_{i=0}^{s^2}\sum_{j=0}^{B} 1_{ij}^{obj}\left[(x_i - x_j)^2 + (y_i - y_j)^2\right] + \lambda_{coord}\sum_{i=0}^{s^2}\sum_{j=0}^{B} 1_{ij}^{obj}\left[(\sqrt{w_i} - \sqrt{\hat{w}_i})^2\right. \\ \left. + \left(\sqrt{h_i} - \sqrt{\hat{h}_i}\right)^2\right] + \sum_{i=0}^{s^2}\sum_{j=0}^{B} 1_{ij}^{obj}\left(C_i - \hat{C}_i\right) \\ + \lambda_{coord}\sum_{i=0}^{s^2}\sum_{j=0}^{B} 1_{ij}^{noobj}\left(C_i - \hat{C}_i\right) + \sum_{i=0}^{s^2} 1_i^{obj}\sum_{c\in classes}(p_i(c) - \hat{p}_i(c))^2\end{aligned} \tag{3.16}$$

In a particular cell i, (x_i, y_i) denotes the center of the square concerning the boundaries of the grid cell, $(\mathrm{w}_i, \mathrm{h}_i)$ is the measured width and height based on the size of the image, C_i is the degrees of confidence,1_i^{obj} represents the presence of the objects and 1_{ij}^{obj} indicates that the prediction has been made through predicting jth bounding box. It has to be noted that only in the case where the object is in grid cell, the loss function penalizes the errors of the classification. Likewise, in the case where the predictor has been "responsible" for the basic truth fund (in other words, the maximum IoU of any predictor in the grid cell has been accomplished), errors in the bounding square coordinates have been penalized [69].

YOLO is made up of 2 FC layers and24 layers of conversion, in which some conversion layers create groups of starting units with 1x1 reduction coats, which are followed by 3x3 conversion layers. The network has the ability of processing the images in the real-time at a 45fps, and simplified Fast YOLO version may reach 155fps with more sufficient results compared to other real-time detectors. In addition to that, the YOLO creates a smaller number of the false positive cases in background, which makes it possible for the cooperation with the Fast R-CNN [69].

- SSD (Single Shot Multibox Detector)

YOLO has a difficulty handling small objects in groups, resulting from the strong spatial restrictions that have been imposed on the predictions of the bounding [82]. In the meantime, YOLO is striving for the generalization of the objects in the unusual/new configurations / proportions and results in the production of rather rough features as a result of the multiple operations of the reduction.

Aim at those issues, Liu et al. SSD was suggested, which has been inspired by anchors which are approved in the RPN and MultiBox as well as the multi-band representations. Looking at a certain feature map, rather than the fixed networks which are supported in the YOLO, SSD benefits from a group of the default anchor boxes that have various aspect ratio values and metrics to estimate surrounding boxes' output space. To handle objects of different sizes, the grid combines forecasts from many feature maps with various resolutions.

SSD a one-shot detector that uses the VGG16 network as a characteristic extractor (equal to CNN into Faster R-CNN). Next, we add the custom (blue) warp layers and then use the (green) warp filters to make predictions as explain in Figure 3.21 [75].

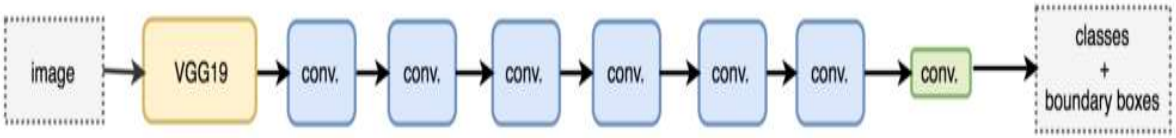

Figure 3.21: Single shot prediction for both classification and location.

However, warping layers reduce spatial dimension and accuracy, so the above model can only detect large objects. To fix this, we make standalone object discoveries from multiple feature maps.

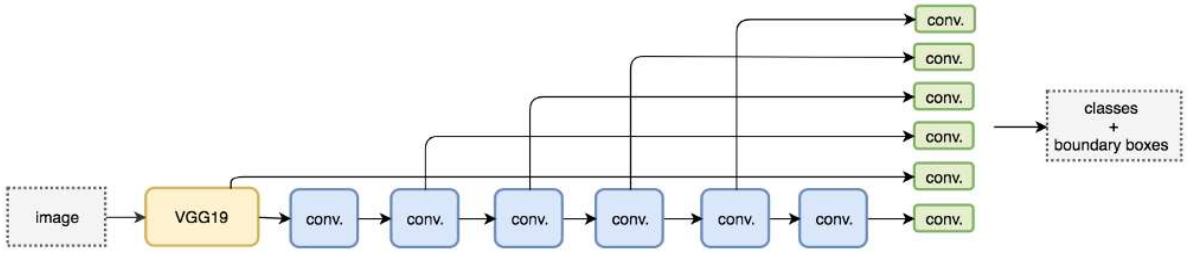

Figure 3.22: Use multi-scale feature maps for detection.

The VGG16 infrastructure, SSD, adds multiple feature layers to the network end, as shown in Fig.3.22 and is responsible for predicting virtual squares displacement with different cross-scale metrics and ratios and the related confidence. The network has been trained in a reasonable amount of translation loss (like Smooth L-1) and loss of confidence (like Softmax). The final results of the detection are obtained through performing NMS on the redundant multiplex link boxes[69].

Integration with hard passive mining increased data and carefully selected virtual anchors. The SSD was significantly outperforming the Raster R-CNN in accuracy over the COCO and PASCAL VOC while being 3 times faster. SSD-300 (Size of the Input Image is 300300) runs at 59fps, which is more precise and sufficient compared to the YOLO. None-the-less, the SSD is not adept at handling small objects, and that may be mitigated through the

adoption of better backbones for the feature extraction (such as ResNet101), and adding decoupling layers with skip connections in order to provide better network architecture design (such as block Trunk and dense mass) and additional broadband context, as shown in Figure.3.23 [73].

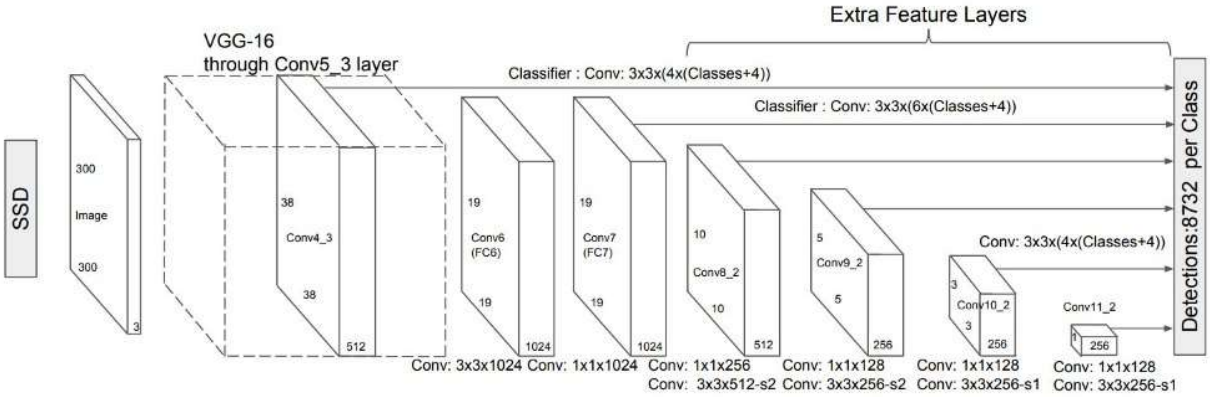

Figure 3.23: The architecture of SSD 300 [75].

- YOLO-V3

YOLO-v3 has been presented in Apr. of 2018 and it has added additional small enhancements, which included the fact that the bounding boxes are predicted at a variety of the scales. The underlying network's meaty part, Dark-Net, has been expanded in that version so as to be having 53 convolution layers, as can be seen Figure 3.24 [83].

Change YOLOv3 to the most complex backbone of feature extraction. Fundamentally, the Darknet-53 includes 3x3 and 1x1 filters with skip like the connections such as the Res-Net's remaining network. Darknet53 has billion floating-point operations (BFLOP) less than the ResNet-152 however, it accomplishes the same accuracy of sorting with double the speed [75].

YOLO has approached this problem from object detection in a radically different manner. Redirects the entire image only one over the net. The SSD is an additional algorithm of object detection, which is redirecting the image once through a deep learning net, however, the YOLO-v3 has been considerably faster compared to the SSD while accomplishing quite

similar level of the accuracy. YOLO-v3 delivered faster compared to the real-time results on the M-40, Titan-X, or 1080Ti GPU.

Divides the image to a 13x13 cell grid. The size of those 169 cells varies according to the size from the entry. For the input size of 416 x 416 that used, the cell size was 32 x 32. Then each cell was responsible for the prediction of the number of the squares in this image. [84].

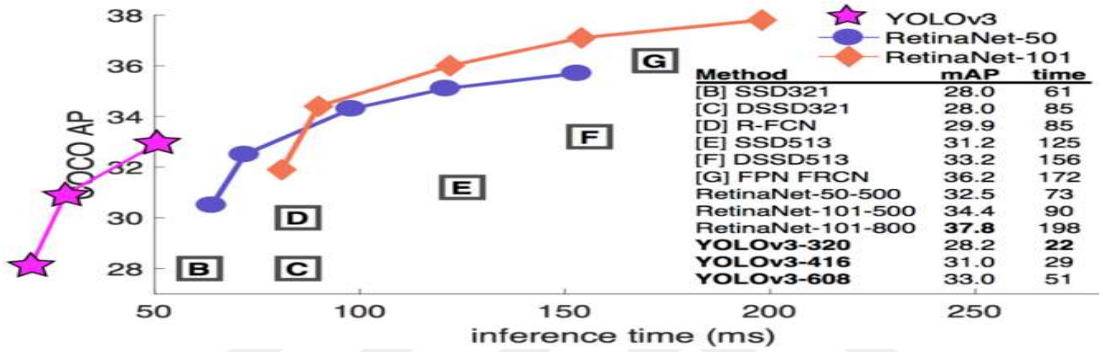

Figure 3.24: YOLO-v3 runs operates considerably faster compared to the other approaches of detection with similar efficiency. The times from either a TitanX or M-40, they have been basically same GPU[83].

DarkNet can be defined as framework for NN training, is open source, and written in C / CUDA (Compute Unified Device Architecture) and plays the role of a basis for the YOLO. The original repository, by J Reedmen (who is also the first writer of YOLO Paper). Dark Net is utilized as a model for the YOLO training, which means it defines the network's structure as can be seen Figure 3.25.

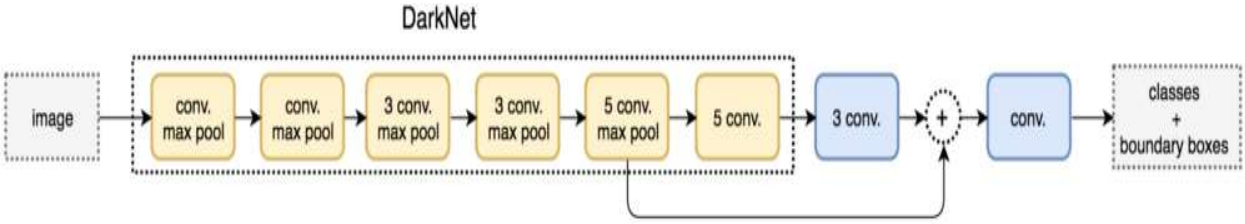

Figure 3.25: YOLOV3 DarkNet is used to perform detection feature, followed by convolutional layers.

3.3 ESTIMATE TRAFFIC CONGESTION

The methods detect the vehicles and count the number of the vehicles in the frame every T second (the T value is assigned by the admin, for example, T=10 second).

If the count of vehicles is greater than the given threshold (threshold value is assigned by the admin, for example, threshold=30 vehicles per T) and the threshold value also varies from street to street depending on the street size and the difference in vehicle movement in the street between day, night and rush hour, the system will send a notification to the driver of traffic congestion on this road through electronic traffic signs or mobile applications by taking alternative routes or reducing the speed of vehicles.

If the number of vehicles is below the specified limit, the system will send a notification to the traffic status driver on this road as normal.

3.4 MATERIALS

In this study, two data sets are used, which are Pascal visual object classes (VOC) and KITTI Dataset. The first method is implemented with deep learning classifier and the second is implemented with haar cascades.

3.4.1 Pascal Visual Object Classes (VOC) Dataset

The (VOC) test is a standard in recognizing and detecting the visible object class, providing seeing and machine learning groups a standard data set of images, annotations, and regular evaluation procedures. The challenge and its associated data sets are organized annually since 2005 until now as a standard for object discovery [85].

The classic Faster R-CNN achieved better accuracy for the classification of 1,000 objects and tests (including various vehicle targets). The original set of training images comes from the MIT dataset, the California Institute of Technology vehicle data-base, and images of various vehicle types on network, which include cars, minibuses, and SUVs 3[80].

The car sample data set was produced: Image: 5042 / JPG, Posters: 5042 / XML, Binding Boxes: 5038 Car / Rectangles, 987 Minibus / Rectangles, SUV1207 / Boxes. Amongst them, various types of vehicle datasets are specified in Table3.1.

Table 3.1: Special Characters' Frequency

Vehicles	Total data-set	Training set	Test set
car	5,038	3,358	1,680
minibus	987	658	329
Suv	1,207	804	403

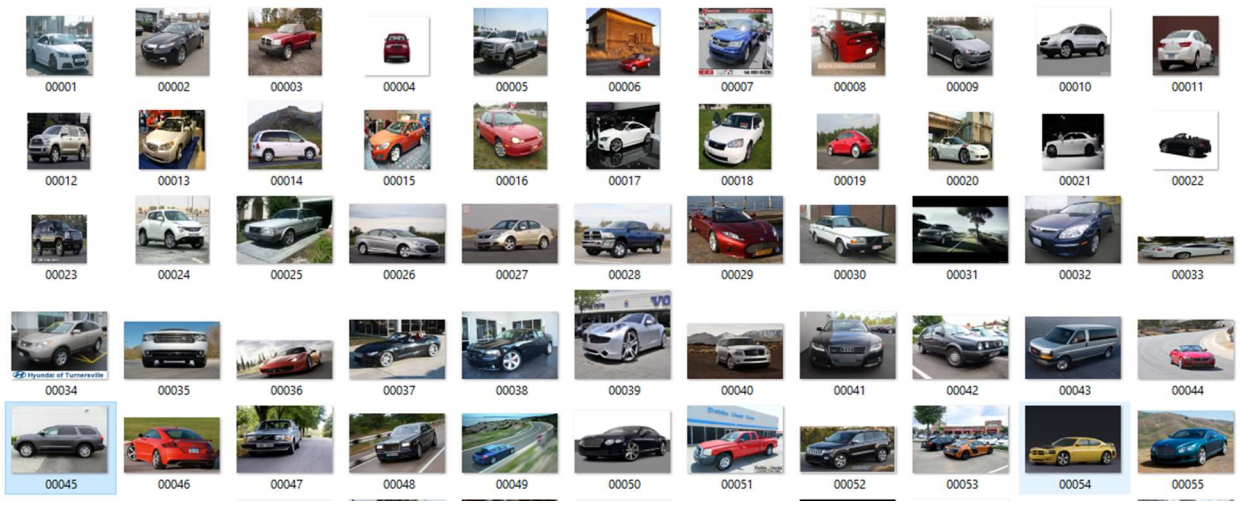

Figure 3.26: Original image examples of various vehicle types.

3.4.2 KITTI Dataset

Building a vehicle detection operation required an annotated data set for vehicles, creating ideal threshold values for features, and training workbooks. Also, the system requires a camera to detect vehicles in real-time. This system is implemented in python and detects vehicles at a rate of 10-20 fps for images with a resolution of 500 x 400. The size of the Detector window is 40 x 40 pixels.

The object detection and object orientation assessment standard consist of 7,481 training images and 7,518 testing images, which comprise totally 80256 named objects. All of the images have been colored and saved in the .png format. Authors calculated fine summation curves to detect objects and similarities and summon curves to explore common objects and estimate the direction. In the latter case, not only the square surrounding the two-dimensional object should be correctly positioned, but the direction estimate is also evaluated in the bird's eye width. For the classification of methods, the authors calculated the average accuracy and average similarity direction [86]. The data set is used by the haar cascades classifier for taking positive and negative images, as shown in Figures 3.27 and 3.28.

Figure 3.27: KITTI Dataset positive images.

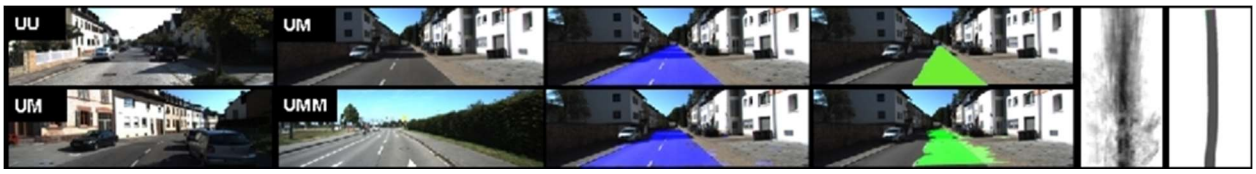

Figure 3.28: KITTI Dataset negative images.

4. IMPLEMENTATION AND RESULTS

In this chapter, the methods that were used to test and evaluate with several system variables which include number of vehicles, threshold value, feature extraction process (vehicle detection, computation of the number of vehicles in a frame per second, and determination threshold value).

The method used was implemented by using the programming language python, with the library OpenCV, NumPy, cvlib, and matplotlib. The system specification are as follows: the processor Intel corei7-8565U CPU@1.80GHz 1.99GHz, the installed memory of 8 GB DDR4 and system type 64-bit OS, and x64 based processor.

4.1 CONFUSION MATRIX

The confusion matrix can be defined as the summary of the forecast results at a problem of classification. The number of the correct and incorrect forecasts are summed up with the count values and are divided according to each category. Which has been considered as the key to confusion matrix which illustrates how your rating form is confused when it does forecasts. It provides us insight into not only the mistakes a classifier makes however more importantly the types of mistakes that are made [87].

	Class 1 Predicted	*Class 2 Predicted*
Class 1 Actual	**TP**	**FN**
Class 2 Actual	**FP**	**TN**

Figure 4.1: Two classes Confusion matrix that described in the below paragraph.

Definition of the Terms:

- Positive (P): It refers to Traffic congestion.
- Negative (N): It refers to normal.
- True Positive (TP): notice is positive, and has been expected to be positive for example, the incoming video is a traffic congestion and is classified as a traffic congestion.
- False Negative (FN): notice is positive, but has been expected negative (for example, the entrance video is a traffic congestion and is classified as a normal).
- True Negative (TN): notice is negative, and has been expected to be negative (for example, the entrance video is normal and is classified as a normal).
- False Positive (FP): notice is negative, but has been expected positive (for example, the entrance video is normal and is classified as a Traffic congestion).

1-Accuracy: The sum of accurate predictions that have been divided by the total of all data. The optimal accuracy is 1, while the worst is 0.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

2-Sensitivity (SN): is counted as the number of the correct positive predictions that have been divided by the total number of the positive cases. The optimal sensitivity is 1, while the worst is 0.

$$\text{Sensitivity (SN)} = \frac{TP}{TP + FN} \tag{4.2}$$

3- Specificity (SP): is rated as the number of correct negative forecasts divided by the total number of negatives. The best specificity is 1.0, while the worst is 0.0.

$$\text{Specificity (SP)} = \frac{TN}{TN + FP} \tag{4.3}$$

4- F-score: represents the harmonic average of sensitivity and specificity.

$$\text{F} - \text{score} = \frac{2TP}{2TP + FP + FN} \tag{4.4}$$

5- The false-positive rate (FPR): is counted as the amount of the false-positive forecasts split by the negatives' total number. The better false positive rate is 0, while the worse is 1.

$$\text{FPR} = \frac{FP}{FP + TN} \tag{4.5}$$

6- The false-negative rate (FNR): is computed as the number of the false-negative predictions split by the total positive number. The best false-negative rate of 0, while the worse of 1.

$$\text{FNR} = \frac{FN}{FN + TP} \tag{4.6}$$

7-Error: The sum of incorrect forecasts divided by total of the entire data. The optimal value for the error is 0, while the worst is 1.

$$Error = 1 - Accuracy \tag{4.7}$$

4.2 THE TEST METHODS

Traffic congestion detection system performance was tested based on the vehicle identification algorithms used (haar cascade and deep learning) using surveillance cameras. The used system has been tested and evaluated using the confusion matrix outcomes. Depend on of test results of Accuracy as explain in (Equation 4.1), it is described as the ratio between correctly classified samples to total samples tested. The test results focus on an F score (Equation 4.4) because it is the harmonic average of specificity and sensitivity. Thus, this score takes both FP and FN into account. On the other hand, the FPR confirmation (Equation 4.5) is essential because it blends the data that have been classified Error (non-traffic congestion).

4.3 EXPERIMENTAL RESULTS USING HAAR CASCADE

Samples were taken from different videos, including non-traffic congestion and traffic congestion. The test was performed using Haar cascade to reach the final results, either traffic congestion or a non-Traffic congestion for each segment and according to the time specified for each video. At the beginning, the vehicles are determined, calculated, and the threshold value determined (e.g. threshold=15).

Each video has been tested as the following:

At the beginning of testing, the number of vehicles is calculated in one second from each video, and accordingly the results for each second are shown in Table 4.1. as follows:

- **Traffic congestion:** It's occurred when the number of vehicles in the tested frame is ≥ the value of the threshold.
- **Non-traffic congestion:** It's occurred when the number of vehicles in the tested frame is < threshold value.

The final state of the tested video has been calculated based on the accumulated results. If the traffic congestion count is greater than non-traffic congestion count, then the final result of the video is traffic congestion and vice versa.

Table 4.1: Experimental results using Haar cascade for each one second

Videos	Status Each one Second		predicted		Final Status
	Traffic jam	Non-Traffic	Traffic%	Non-Traffic%	
TCV1	40	21	65.57%	34.43%	Traffic
TCV2	63	11	85.13%	14.87%	Traffic
NTV3	0	59	0.0%	100.0%	Non-Traffic
TCV4	55	6	90.16%	9.84%	Traffic
TCV5	49	12	80.32%	19.68%	Traffic
NTV6	35	26	57.37%	42.63%	Traffic
NTV7	11	50	18.04%	81.96%	Non-Traffic
NTV8	34	14	70.8%	29.2%	Traffic
TCV9	52	10	80.33%	19.67%	Traffic
NTV10	44	18	70.96%	29.04%	Traffic
NTV11	47	16	74.60%	25.4%	Traffic
NTV12	3	60	4.77%	95.23%	Non-Traffic
TCV13	44	10	81.48%	18.52%	Traffic
TCV14	35	1	97.22%	2.78%	Traffic
NTV15	0	35	0.0%	100.0%	Non-Traffic
NTV16	0	61	0.0%	100.0%	Non-Traffic
NTV17	0	61	0.0%	100.0%	Non-Traffic

Where, TCV is the traffic congestion video, NTV is the non-traffic video, status is the video state that is either traffic congestion or non-traffic congestion based on the tested frame each time.

Second test, see Table 4.2. where the number of vehicles five second is calculated from each video, and accordingly the results for each five second are shown as follows:

Table 4.2: Experimental results using Haar cascade for every one seconds

Videos	Status Each Five Second		predicted		Final Status
	Traffic	Non-Traffic	Traffic%	Non-Traffic%	
TCV1	10	3	76.925	23.08%	Traffic
TCV2	14	1	93.33%	6.67%	Traffic
NTV3	0	12	0.0%	100.0%	Non-Traffic
TCV4	11	1	91.88%	8.12%	Traffic
TCV5	12	1	92.30%	7.7%	Traffic
NTV6	6	7	46.16%	53.84%	Non-Traffic
NTV7	5	8	38.47%	61.53%	Non-Traffic
NTV8	6	4	60.0%	40.0%	Traffic
TCV9	11	2	84.61%	15.39%	Traffic
NTV10	9	4	69.23%	30.77%	Traffic
NTV11	10	3	76.92%	23.08%	Traffic
NTV12	0	13	0.0%	100.0%	Non-Traffic
TCV13	9	2	81.81%	18.19%	Traffic
TCV14	7	0	100.0%	0.0%	Traffic
NTV15	0	7	0.0%	100.0%	Non-Traffic
NTV16	0	13	0.0%	100.0%	Non-Traffic
NTV17	0	13	0.0%	100.0%	Non-Traffic

Where, TCV is the traffic congestion video, NTV is the non-traffic video, TP represents the true positive, TN represents the True Negative, FP represents the false positive and FN represents the false negative.

By using Haar cascade test which seeks to find out how accurate the haar cascade classifier can be with the used videos. The Confusion matrix is used to calculate all performance measures. As discussed in Section (4.1), the results in Table4.3 illustrate that average accuracy is 79.41%, the average F-score is 80.06%, and the average FPR is 0.35.

Table 4.3: Experimental results using Haar cascade in details

Threshold	Accu	Spec	Sens	F1	FPR	FNR	error
One	76.47	60.0	100.0	77.77	0.4	0.0	0.23
Five	82.35	70.0	100.0	82.35	0.3	0.0	0.17
Avg.	**79.41**	**65.0**	**100.0**	**80.06**	**0.35**	**0.0**	**0.2**

Where, Accu is the accuracy, Spec is the specificity, Sens is the sensitivity, Spec is the specificity, F1 is the F1-score, FPR represents the false positive rate, FNR represents the false negative rate, and Avg is the Average.

4.4 EXPERIMENTAL RESULTS OF THE DEEP LEARNING

Samples were taken from different videos, including non-traffic congestion and traffic congestion. The test was performed using deep learning to reach the final results, either traffic congestion or a non-Traffic congestion for each segment and according to the time specified for each video. At the beginning, the vehicles are determined, calculated, and the threshold value determined (e.g. threshold=15).

Each video has been tested as the following:

At the beginning of testing, the number of vehicles is calculated in one second from each video, and accordingly the results for each second are shown in Table 4.1. as follows:

- **Traffic congestion:** It's occurred when the number of vehicles in the tested frame is ≥ the threshold value.
- **Non-traffic congestion:** It's occurred when the number of vehicles in tested frame is < threshold value.

The final state of the tested video has been calculated based on the accumulated results. If the traffic congestion count is greater than non-traffic congestion count, then the final result of the video is traffic congestion and vice versa.

Table 4.4: Test videos of each one second using Deep learning

Videos	Status Each one Second		predicted		Final Status
	Traffic jam	Non-Traffic	Traffic	Non-Traffic%	
TCV1	34	26	56.66	43.34	Traffic
TCV2	65	8	89.04	10.96	Traffic
NTV3	0	58	0.0	100.0	Non-Traffic
TCV4	41	19	68.33	31.67	Traffic
TCV5	48	12	80.0	20.0	Traffic
NTV6	1	59	1.61	98.33	Non-Traffic
NTV7	0	60	0.0	100.0	Non-Traffic
NTV8	10	37	21.28	78.72	Non-Traffic
TCV9	31	30	50.81	49.19	Traffic
NTV10	0	61	0.0	100.0	Non-Traffic
NTV11	20	43	31.75	68.25	Non-Traffic
NTV12	1	61	1.62	98.38	Non-Traffic
TCV13	35	19	64.81	35.19	Traffic
TCV14	34	1	97.14	2.86	Traffic
NTV15	0	35	0.0	100.0	Non-Traffic
NTV16	1	60	1.64	98.36	Non-Traffic
NTV17	0	61	0.0	100.0	Non-Traffic

Where, TCV is the traffic congestion video, NV is the non-traffic video, TP represents the true positive, TN represents the True Negative, FP is the false positive and FN represents the false negative.

Second test, see Table 4.5. where the number of vehicles five second is calculated from each video, and accordingly the results for each five second are shown as follows:

Table 4.5: Test videos of each five second using Deep learning

Videos	Status Each one Second				Final Status
	Traffic jam	Normal	Traffic%	Non-Traffic%	
TCV1	8	5	61.53%	38.47%	Traffic
TCV2	12	2	85.66%	14.34%	Traffic
NTV3	0	12	0.0%	100.0%	Non-Traffic
TCV4	8	4	66.66%	33.34%	Traffic
TCV5	9	3	75.00%	25.0%	Traffic
NTV6	0	13	0.0%	100.0%	Non-Traffic
NTV7	0	13	0.0%	100.0%	Non-Traffic
NTV8	2	7	22.23%	77.77%	Non-Traffic
TCV9	5	7	41.67%	58.33%	Non-Traffic
NTV10	0	13	0.0%	100.0%	Non-Traffic
NTV11	4	8	33.34%	66.66%	Non-Traffic
NTV12	0	13	100.0%	100.0%	Non-Traffic
TCV13	7	3	70.0%	30.0%	Traffic
TCV14	7	0	100.0%	0.0%	Traffic
NTV15	0	6	0.0%	100.0%	Non-Traffic
NTV16	0	13	0.0%	100.0%	Non-Traffic
NTV17	0	13	0.0%	100.0%	Non-Traffic

Where, TCV is the traffic congestion video, NTV is the non-traffic video, TP represents the true positive, TN is the True Negative, FP represents the false positive and FN is the false negative.

This test seeks to find out how accurate the Deep learning classifier can be with the used videos. The Confusion matrix is used to calculate all performance measures. As discussed in Section4.1, the results in Table4.6 illustrate that average accuracy is 97.11, the average F-score is 92.15, and the average FPR is 0.0.

Table 4.6: Test to accurate the Deep learning classifier

Threshold	Accu	Spec	Sens	F1	FPR	FNR	error
One	100.0	100.0	100.0	100.0	0.0	0.0	0.0
Five	94.11	100.0	85.71	92.30	0.0	0.14	0.05
Avg	97.11	100.0	92.85	96.15	0.0	0.07	0.025

Where, Accu is the accuracy, Spec is the specificity, Sens is the sensitivity, Spec is the specificity, F1 is the F1-score, FPR represents the false positive rate, FNR is the false negative rate, and Avg represents the average.

After obtaining the classifiers' results to detect traffic congestion, a comparison was made between them. After the comparison, it was found that deep learning is better in terms of accuracy, F1, FPR, processing speed (implementation time), and lower error rate. Compared to Haar cascade, as shown in the table below 4.8

Table 4.7: The comparison between haar cascade and deep learning

Method	Accu	Spec	Sens	F1	FPR	FNR	error
Haar	79.41	65.0	100.0	80.06	0.35	0.0	0.2
Deep NN	**97.11**	**100.0**	**92.85**	**96.15**	**0.0**	**0.07**	**0.025**

Where, Accu is the accuracy, Spec is the specificity, Sens is the sensitivity, Spec is the specificity, F1 is the F1-score, FPR is the false positive rate and FNR is the false negative rate.

4.5 COMPARISON WITH OTHERS PUBLICATIONS

Our proposed system in real-time traffic jam detection has achieved excellent results in both classifiers (Haar cascade and Deep learning). Table4.9 lists a comparison between the proposed methods and other publications.

Table 4.8: The comparison with other published works

Authors	Classifiers	Dataset	Data type	Metrics (%)
Shaif in 2017 [40]	Haar-Like Features	Private	3 Videos	97.10% Accuracy
Pranamesh et al in 2018 [44]	Yolo & DCCN	Private	Images	91.50% Accuracy
Revanth in2.18[42]	DCNN	Private	Images	90.20% Accuracy
Chan-Tong in 2019[45]	Yolov3	Private	Images	87.00% Accuracy
Our method	Haar cascade	Private	17 Videos	79.41% Accuracy
Our method	**Deep learning**	**Private**	**17 Videos**	**97.11% Accuracy**

The proposed method has proved to be detectable in traffic congestion with high acceptance accuracy than the other methods shown in Table 4.9. The methods were used in other research was different. This study was used video while the other research were used images as demonstrated, which affect processing time, cost, execution speed, and inconsistency in identifying traffic congestion in real-time.

5. CONCLUSIONS AND RECOMMENDATIONS

5.1 CONCLUSIONS

In the preceding chapters, the proposed methods needed to achieve a system for detecting vehicles in the highway roads to reduce the traffic congestion, and all the results obtained from all proposed methods were listed.

This chapter will outline the most critical conclusions reached after studying the problem and the proposed solution, as summarized in the following list:

1- The data type that have been used in this thesis was were video recorded with CCTV cameras. The total number of videos were 17 videos, with a duration of 16.38 minutes.
2- The pre-processing stage was applied to the videos after framing it; first, they converted each colored frame into a grayscale frame. Second, illumination improvement using the gamma correction and histogram equalization methods. Third, noise reduction has been applied using a median filter and average filter.
3- In the processing stage, two classifiers have been applied to the data for vehicle detection. The first one was the Haar cascade classifier based on digital image processing. The second one was, Deep learning classifier based on machine learning.
4- Traffic congestion estimation has been calculated based upon the number of vehicles per time using a specific threshold depending on the city, road, workdays, etc. Two approaches were used to estimate the traffic congestions either every one second or every five seconds. For example, count the number of the vehicles inside the frame and determine the threshold value (i.e., threshold=15); if the number of vehicles is greater than or equal the value of the threshold, then the frame's status is traffic congestion, otherwise the frame's status is non-traffic.
5- Model evaluation performance was weighed using seven confusion matrix metrics: accuracy, sensitivity, specificity, F1 score, FPR, FNR, and the error.
6- The experimental results using the Haar cascade classifier are an average accuracy of 79.41%. The experimental results using Deep learning are an average of 97.11%.

7- Finally, the researcher prove that the Deep learning techniques have excellent performance in vehicle detection more than Haar cascade techniques. The proposed method improves performance, saves time, and high accuracy.

5.2 FUTURE WORKS AND RECOMMENDATIONS

Due to the problems it causes and impedes the traffic movement in various cities, traffic congestion is receiving significant attention from researchers. The researcher solved overcrowding using the Haar cascade classifier and Deep learning classifier, instead of the traditional methods used that require time and cost. The proposed method showed acceptable results compared with other techniques offered by researchers in recent years. The proposed work included immediate processing of the road status provisions without any delay. In future studies, the researcher will focus on new methods using more accurate classifiers and try to combine algorithms of machine learning with Internet of things (IoT) to reach the best results at the lowest cost.

6. REFERENCES

[1] F. Knorr, D. Baselt, M. Schreckenberg, and M. Mauve, "Reducing traffic jams via VANETs," *IEEE Transactions on Vehicular Technology,* vol. 61, no. 8, pp. 3490-3498, 2012.

[2] Z. Wang, M. Lu, X. Yuan, J. Zhang, and H. Van De Wetering, "Visual traffic jam analysis based on trajectory data," *IEEE transactions on visualization and computer graphics,* vol. 19, no. 12, pp. 2159-2168, 2013.

[3] F. Matin, G. M. Herani, and U. A. Warraich, "Factors affecting traffic jam in Karachi and its Impact on performance of economy," *KASBIT Journal of Management & Social Science,* vol. 5, pp. 25-32, 2012.

[4] A. Ruta, F. Porikli, S. Watanabe, and Y. Li, "In-vehicle camera traffic sign detection and recognition," *Machine Vision and Applications,* vol. 22, no. 2, pp. 359-375, 2011.

[5] L. Wei and D. Hong-ying, "Real-time road congestion detection based on image texture analysis," *Procedia engineering,* vol. 137, pp. 196-201, 2016.

[6] S. Hu, J. Wu, and L. Xu, "Real-time traffic congestion detection based on video analysis," *Journal of Information and Computational Science,* vol. 9, no. 10, pp. 2907-2914, 2012.

[7] M. Molina, J. HERN, and J. É. CUENA, "A structure of problem-solving methods for real-time decision support in traffic control," *International Journal of Human-Computer Studies,* vol. 49, no. 4, pp. 577-600, 1998.

[8] A. M. de Souza, R. S. Yokoyama, G. Maia, A. Loureiro, and L. Villas, "Real-time path planning to prevent traffic jam through an intelligent transportation system," in *2016 IEEE Symposium on Computers and Communication (ISCC)*, 2016: IEEE, pp. 726-731.

[9] A. M. de Souza, R. S. Yokoyama, L. C. Botega, R. I. Meneguette, and L. A. Villas, "Scorpion: A solution using cooperative rerouting to prevent congestion and improve traffic condition," in *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, 2015: IEEE, pp. 497-503.

[10] G. V. Paul, G. J. Beach, C. J. Cohen, and C. J. Jacobus, "Realtime object tracking system," ed: Google Patents, 2010.

[11] K. Huang, L. Wang, T. Tan, and S. Maybank, "A real-time object detecting and tracking system for outdoor night surveillance," *Pattern Recognition,* vol. 41, no. 1, pp. 432-444, 2008.

[12] H. Naeem, J. Ahmad, and M. Tayyab, "Real-time object detection and tracking," in *INMIC*, 2013: IEEE, pp. 148-153.

[13] J. J. Uebbing, "Motion detecting camera system," ed: Google Patents, 2010.

[14] L. Zhao and C. Thorpe, "Qualitative and quantitative car tracking from a range image sequence," in *Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No. 98CB36231)*, 1998: IEEE, pp. 496-501.

[15] D. Koller, J. Weber, and J. Malik, "Robust multiple car tracking with occlusion reasoning," in *European Conference on Computer Vision*, 1994: Springer, pp. 189-196.

[16] M. Betke, E. Haritaoglu, and L. S. Davis, "Multiple vehicle detection and tracking in hard real-time," in *Proceedings of Conference on Intelligent Vehicles*, 1996: IEEE, pp. 351-356.

[17] R. R. Barton, "Vehicle tracking system," ed: Google Patents, 1986.

[18] X. Li, Y. She, G. Yang, Y. Zhao, and D. Luo, "A traffic congestion detection method for surveillance videos based on macro optical flow velocity," in *ICCTP 2011: Towards Sustainable Transportation Systems*, 2011, pp. 1569-1578.

[19] R. Bauza and J. Gozálvez, "Traffic congestion detection in large-scale scenarios using vehicle-to-vehicle communications," *Journal of Network and Computer Applications,* vol. 36, no. 5, pp. 1295-1307, 2013.

[20] H. Jianming, M. Qiang, W. Qi, Z. Jiajie, and Z. Yi, "Traffic congestion identification based on image processing," *IET Intelligent Transport Systems,* vol. 6, no. 2, pp. 153-160, 2012.

[21] J.-P. Lin and M.-T. Sun, "A YOLO-based traffic counting system," in *2018 Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, 2018: IEEE, pp. 82-85.

[22] G. Palubinskas, F. Kurz, and P. Reinartz, "Detection of traffic congestion in optical remote sensing imagery," in *IGARSS 2008-2008 IEEE International Geoscience and Remote Sensing Symposium*, 2008, vol. 2: IEEE, pp. II-426-II-429.

[23] U. Nagaraj, J. Rathod, P. Patil, S. Thakur, and U. Sharma, "Traffic jam detection using image processing," *International Journal of Engineering Research and Applications,* vol. 3, no. 2, pp. 1087-1091, 2013.

[24] S. Soo, "Object detection using Haar-cascade Classifier," *Institute of Computer Science, University of Tartu,* pp. 1-12, 2014.

[25] A. P. Jana and A. Biswas, "YOLO based Detection and Classification of Objects in video records," in *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, 2018: IEEE, pp. 2448-2452.

[26] C. I. Patel and R. Patel, "Contour based object tracking," *International Journal of Computer and Electrical Engineering,* vol. 4, no. 4, p. 525, 2012.

[27] Z. Chen, T. Ellis, and S. A. Velastin, "Vehicle detection, tracking and classification in urban traffic," in *2012 15th International IEEE Conference on Intelligent Transportation Systems*, 2012: IEEE, pp. 951-956.

[28] T. Lindeberg, "Scale invariant feature transform," 2012.

[29] K. Mu, F. Hui, and X. Zhao, "Multiple Vehicle Detection and Tracking in Highway Traffic Surveillance Video Based on SIFT Feature Matching," *Journal of Information Processing Systems,* vol. 12, no. 2, 2016.

[30] Y. Liu, Y. Lu, Q. Shi, and J. Ding, "Optical flow based urban road vehicle tracking," in *2013 ninth international conference on computational intelligence and security*, 2013: IEEE, pp. 391-395.

[31] W. Wang, W. Zhang, and X. Yang, "Traffic micro-simulation of urban road network," in *ICTIS 2011: Multimodal Approach to Sustained Transportation System Development: Information, Technology, Implementation*, 2011, pp. 1042-1052.

[32] K. Mandal, A. Sen, A. Chakraborty, S. Roy, S. Batabyal, and S. Bandyopadhyay, "Road traffic congestion monitoring and measurement using active RFID and GSM technology," in *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2011: IEEE, pp. 1375-1379.

[33] M. Arora and V. Banga, "Real time traffic light control system using morphological edge detection and fuzzy logic," in *2nd International Conference on Electrical, Electronics and Civil Engineering (ICEECE'2012) Singapore April*, 2012, pp. 28-29.

[34] F. Soylemezgiller, M. Kuscu, and D. Kilinc, "A traffic congestion avoidance algorithm with dynamic road pricing for smart cities," in *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2013: IEEE, pp. 2571-2575.

[35] K. Kanistras, G. Martins, M. J. Rutherford, and K. P. Valavanis, "A survey of unmanned aerial vehicles (UAVs) for traffic monitoring," in *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2013: IEEE, pp. 221-234.

[36] M. Milojevic and V. Rakocevic, "Distributed road traffic congestion quantification using cooperative VANETs," in *2014 13th annual Mediterranean Ad Hoc networking workshop (MED-HOC-NET)*, 2014: IEEE, pp. 203-210.

[37] A. Nidhal, U. K. Ngah, and W. Ismail, "Real time traffic congestion detection system," in *2014 5th International Conference on Intelligent and Advanced Systems (ICIAS)*, 2014: IEEE, pp. 1-5.

[38] E. D'Andrea, P. Ducange, B. Lazzerini, and F. Marcelloni, "Real-time detection of traffic from twitter stream analysis," *IEEE transactions on intelligent transportation systems,* vol. 16, no. 4, pp. 2269-2283, 2015.

[39] I. Ali, A. Malik, W. Ahmed, and S. A. Khan, "Real-time vehicle recognition and improved traffic congestion resolution," in *2015 13th International Conference on Frontiers of Information Technology (FIT)*, 2015: IEEE, pp. 228-233.

[40] S. Choudhury, S. P. Chattopadhyay, and T. K. Hazra, "Vehicle detection and counting using haar feature-based classifier," in *2017 8th annual industrial automation and electromechanical engineering conference (IEMECON)*, 2017: IEEE, pp. 106-109.

[41] C.-T. Lam, H. Gao, and B. Ng, "A real-time traffic congestion detection system using on-line images," in *2017 IEEE 17th International Conference on Communication Technology (ICCT)*, 2017: IEEE, pp. 1548-1552.

[42] R. Ayala Somayajula, "Real time traffic congestion detection using images," 2018.

[43] A. Ćorović, V. Ilić, S. Durić, M. Marijan, and B. Pavković, "The real-time detection of traffic participants using yolo algorithm," in *2018 26th Telecommunications Forum (TELFOR)*, 2018: IEEE, pp. 1-4.

[44] P. Chakraborty, Y. O. Adu-Gyamfi, S. Poddar, V. Ahsani, A. Sharma, and S. Sarkar, "Traffic congestion detection from camera images using deep convolution neural networks," *Transportation Research Record,* vol. 2672, no. 45, pp. 222-231, 2018.

[45] C.-T. Lam, B. Ng, and C.-W. Chan, "Real-Time Traffic Status Detection from on-Line Images Using Generic Object Detection System with Deep Learning," in *2019 IEEE 19th International Conference on Communication Technology (ICCT)*, 2019: IEEE, pp. 1506-1510.

[46] E. D. Pisano *et al.*, "Contrast limited adaptive histogram equalization image processing to improve the detection of simulated spiculations in dense mammograms," *Journal of Digital imaging,* vol. 11, no. 4, p. 193, 1998.

[47] M. Freeman, *Digital SLR handbook*. Ilex, 2011.

[48] K. Kumar, J.-P. Li, and R. A. Shaikh, "Content based image retrieval using gray scale weighted average method," *Int. J. Adv. Comput. Sci. Appl,* vol. 7, no. 1, pp. 1-6, 2016.

[49] J. A. Hussein, "Spatial domain watermarking scheme for colored images based on log-average luminance," *arXiv preprint arXiv:1001.3496,* 2010.

[50] D. R. Bull, *Communicating pictures: A course in Image and Video Coding*. Academic Press, 2014.

[51] T. McReynolds and D. Blythe, *Advanced graphics programming using OpenGL*. Elsevier, 2005.

[52] P. Garg and T. Jain, "A comparative study on histogram equalization and cumulative histogram equalization," *International Journal of New Technology and Research,* vol. 3, no. 9, 2017.

[53] J.-Y. Kim, L.-S. Kim, and S.-H. Hwang, "An advanced contrast enhancement using partially overlapped sub-block histogram equalization," *IEEE transactions on circuits and systems for video technology,* vol. 11, no. 4, pp. 475-484, 2001.

[54] S. Milani, R. Bernardini, and R. Rinaldo, "Adaptive denoising filtering for object detection applications," in *2012 19th IEEE International Conference on Image Processing*, 2012: IEEE, pp. 1013-1016.

[55] V. V. Khryashchev, A. L. Priorov, I. V. Apalkov, and P. S. Zvonarev, "Image denoising using adaptive switching median filter," in *IEEE International Conference on Image Processing 2005*, 2005, vol. 1: IEEE, pp. I-117.

[56] D. G. Bailey, *Design for embedded image processing on FPGAs*. John Wiley & Sons, 2011.

[57] J.-S. Lee, "Digital image smoothing and the sigma filter," *Computer vision, graphics, and image processing,* vol. 24, no. 2, pp. 255-269, 1983.

[58] B. Justusson, "Median filtering: Statistical properties," in *Two-Dimensional Digital Signal Prcessing II*: Springer, 1981, pp. 161-196.

[59] M. Kirchner and J. Fridrich, "On detection of median filtering in digital images," in *Media forensics and security II*, 2010, vol. 7541: International Society for Optics and Photonics, p. 754110.

[60] S. King, "A beginners' guide to statistical parametric speech synthesis," *The Centre for Speech Technology Research, University of Edinburgh, UK,* 2010.

[61] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, 2001, vol. 1: IEEE, pp. I-I.

[62] S. Guennouni, A. Ahaitouf, and A. Mansouri, "A comparative study of multiple object detection using Haar-like feature selection and local binary patterns in several platforms," *Modelling and Simulation in Engineering,* vol. 2015, 2015.

[63] P. Viola and M. J. Jones, "Robust real-time face detection," *International journal of computer vision,* vol. 57, no. 2, pp. 137-154, 2004.

[64] C. Zhang and Z. Zhang, "A survey of recent advances in face detection," 2010.

[65] S. Abirami, G. Kousalya, and R. Karthick, "Varied expression analysis of children with asd using multimodal deep learning technique," in *Deep Learning and Parallel Computing Environment for Bioengineering Systems*: Elsevier, 2019, pp. 225-243.

[66] J. Zhu and Z. Chen, "Real time face detection system using adaboost and haar-like features," in *2015 2nd International Conference on Information Science and Control Engineering*, 2015: IEEE, pp. 404-407.

[67] W. Berger, "Deep Learning Haar Cascade Explained," *línea]. Available: http://www. willberger. org/cascade-haar-explained/.[Último acceso: 12 08 2019],* 2018.

[68] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks,* vol. 61, pp. 85-117, 2015.

[69] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE transactions on neural networks and learning systems,* vol. 30, no. 11, pp. 3212-3232, 2019.

[70] H. Song, H. Liang, H. Li, Z. Dai, and X. Yun, "Vision-based vehicle detection and counting system using deep learning in highway scenes," *European Transport Research Review,* vol. 11, no. 1, p. 51, 2019.

[71] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91-99.

[72] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in neural information processing systems*, 2016, pp. 379-387.

[73] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117-2125.

[74] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263-7271.

[75] W. Liu *et al.*, "Ssd: Single shot multibox detector," in *European conference on computer vision*, 2016: Springer, pp. 21-37.

[76] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980-2988.

[77] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580-587.

[78] H. Nguyen, "Improving Faster R-CNN Framework for Fast Vehicle Detection," *Mathematical Problems in Engineering,* vol. 2019, 2019.

[79] A. A. Yilmaz, M. S. Guzel, I. Askerbeyli, and E. Bostanci, "A vehicle detection approach using deep learning methodologies," *arXiv preprint arXiv:1804.00429,* 2018.

[80] L. Suhao, L. Jinzhao, L. Guoquan, B. Tong, W. Huiqian, and P. Yu, "Vehicle type detection based on deep learning in traffic scene," *Procedia computer science,* vol. 131, pp. 564-572, 2018.

[81] D.-s. Kim, K. W. Figueroa, K.-W. Li, A. Boroujerdi, T. Yolo, and Z. D. Luo, "Profiling of dynamically changed gene expression in dorsal root ganglia post peripheral nerve injury and a critical role of injury-induced glial fibrillary acetic protein in maintenance of pain behaviors," *Pain,* vol. 143, no. 1-2, pp. 114-122, 2009.

[82] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779-788.

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

[84] S. Nayak, "Deep learning based object detection using yolov3 with opencv(python/ c++), Aug 2018," *URL: https://www. learnopencv. com/deep-learning-based-object-detection-using-yolov3-with-opencv-python-c/[cited 01.08. 2019]*.

[85] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision,* vol. 88, no. 2, pp. 303-338, 2010.

[86] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research,* vol. 32, no. 11, pp. 1231-1237, 2013.

[87] S. Haghighi, M. Jasemi, S. Hessabi, and A. Zolanvari, "PyCM: Multiclass confusion matrix library in Python," *Journal of Open Source Software,* vol. 3, no. 25, p. 729, 2018.