

AN EFFICIENT ALGORITHM FOR TRAFFIC CONGESTION CONTROL

Submitted in partial fulfillment of the
requirements for the award of
Bachelor of Engineering degree in Computer Science and
Engineering by

DUNDIGALLA ACHYUTH KUMAR

(37110202)

&

GALI AMRUTH SHOW REDDY

(37110217)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI
SALAI, CHENNAI – 600 119**

APRIL - 2021



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with "A" grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119
www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of **Dundigalla Achyuth Kumar** (37110202) & **Gali Amruth Show Reddy** (37110217) who carried out the project entitled "**An Efficient Algorithm For Traffic Congestion Control**" under our supervision from November, 2020 to April 2021.

Internal Guide

Dr. A. Mary Posonia, M.E., Ph.D.,

Head of the Department

Dr. S. VIGNESHWARI, M.E., Ph.D.,
Dr.L.LAKSHMANAN, M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

We, **Dundigalla Achyuth Kumar and Gali Amruth Show Reddy** hereby declare that the Project Report entitled “**An Efficient Algorithm For Traffic Congestion Control**” is done by us under the guidance of Associate Professor **Dr. A. Mary Posonia, M.E., Ph.D.**, of the Department of CSE at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATES

1.

2.

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D.**, Dean, School of Computing, **Dr. S. Vigneshwari, M.E., Ph.D.**, and **Dr. L. Lakshmanan, M.E., Ph.D.**, Heads of the Department, Dept. of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. A. Mary Posonia, M.E., Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of who were helpful in many ways for the completion of the project.

ABSTRACT

Recently, car parking space is one of the hard tasks in modern-day traits. area operation and supervision of vehicle area is now a demandable area of study. searching for an unfilled parking space in overfilled visitors is a time-consuming manner. the prevailing to be had parking area popularity techniques are not robust or worldwide for photos took from one-of-a-kind photographic digicam lookouts. final results of a right parking space in a hectic town is simply a stimulating trouble and public are facing this elaborate on a every day base. the most reason of this observe is to extensively speak the previous research of vehicle parking area recognition and equivalence them from modified factors. to conquer these problems, we suggest a aircraft-primarily based method which adopts an operational three-D parking lot characteristic such as abundant planar aspects. The aircraft-based totally three-D segment model performs a key element in conduct inter-item occlusion and point of view alteration. however, to relieve the interference of unpredictable illumination deviations and sun shades, we endorse a aircraft-based class system. furthermore, by using supplying a Bayesian categorized framework to participate the three-D prototypical with the aircraft-based organisation technique, we methodically infer the space popularity. final, to triumph over the insufficient lighting fixtures within the night-time, we additionally gift a pre-processing step to improve picture excellence. The investigational consequences expression that the deliberate framework can comprehend strong locating of car parking places in each sunlight hours and night-time.

TABLE OF CONTENTS :

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	v
1	INTRODUCTION	
	1.1 TRAFFIC CONGESTION	2
	1.2 ALGORITHM FOR TRAFFIC CONGESTION	3
	1.2.1 SMART TRAFFIC LIGHT MANAGEMENT SYSTEM	3
	1.2.2 LAW ENFORCEMENT BY RFID READER	3
	1.3 SMART TRAFFIC LIGHT MANAGEMENT SYSTEM	5
	1.3.1 CAMERA PLACING CALCULATION	5
	1.3.2 TRAFFIC LIGHT CONTROLLING MECHANISM	5
	1.4 TRAFFIC LIGHT CONTROLLING MECHANISAM	6
	1.5 LAW ENFORCEMENT BY RFID READER	7
	1.6 BACKGROUND AND CONTEXT	9
	1.7 IMAGE PROCESSING	12
	1.8 CONVERTING TO GRAY SCALE	13
	1.9 IMAGE PROCESSING TECHNIQUES	14
2	AIM AND SCOPE	
	2.1 EXISTING SYSTEM	19
	2.2 PROPOSED SYSTEM	19
	2.3 LITERATURE REVIEW	20
3	REQUIREMENT SPECIFICATIONS	
	3.1 HARD REQUIREMENTS	27
	3.2 SOFTWARE REQUIREMENTS	27
	3.3 MATLAB	27
4	DESIGN METHDOLOGY	
	4.1 MODULES DESCRIPTIONS	32

	VIDEO TP FRAMES CONVERSION	32
	SEGMENTATION	32
	BACKGROUND SUBTRACTION	33
5	RESULTS AND DISCUSSIONS	
	NAMING CONVENTIONS	35
	TEST PROCEDURE	37
	TEST DATA AND OUTPUT	37
	WHITE BOX TESTING	40
	BLACK BOX TESTING	40
6	CONCLUSIONS	44
	REFERENCES	45
	APPENDIX	
	A.SOURCE CODE	47
	B.SCREEN SHOTS	56

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NUMBER
1.1	BLOCK DIAGRAM OF THE PROPOSED MODEL	2
1.2	APPROPRIATE HEIGHT FOR CAMERA PLACEMENT	3
1.2.1	CAR AND VEHICLE DETECTION	4
1.2.2	CAR DETECTION PROCESS	5
1.5	RFID READER DETECTION	8
1.7	CONVERSION COLOUR TO GRAY SCALE IMAGE	13
1.7	IMAGE REPRESENTATION	14
4	BLOCK DIAGRAM	32

CHAPTER 1

INTRODUCTION

Traffic light posts are positioned at road intersections and pedestrian crossings. Traffic light posts blink the light Signals after a certain time period which is not a complete systematic system as it cannot solve the traffic problems fully. Thus traffic jams take place. Lack of trained traffic police officers and old manual traffic light control system made this problem worse in many cities like Dhaka, Chittagong etc. Since the traffic vehicle pressure is not same at every road at the same time, Traffic lights should be controlled by an adaptive system which will detect the traffic conditions and use traffic light signals accordingly. Instead of using electronic sensors embedded in the road our system is based on Video processing which is a form of signal processing and for which the input is a video frame; the output of video processing may be either an image or a set of characteristics or parameters related to the image. Our system will detect vehicles through video frames instead of using any other Mechanism.

A camera will be installed alongside the traffic light. It will capture video sequences of traffic condition on road. This technique will analyze the videos from cameras and count the number of vehicles for each direction and also import to the controller. Then the controller estimates a period of time needed by each path to open and each traffic light to turn on or off based on the number of vehicles in a fixed sequence. It will also be used to monitor the traffic conditions. It can reduce the traffic congestion and avoid the time being wasted by a green light on an empty road. This system is more reliable to detect vehicle presence because it uses actual traffic condition images. This system is intended for country like Bangladesh where traffic policeman can take clever, critical decisions and handle emergencies but with the help of our automatic system they can use preset signal timings to control traffic at intersections because it provides more accurate information for signal decision making.

Traffic Congestion:

In our proposed model, there will be four cameras in one intersection for a four way road. A CPU will be connected with these cameras which will be responsible for video processing. The RFID will be placed under the road for detection of the car. The hardware's that we will be using are: HD Camera, CPU (For video processing), Microprocessor (For traffic light controlling) and RFID Reader (For vehicle detection) which will be beneath the road. According to Figure 1 a High definition camera placed on poles will observe the vehicular traffic flow continuously on a road. Then using frame by frame Real time video analysis through our developed algorithm, we can detect how much cars are present on the road. Depending on the number of detected vehicles we have developed and implemented a sequential traffic timer system. Microcontroller will detect the signal from CPU and start the sequential traffic light. While the light phase goes from green to red, our microcontroller or arduino will send a signal to CPU and CPU will energize the RFID reader. RFID reader will detect the car which already has a RFID tag. This information will be transmitted to CPU or the central database. Thus our system will detect the law breakers who move regardless of the red light. According to this automatic traffic system, the traffic light ON/OFF will depend on the number of vehicles on the road. The HD camera will be installed in the traffic light post at a height of 19-25 feet above the road which is illustrated in Figure 2.

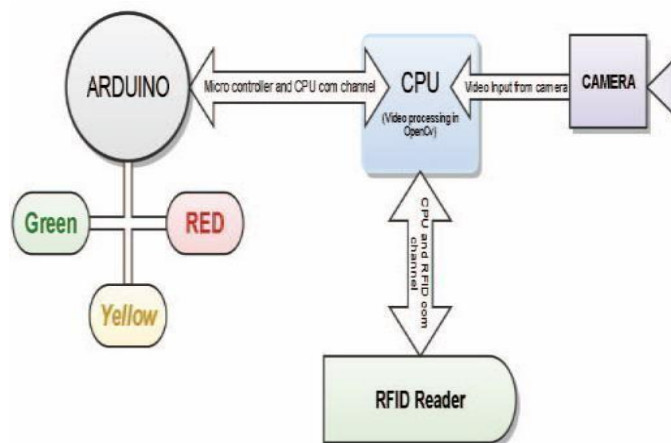


Fig 1.1 Block diagram of the proposed model

This camera will take the live video footage of the road and send it to a computer where video analysis will be done. For a 4 way intersection, CPU will detect each and every car and will count the vehicle number in the road by using our developed algorithm. It will also do the same thing with other road by using another camera. CPU then compares vehicle number of both roads. The road which has more vehicles will get the preference and green light for that road will be on and red signal will be shown automatically to the other road. Traffic lights will be connected to the computer and intelligent system will control the traffic light system. Our project is divided into three parts:

Algorithm for Traffic Detection

Smart Traffic Light Management System

Law Enforcement by RFID Reader

These parts will be discussed in the following sections.

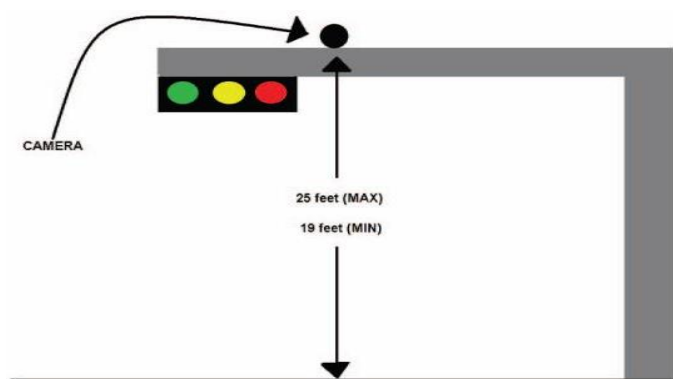


Fig 1.2 Appropriate Height for Camera placement

Algorithm for Traffic Detection

Car detection by video processing is the most important process to implement our project. Video processing will be done by using Open CV (OC) instead of MATLAB for its faster processing. Installed camera will send the output to the main server computer that will analysis that video and give its after analyzing result to the microcontroller. The vehicles are detected with the help of OC and camera.

Here first the vehicles are detected by some car models in xml file. Figure 4 depicted the result of the code. By using Cascade Classifier we have created haar cascade. It was trained in that xml file with some rickshaw model. Then we have subtracted the background and the shadow by Background Subtraction (BS) with the help of BGS library. Another technique that we have used is Blob detection for better detection. We have filtered the video frames by area, circularity, convexity and inertia.

There can be limit of counting vehicles for a segment of road and that will be given by the user. When the given amount of vehicles are detected which are standing on a jam, it prompts a message to the user. Then it changes the camera and works in the same way for other segments of the road. Finally it makes decision after calculating the jam situation. 4 cameras will be installed in every lane in 4 lane intersection. All 4 cameras will send its data to the main computer and computer will send it to microcontroller. The microcontroller will follow some mechanism to decide which traffic light will be on/off in the road. The mechanism is described in later part of this paper.



Fig 1.2.1 Car and Vehicle Detection

Car detection procedures and steps are shown bellow in the flowchart.

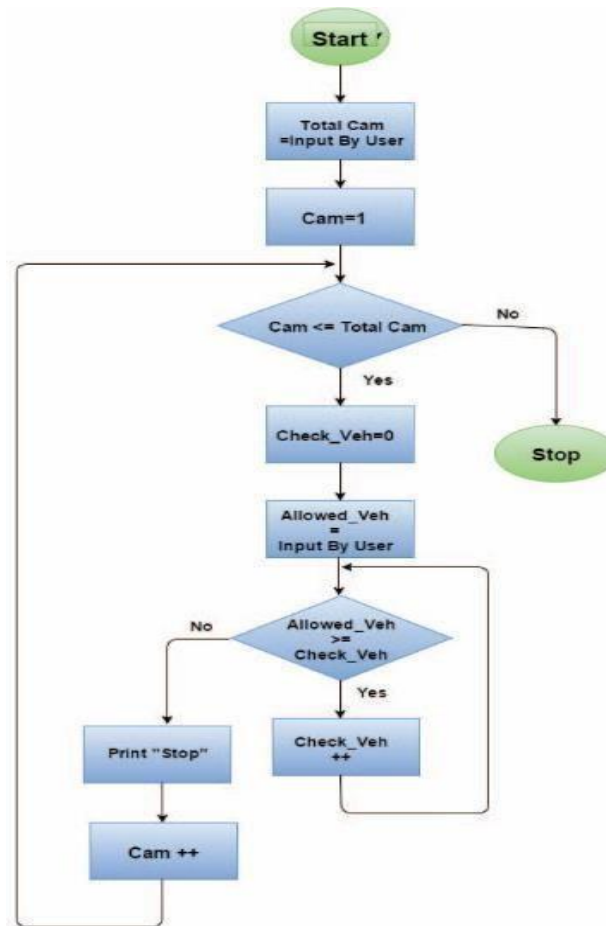


Fig 1.2.2 Flowchart of Car Detection Process.

Smart Traffic Light Management System

Smart traffic light system working procedure can be divided into two parts.

Camera Placing Calculation

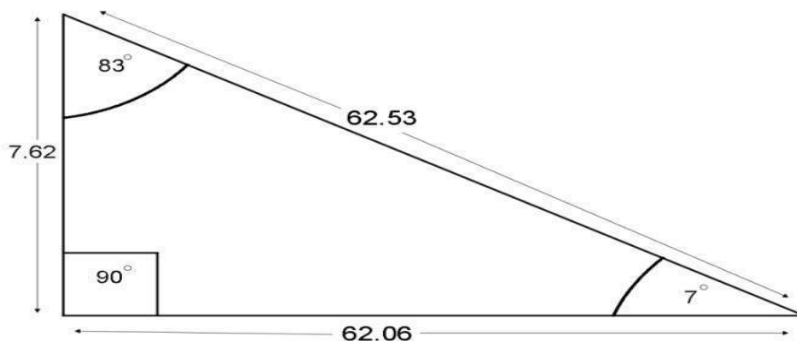
Traffic Light Controlling Mechanism

Camera Placing Calculation:

Car detection is the primary work of the system. For detecting car or any other vehicle, Camera placement is very important for better vehicle detection and accuracy.

Perfect height and angle for camera yield high accuracy for car detection, by ensuring the most area coverage. The higher the camera coverage is, the better decision this system will make. According to proposed project, we will set up a camera in the Light post. The height has to be in a certain range so that the software could detect car and detect as many car as possible in a road. After Taking samples from various heights, we calculated that when the camera's height is in range between 19 feet to 25 it gives us the best result. So camera's height should not exceed 25 feet otherwise there will be problem to detect car for the software. Any kind of obstacle must not come in front of camera so; it should be placed in such region where clear line of sight is available.

Our goal is to get more than 40 cars in the frame. By this arrangement we can get coverage of 62.06 meter. Now we need to calculate how many cars could be possible to detect in that road within our coverage. Average length of a sedan car is approximately 4.5 meter; so,



Camera placing calculation

Traffic light controlling mechanism:

As we are working with a four way intersection, for the time being, we will break this down for just one road only. For each road we will check twice for traffic. We have shown before that we can cover almost 50 cars (48.2 cars to be exact) with our camera for a specific road. So we will put a threshold value of 48 cars, means if our video processor could detect 48 cars then the OC will send a specific string in

our com port of

arduino. Microcontroller will keep checking if there is the string or not. This will lead us to two situations. If arduino could detect the string in its com port it will consider the road is crowded which is situation 1. If arduino could not detect the string in its com port then that will be situation 2 which indicates the road is not so crowded. For situation 1 (Table 1) (When a road reaches its car threshold value) we will turn the green light on this light will be turned on for 60 seconds. After passing the Assigned 60 second the OC will check again for the traffic. For situation 2 (When a road is not near its car threshold value) we will turn the green light as well, but for shorter period of time which is 30 seconds. After passing assigned 30 second the OC will check again for the traffic. This whole process will be repeated by the system for the same road again. So in total a road will get 90 (60+30) second at most and 30 second at least. Car threshold can be changed according to traffic conditions. Traffic situation is also not same all the time. Traffic pressure is very much high during rush hours or any other emergency period but it is much less at late night. So interval between lights and threshold car value will be changed according to time. Threshold value will be higher during traffic congestion and less during late night.

Law Enforcement by RFID Reader

This system will be built to detect and identify any vehicle which breaks law specially traffic signal law. RFID Tags are intelligent bar code that contains some information which can be read by using RFID reader. It exchanges the information by using electromagnetic field to transfer data for the purposes of automatically identifying and tracking tags attached to objects.

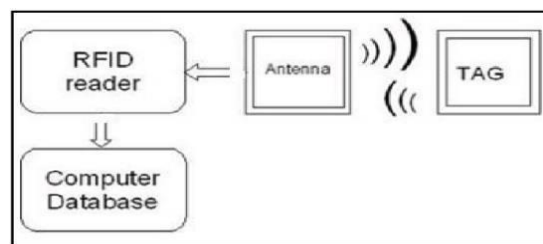
The chip typically is capable of carrying 2,000 bytes of data or less [5]. A RFID reader has basically three parts. They are:

- Antenna.
- A transceiver with a decoder to interpret the data
- The RFID tag - that has been programmed with information.
- Computer which will show the information.

RFID (Radio Frequency Identification) tags are already installed in almost every vehicles number plate in Dhaka this contains the basic information of the owner of the vehicle. For every car there will be a unique passive RFID. This passive RFID

will be

energized by the reader. In Fig. 6 we have tried to illustrate the system in a block diagram. According to this system a RFID reader will be placed in the joining point of the road junction where a car must stopped when the traffic signal shows the Red signal. During the red signal this RFID reader will be active and other time it will be deactivated. During its active time if any car breaks the law and move despite red signal, those cars will have to cross the RFID reader which will be placed beneath the road and RFID [Fig. 7] will then detect the RFID tag of the car. It will read the stored information of the tag which will be used to detect the owner and penalized him according to law. RFID reader detection zone area radius should not exceed more than 1 meter otherwise it could detect legally parked car also. It will be better if the radius of the RFID detection zone is lengthwise. So according to these conditions of this project, it will be best if we use High frequency (HF) RFID. More than one RFID can be placed beneath the road for wide road.



Block Diagram of Vehicle detection Control System

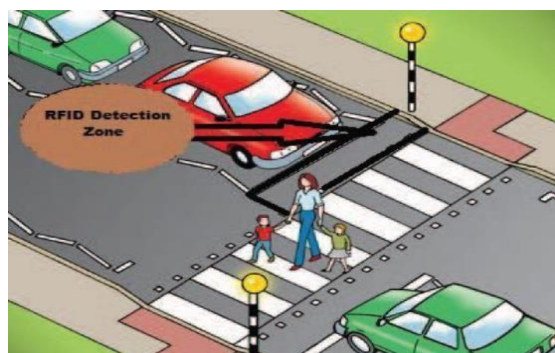


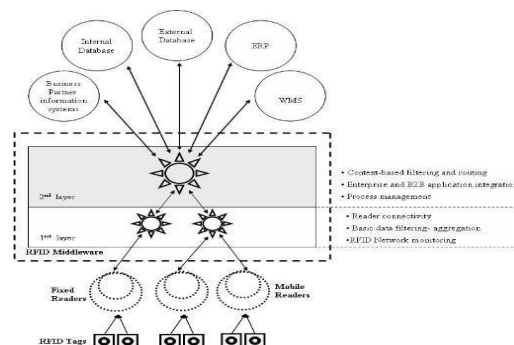
Fig 1.5 RFID Reader Detection

Background and context

The middleware as a key component of the RFID system

RFID technology is classified as a wireless automatic identification and data capture (AIDC). A basic RFID system is composed of a tag containing a microprocessor, a reader and its antennas, and a computer equipped with a middleware program, in which business rules are configured (Asif and Mandviwalla, 2005). The tag generally attached to a product communicates through radio frequencies with the reader's antennas. The reader sends the location and unique identification of the product to a computer. Based on preconfigured rules, the middleware can adjust or initiate business processes automatically.

RFID middleware is considered as one essential intelligence- added component of any RFID system and could be linked with other firm information systems (see Figure 1). More precisely, the middleware consists of the operating system, the data repository, and the processing algorithms that convert multiple tag inputs into visible tracking or identification data. The middleware could provide the following: (i) multiple reader configuration, control and monitoring; (ii) access coordination to an environment with multiple applications; (iii) receiving events generated by a reader; (iv) data filtering, smoothing and aggregating; (v) data routing to enterprise applications such as a Warehouse Management System (WMS), Enterprise Resource Planning (ERP), Transport Management System (TMS) or a Manufacturing Execution System (MES) (Floerkemeier and Lampe, 2005; Nurminen, 2006).



Positioning RFID middleware in an RFID system (www.forrester.com)

Driven by the rising number of RFID applications and by mandates from important players such as Wal-Mart, the US Department of Defense, Tesco, Target and Metro AG and from their own major suppliers, the market for RFID middleware has grown rapidly in recent years. O'Connor (2006) estimates that the global market for RFID middleware had expanded from 2004 to 2005 by almost 162%. Although the market for RFID middleware is estimated to reach for \$135 million in 2007 or roughly 3% of RFID systems revenues, it is projected to grow to almost \$1,557.5 million in 2011 (Winter Green Research, 2005). The RFID middleware offering is characterized by the presence of numerous firms, ranging from (i) pure-play vendors such as Globe Ranger, and OAT Systems, (ii) application vendors such as Manhattan Associates, Red Prairie, Oracle and SAP, (iii) integration specialists such as TIBCO Software and Ascential Software, to (iv) platform giants like Sun Microsystems, IBM, Oracle, and Microsoft (O'Connor, 2007).

Some of the major obstacles to widespread RFID adoption tend to disappear. First, the costs of an RFID middleware have decreased drastically, moving from about \$125,000 per installed site a few years ago to \$5,000 to \$20,000 today (Nurminen, 2006). This downward trend positions RFID technology in a more affordable spectrum of potential investments and will eventually facilitate its adoption. Second, the current correct tags reading has reached 100% at the pallet level (Paxar, 2005) and is constantly improving at the item level. Third, chip technology has improved to a point where tags have become much more affordable. The concerns are now more focused on middleware issues such as the capacity needs in order to manage large number of readers and the huge amount of data these readers generate (Floerkemeier and Lampe, 2005). In addition, the next trends in the RFID middleware will be more and more integrated with other technologies. For example, the integration with wireless technologies, such as Local Area Network (LAN), General Packet Radio Service (GPRS) and Global Positioning System (GPS) could enable real time tracking and tracing (LogicaCMG, 2004).

In all cases, "installing the software is only the start" (G X S, p. 11, 2005). Indeed, applications at the process level as well as the technical level need to be

configured in the RFID middleware in order to trigger appropriate actions when specific events occur. Finally, the increasing concerns from supply chain members with the accuracy of

inventory data and validity of RFID tagged items (O'Connor, 2004) can be effectively addressed by the RFID middleware which acts as a bridge between the “physical RFID world” and the “software RFID infrastructure”. The middleware thus allows the automatic interpretation and the semantic transformation of observations generated from the automatic data collection on tagged items into business logic data prior to their integration into existing information systems (e.g. ERP, WMS) (Fusheng and Peiya, 2005).

Previous work on the role of the middleware in RFID applications within a supply chain context is still scarce but represents an emerging and fast growing area of research. Among the recent conceptual papers, Gunasekaran and Ngai (2005) suggest that RFID technology may facilitate the development of supply chain configurations by acting as an enabler of a build to order (BTO) strategy. In the same line of thought, Pramataris et al. (2005) suggest that RFID technology may constitute a link to more collaborative approaches such as CPFR (Collaborative Planning, Forecasting and Replenishment). Kelepouris et al. (2007) also suggests that RFID technology can act as an enabler of traceability in the food supply chain. While exploring the impacts of RFID technology in a retail supply chain, Lefebvre et al. (2005) also identify the emergence of “intelligent processes” to support RFID enabled Business-to-Business electronic commerce applications. Finally, Loebbecke and Palmer (2006) also examine the results of a joint RFID pilot project conducted between Kaufhof Department Stores, a leading European retailer and Gerry Weber, a fashion merchandise manufacturer. Results from the pilot study reveals that data derived from RFID technology about “products, processes, product movement, and even customer behavior can be used for proprietary and distinctive capabilities to gain competitive advantage, if turned into understandable and usable “content”. The above mentioned work points to the overriding importance of middleware configuration and integration as a key aspect of RFID strategy.

INTRODUCTION TO IMAGE PROCESSING:

Image Processing is a method to enhance raw images received from cameras/sensors placed on satellites, space probes and aircrafts or pictures taken in normal day-to-day life for various applications. Various techniques have been developed in Image Processing during the last four to five decades. Most of the techniques are developed for enhancing images obtained from unmanned spacecrafts, space probes and military reconnaissance flights. Image Processing systems are becoming popular due to easy availability of powerful personnel computers, large size memory devices, graphics software's etc. Image processing is a physical process used to convert an image signal into a physical image. The image signal can be either digital or analog. The actual output itself can be an actual physical image or the characteristics of an image.

Image processing is photography. In this process, an image is captured or scans using a camera to create a digital or analog image. In order to produce a physical picture, the image is processed using the appropriate technology based on the input source type. In digital photography, the image is stored as a computer file. This file is translated using photographic software to generate an actual image. The colors, shading, and nuances are all captured at the time the photograph is taken the software translates this information into an image. When creating images using analog photography, the image is burned into a film using a chemical reaction triggered by controlled exposure to light. The image is processed in a darkroom, using special chemicals to create the actual image. This process is decreasing in popularity due to the opening of digital photography, which requires less effort and special training to product images. The field of digital imaging has created a whole range of new applications and tools that were previously impossible. Face recognition software, medical image processing and remote sensing are all possible due to the development of digital image processing. Specialized computer programs are used to enhance and correct images.

Binary Images: Binary images use only a single bit to represent each pixel. Since a bit can only exist in two states- ON or OFF; every pixel in a binary image must be one of two colors, usually black or white. This inability to represent intermediate

shades of gray is what limits their usefulness in dealing with photographic images.

CONVERTING COLOR TO GRAYSCALE

Conversion of a color image to grayscale is not unique; different weighting of the color channels effectively represents the effect of shooting black-and-white film with different-colored photographic filters on the cameras.

To convert any color to a grayscale representation of its luminance, first one must obtain the values of its red, green, and blue (RGB) primaries in linear intensity encoding, by gamma expansion. Then, add together 30% of the red value, 59% of the green value, and 11% of the blue value (these weights depend on the exact choice of the RGB primaries, but are typical). Regardless of the scale employed (0.0 to 1.0, 0 to 255, 0% to 100%, etc.) the resultant number is the desired linear luminance value; it typically needs to be gamma compressed to get back to a conventional grayscale representation.



Fig.1.7 converting color to grayscale image

This is not the method used to obtain the luma in the YUV and related color models, used in standard color TV and video systems as PAL and NTSC, as well as in the L^*a^*b color model. These systems directly compute a gamma-compressed luma as a linear combination of gamma-compressed primary intensities, rather than use linearization via gamma expansion and compression.

To convert a gray intensity value to RGB, simply set all the three primary color components red, green and blue to the gray value, correcting to a different gamma if necessary.

DIGITAL IMAGE PROCESSING TECHNIQUES

The various Image Processing techniques are:

- Image representation
- Image preprocessing
- Image enhancement
- Image restoration
- Image analysis
- Image reconstruction
- Image data compression

IMAGE REPRESENTATION

An image defined in the "real world" is considered to be a function of two real variables, for example, $f(x,y)$ with f as the amplitude (e.g. brightness) of the image at the *real* coordinate position (x,y) . The effect of digitization is shown in Figure.

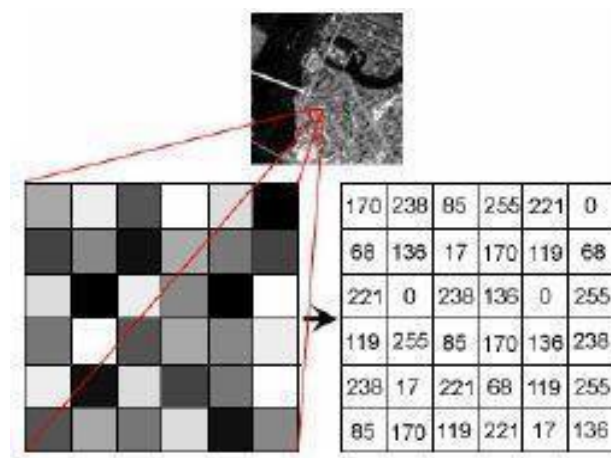


Fig.1.7 Image Representation

The 2D continuous image $f(x,y)$ is divided into N rows and M columns. The intersection of a row and a column is called as *pixel*. The value assigned to the integer coordinates $[m,n]$ with $\{m=0,1, 2,...,M-1\}$ and $\{n=0,1,2,...,N-1\}$ is $f[m,n]$. In fact, in most cases $f(x, y)$ which we might consider to be the physical signal that impinges on the face of a sensor. Typically an image file such as BMP, JPEG, TIFF etc., has some header and picture

information. A header usually includes details like format identifier (typically first information), resolution, number of bits/pixel, compression type, etc.

IMAGE ENHANCEMENT

Sometimes images obtained from satellites and conventional and digital cameras lack in contrast and brightness because of the limitations of imaging sub systems and illumination conditions while capturing image. Images may have different types of noise. In image enhancement, the goal is to accentuate certain image features for subsequent analysis or for image display. Examples include contrast and edge enhancement, pseudo-coloring, noise filtering, sharpening, and magnifying. Image enhancement is useful in feature extraction, image analysis and an image display. The enhancement process itself does not increase the inherent information content in the data. It simply emphasizes certain specified image characteristics. Enhancement algorithms are generally interactive and application-dependent.

IMAGE SEGMENTATION:

Image segmentation is the process that subdivides an image into its constituent parts or objects. The level to which this subdivision is carried out depends on the problem being solved, i.e., the segmentation should stop when the objects of interest in an application have been isolated e.g., in autonomous air-to-ground target acquisition, suppose our interest lies in identifying vehicles on a road, the first step is to segment the road from the image and then to segment the contents of the road down to potential vehicles.

FUNDAMENTAL IN DIGITAL IMAGE PROCESSING

The various basic steps are as follows

- Image Acquisition

- Image preprocessing
- Image segmentation
- Image Representation and Description
- Image Recognition and Interpretation and
- Knowledge base.

IMAGE ACQUISITION

In the process, the first step in the process is image acquisition that is, to acquire a digital image. It highly requires an imaging sensor and the capability to digitize the signal produced by the sensor. The sensor could be a monochrome or color TV camera that produces an entire image of the problem domain every 1/30 sec. The imaging sensor could also be a line-scan camera that produces a single image line at a time. In this case, the object's motion past the line scanner produces two-dimensional image. If the output of the camera or other imaging sensor is not already in digital form, an analog-to-digital converter digitizes it. The nature of the sensor and image it produces are determined by the application.

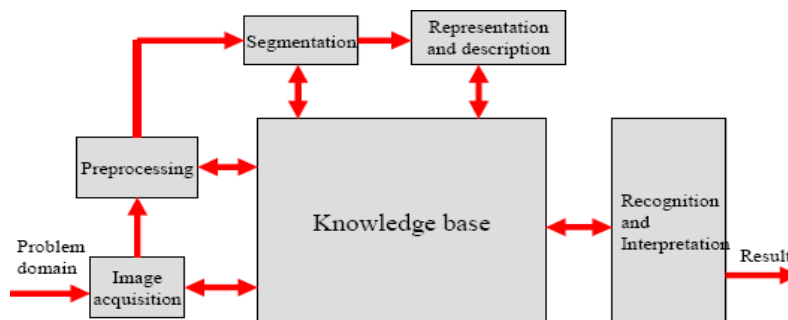


Fig 1.6 Fundamental steps in digital image processing

IMAGE PREPROCESSING

The key function of preprocessing is to improve the image in ways that increase the chances for success of the other processes. Preprocessing typically deals with

techniques for enhancing contrast, removing noise, and isolating regions whose texture indicate a likelihood of alphanumeric information.

- Image Enhancement.
- More suitable than original image for a specific application.
- Image Restoration.
- A process that attempts to reconstruct or recover an image that has been degraded using prior knowledge of the degradation concept.
-

IMAGE SEGMENTATION

The next stage deals with segmentation. Segmentation partitions an input image into its small constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing. On the one hand, a rugged segmentation procedure brings the process a long way towards the successful solution of an imaging problem. On the other hand, weak or erratic segmentation algorithms almost guarantee eventual failure.

In terms of character recognition, the key role of segmentation is to extract individual characters and words from the background.

IMAGE REPRESENTATION AND DESCRIPTION

The output of the segmentation stage usually is raw pixel data, constituting either the boundary of a region or all the points in the region itself. In either case converting the data to a form suitable for computer processing is necessary. The first decision that must be made, whether the data should be represented as a boundary or as a complete region. Boundary representation is appropriate when the focus is on external shape characteristics, such as corners and inflections. Regional representation is appropriate when the focus is on internal properties, such as texture or skeletal shape. In some applications, however, these representations coexist. This situation occurs in character recognition applications, which often require algorithms based on boundary shape as well as skeletons and other internal properties.

IMAGE RECOGNITION AND INTERPRETATION

Recognition is the process that assigns a label to an object based on the information provided by its descriptors. Interpretation involves assigning meaning to an ensemble of recognized objects. For example, identifying a character as, say, c requires associating the descriptors for that character with the label c. Interpretation attempts to assign meaning to a set of labeled entities. For example, a string of five numbers are followed by a hyphen and four more numbers can be interpreted to be a ZIP code.

CHAPTER 2

AIM AND

SCOPE

EXISTING SYSTEM:

Surveillance cameras are readily available in most car parking lots, so in many cases the solution is only to adequately process the information available from the already existing cameras, or complete the deployment by adding some cameras to have a full coverage that allows the system to operate. Image segmentation based systems try to differentiate, in each considered frame, between vehicles and parking spots. In this proposed system, our system will detect vehicles through video frames instead of using any other Mechanism. A camera will be installed alongside the traffic light. It will capture video sequences of traffic condition on road. This technique will analyze the videos from cameras and count the number of vehicles for each direction. It can reduce the traffic congestion and avoid the time being wasted by a green light on an empty road This system is more reliable to detect vehicle presence because it uses actual traffic condition images.

PROPOSED SYSTEM:

In this proposed system, our system will detect vehicles through video frames instead of using any other Mechanism. This technique will analyse the videos and count the number of vehicles for each direction. It can reduce the traffic congestion and avoid the time being wasted by a green light on an empty road. This system is more reliable to detect vehicle presence because it uses actual traffic condition images. It controls the traffic by counting the vehicles.

LITERATURE REVIEW

Project title: A Hierarchical Bayesian Generation Framework for Vacant Parking Space Detection

The main aim of this project is used to detect the car count value of background subtraction algorithm

Author name: Ching-Chun Huang and Sheng-Jyh Wang

Year of Publishing: 2010

Abstract: In this paper, from the viewpoint of scene understanding, a three-layer Bayesian hierarchical framework (BHF) is proposed for robust vacant parking space detection. In practice, the challenges of vacant parking space inference come from dramatic luminance variations, shadow effect, perspective distortion, and the inter-occlusion among vehicles. By using a hidden labeling layer between an observation layer and a scene layer, the BHF provides a systematic generative structure to model these variations. In the proposed BHF, the problem of luminance variations is treated as a color classification problem and is tackled via a classification process from the observation layer to the labeling layer, while the occlusion pattern, perspective distortion, and shadow effect are well modeled by the relationships between the scene layer and the labeling layer. With the BHF scheme, the detection of vacant parking spaces and the labeling of scene status are regarded as a unified Bayesian optimization problem subject to a shadow generation model, an occlusion generation model, and an object classification model. The system accuracy was evaluated by using outdoor parking lot videos captured from morning to evening. Experimental results showed that the proposed framework can systematically determine the vacant space number, efficiently label ground and car regions, precisely locate the shadowed regions, and effectively tackle the problem of luminance variations.

Project title: A Surface-based Vacant Space Detection for an Intelligent Parking Lot

Author name: Ching-Chun Huang, Yu-Shu Dai and Sheng-Jyh Wang

Year o Publishing: 2012

*Abstract:*We proposed a surface-based vacant parking space detection system. Unlike many car-oriented or space-oriented methods, the proposed system is parking-lot- oriented. In the system, we treat the whole parking lot as a structure consisting of plentiful surfaces. A surface-based hierarchical framework is then proposed to integrate the 3-D scene information with the patch-based image observation for the inference of vacant space. To be robust, the feature vector of each image patch is extracted based on the Histogram of Oriented Gradients (HOG) approach. By incorporating these texture features into the proposed probabilistic models, we could systematically infer the optimal hypothesis of parking statuses while dealing with occlusion effect, shadow effect, perspective distortion, and fluctuation of lighting condition in both day time and night time.

Project title: Car-Park Occupancy Information System

*Author name:*D.B.L.Bong, K.C.Ting, N.Rajae

Year o Publishing: 2015

*Abstract:*Drivers always face difficulty of finding available parking lot while entering into a huge parking area. A Car-Park Occupancy Information System (COINS) is perhaps a viable solution to reduce the amount of time needed to search for a vacant car-park lot especially in a huge parking area. With this system, images captured by a surveillance camera were processed in real-time to identify the occupancies of the parking lots. This occupancy information is further processed by a central control unit and distributed to display panels located at strategic locations at the parking area. The drivers can easily find a vacant parking lot based on the information displayed on the panels. Motivation for developing this system came from the fact that minimum cost is involved because image processing technique is used rather than sensor-based techniques. As surveillance cameras are readily available in most car parks, this technique is much cost effective than installing sensor on each parking lot. The concept of this system and its initial works with positive experiment results are presented in this paper.

Project title: Cost-Effective Single-Camera Multi-Car Parking Monitoring and Vacancy Detection towards Real-World Parking Statistics and Real-Time Reporting

Author name: Katy Blumer, Hala R. Halaseh, Mian Umair Ahsan, Haiwei Dong, and Nikolaos Mavridis

Year of Publishing: 2016

Abstract: Parking is a huge problem in densely populated areas and drivers spend a significant amount of time finding a suitable place to park their cars. A system that could show drivers the nearest available space would result in enormous savings of time, fuel, and street space. In order to achieve that, realworld periodic statistical analysis of car parking areas could help increase efficiency. Ideally, real-time information could also be used to create personalized suggestions to drivers, thus enabling satisficing of a wide range of possible criteria of optimality. We propose a system that uses a single camera for a widearea external parking, followed by a combination of two kinds of algorithms: static image analysis of parking lot spaces using a combination of histogram classification and edge detection, and dynamic image analysis using blob analysis. Our system thus achieves monitoring of parking spaces and reports statistics as well as empty slots in real- time. Our results indicate that almost 90% of empty spots are reported correctly, resulting in significant savings through a highly cost-effective single-camera system which can monitor more than 100 spaces.

Project title: A Multi-Classfier Image Based Vacant Parking Detection System

Author name: Junzhao Liu, Mohamed Mohandes, Mohamed Deriche King Fahd University of Petroleum and Minerals, Saudi Arabia

*Abstract:*With the rapid expansion of major cities, the tremendous increase in the number of vehicles, and the construction of huge buildings and parking areas, there is a need to develop smart parking systems to assist drivers finding the nearest available parking spots. Such systems have witnessed substantial research efforts in

developed countries. There are mainly four categories of car parking management systems: counter-based, wired-sensor-based, wireless-sensor-based, and image-based. In this paper, we

develop, implement, and test an image-based system for the detection of vacant spaces in a parking area. Following an initial edge detection stage, we combine edge density, closed contour density, and foreground/background pixel ratio, at every car parking spot, to identify whether a car is present or not. Combining the features above results in a robust vacant space detection system at low computational cost.

Project title: A BAYESIAN HIERARCHICAL DETECTION

FRAMEWORK FOR PARKING SPACE DETECTION

Author name: Ching-Chun Huang¹, Sheng-Jyh Wang¹, Yao-Jen Chang², and Tsuhan Chen²

Year of Publishing: 2014

Abstract: In this paper, a 3-layer Bayesian hierarchical detection framework (BHDF) is proposed for robust parking space detection. In practice, the challenges of the parking space detection problem come from luminance variations, interocclusions among cars, and occlusions caused by environmental obstacles. Instead of determining the status of parking spaces one by one, the proposed BHDF framework models the inter-occluded patterns as semantic knowledge and couple local classifiers with adjacency constraints to determine the status of parking spaces in a row-by-row manner. By applying the BHDF to the parking space detection problem, the available parking spaces and the labeling of parked cars can be achieved in a robust and efficient manner. Furthermore, this BHDF framework is generic enough to be used for various kinds of detection and segmentation applications.

Project title: Histograms of Oriented Gradients for Human Detection

Author name: Navneet Dalal and Bill Triggs

Abstract: We study the question of feature sets for robust visual object recognition, adopting linear SVM based human detection as a test case. After reviewing existing edge and gradient based descriptors, we show experimentally that grids of Histograms of

Oriented Gradient (HOG) descriptors significantly outperform existing feature sets for human detection. We study the influence of each stage of the computation on performance, concluding that gradients, orientation binning, relatively coarse spatial binning, and high-quality local contrast normalization in overlapping descriptor blocks are all important for good results. The new approach gives near-perfect separation on the original MIT pedestrian database, so we introduce a more challenging dataset containing over 1800 annotated human images with a large range of pose variations and backgrounds.

Project title: ROBUST PARKING SPACE DETECTION CONSIDERING INTER-SPACE CORRELATION

Author name: Qi Wu¹, Chingchun Huang², Shih-yu Wang³, Wei-chen Chiu³, Tsuhan Chen¹

Abstract: A major problem in metropolitan areas is searching for parking spaces. In this paper, we propose a novel method for parking space detection. Given input video captured by a camera, we can distinguish the empty spaces from the occupied spaces by using an 8-class Support Vector Machine (SVM) classifier with probabilistic outputs. Considering the inter-space correlation, the outputs of the SVM classifier are fused together using a Markov Random Field (MRF) framework. The result is much improved detection performance, even when there are significant occlusion and shadowing effects in the scene. Experimental results are given to show the robustness of the proposed approach.

Project title: SURF: Speeded Up Robust Features

Author name: Herbert Bay¹, Tinne Tuytelaars², and Luc Van Gool^{1,2}

Abstract: In this paper, we present a novel scale- and rotation-invariant interest point detector and descriptor, coined SURF (Speeded Up Robust Features). It approximates or even outperforms previously proposed schemes with respect to repeatability,

distinctiveness, and robustness, yet can be computed and compared much faster. This is achieved by relying on integral images for image convolutions; by building on the strengths of the leading existing detectors and descriptors (in casu, using a Hessian matrix-based measure for the detector, and a distribution-based descriptor); and by simplifying these methods to the essential. This leads to a combination of novel detection, description, and matching steps. The paper presents experimental results on a standard evaluation set, as well as on imagery obtained in the context of a real-life object recognition application. Both show SURF's strong performance.

Project title: Vacant Parking Space Detection Based on Plane-based Bayesian Hierarchical Framework

Author name: Ching-Chun Huang, *Member, IEEE*, Yu-Shu Tai, and Sheng-Jyh Wang, *Member, IEEE*

Abstract: In this paper, we propose a vacant parking space detection system that operates day and night. In the daytime, the major challenges of the system include dramatic lighting variations, shadow effect, inter-object occlusion, and perspective distortion. In the nighttime, the major challenges include insufficient illumination and complicated lighting conditions. To overcome these problems, we propose a plane-based method which adopts a structural 3-D parking lot model consisting of plentiful planar surfaces. The plane-based 3-D scene model plays a key part in handling inter-object occlusion and perspective distortion. On the other hand, to alleviate the interference of unpredictable lighting changes and shadows, we propose a planebased classification process. Moreover, by introducing a Bayesian hierarchical framework to integrate the 3- D model with the plane-based classification process, we systematically infer the parking status. Last, to overcome the insufficient illumination in the nighttime, we also introduce a preprocessing step to enhance image quality. The experimental results show that the proposed framework can achieve robust detection of vacant parking spaces in both daytime and nighttime.

Project title: Vacant Parking Space Detection in Static Images

Author name: Nicholas True

*Abstract:*Current parking space vacancy detection systems use simple trip sensors at the entry and exit points of parking lots. Unfortunately, this type of system fails when a vehicle takes up more than one spot or when a parking lot has different types of parking spaces. Therefore, I propose a camera-based system that would use computer vision algorithms for detecting vacant parking spaces. My algorithm uses a combination of car feature point detection and color histogram classification to detect vacant parking spaces in static overhead images.

Project title: Vision-Based Vehicle Surveillance and Parking Management Using Multiple Cameras

Author name: Li-Chih Chen¹, Jun-Wei Hsieh², Wei-Ru Lai¹, Chih-Xuan Wu¹, and Shin- Yu Chen¹

*Abstract:*This paper proposes a vision-based vehicle surveillance system for parking management in outdoor environments. Due to the limited field of view of camera, this system uses multiple cameras for monitoring a wide-area parking lot. Then, an affine transformation is used for linking the relations between cameras. Two major components are included in this system, i.e., vehicle counting and parking place management. For the first one, this system integrates three features, i.e., color, position, and motion together for well tracking vehicles across different cameras. Thus, even though two vehicles are occlude together, they still can be well identified and tracked under different lighting changes. For the second one, we propose a grid-based approach to model the color change of floor for determining whether a parking space is vacant. Due to the perspective effects, the visibility of a parking space is often affected by the vehicle parking on its neighbor space. To tackle this problem, we divide a parking space to different grids with different weights. Then, according to the orientation of parking space. An automatic weighting scheme is proposed for measuring the visibility of each grid. By judging the quantity of edge, we can determine whether the parking spaces are vacant. The experimental results reveal that our system works well and accurately in several different conditions.

CHAPTER 3

REQUIREMENT

SPECIFICATIONS

The requirements specification is a technical specification of requirements for the software products. It is the first step in the requirements analysis process it lists the requirements of a particular software system including functional, performance and security requirements. The requirements also provide usage scenarios from a user, an operational and an administrative perspective. The purpose of software requirements specification is to provide a detailed overview of the software project, its parameters and goals. This describes the project target audience and its user interface, hardware and software requirements. It defines how the client, team and audience see the project and its functionality.

HARDWARE REQUIREMENTS

Hard Disk : 40GB and Above

RAM : 512MB and Above

Processor : Pentium III and Above

SOFTWARE REQUIREMENTS

MATLAB R2013a

Windows Operating System XP and Above

MATLAB R2013a

The MATLAB ® high-performance language for technical computing integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include

- Math and computation

- Algorithm development
- Data acquisition
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. It allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar noninteractive language such as C or Fortran.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most users of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. You can add on toolboxes for signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many other areas.

INTRODUCTION TO MATLAB 8.3.0.532 (R2014a)

The MATLAB® System

The MATLAB system consists of these main parts:

Desktop Tools and Development Environment

This is the set of tools and facilities that help you use and become more productive with MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and Command Window, a command history, an editor and debugger, a code analyzer and other reports, and browsers for viewing help, the workspace, files, and the search path.

Mathematical Function Library

This is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix eigenvalues, Bessel functions, and fast Fourier transforms.

The Language

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create large and complex application programs.

Graphics

MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation,

and presentation graphics. It also includes low-level functions that allow you to fully customize

the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

External Interfaces

This is a library that allows you to write C and Fortran programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), for calling MATLAB as a computational engine, and for reading and writing MAT-files.

WORKING OF MATLAB®

Session Starting a MATLAB®

Session

On Microsoft® Windows® platforms, start the MATLAB® program by double-clicking the MATLAB R2014a shortcut on your Windows desktop.

On Apple® Macintosh® platforms, start MATLAB by double-clicking the MATLAB 7.6 icon on your Macintosh desktop.

On The Open Group UNIX® platforms, start MATLAB by typing matlab at the operating system prompt.

There are alternative ways to start MATLAB on Windows and Macintosh platforms, and you can customize MATLAB startup. For example, you can change the directory in which MATLAB starts or automatically execute MATLAB statements upon startup.

For More Information See Startup and Shutdown in the Desktop Tools and Development Environment documentation.

The Desktop

When you start MATLAB, the desktop appears, containing tools (graphical user interfaces) for managing files, variables, and applications associated with

MATLAB.

The following illustration shows the default desktop. You can customize the arrangement of tools and documents to suit your needs. For more information about the desktop tools, see Desktop Tools and Development Environment.

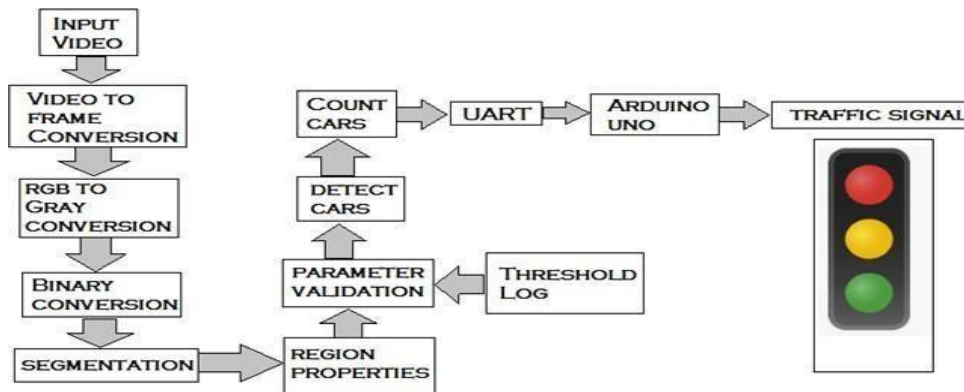
Quitting the MATLAB® Program

To end your MATLAB session, select File > Exit MATLAB in the desktop, or type quit in the Command Window. You can run a script file named finish.m each time MATLAB quits that, for example, executes functions to save the workspace.

CHAPTER 4

DESIGN METHDOLOGY

BLOCK DIAGRAM:



Module Description:

Module I: Video to frame conversion

Take an Input Video and Extract the images from the video and feed the images to YCbCr color space for blob detection.

Module II: Segmentation

Background subtraction is a typical technique used in this category, where an empty image is used to subtract each frame in order to get the foreground mask (vehicles). Then the segmentation technique is applied so that it can detect only cars in the video.

Module III: Bounding box

As we are working with a four way intersection, for the time being, we will break this down for just one road only. For each road we will check twice for traffic. We have shown before that we can cover almost 50 cars (48.2 cars to be exact) with our camera for a specific road. So we will put a threshold value of 48 cars, means if our video

processor could detect the cars. Finally the car count value will be display in the LCD display.

Background Subtraction:

Background subtraction is a technique for separating out foreground elements from the background and is done by generating a foreground mask. This technique is used for detecting dynamically moving objects from static cameras. Background subtraction technique is important for object tracking. There are several techniques for background subtraction

The running average of a function is used to separate foreground from background. In this concept, the video sequence is analyzed over a particular set of frames. During this sequence of frames, the running average over the current frame and the previous frames is computed. This gives us the background model and any new object introduced in the during the sequencing of the video becomes the part of the foreground. Then, the current frame holds the newly introduced object with the background. Then the computation of the absolute difference between the background model (which is a function of time) and the current frame (which is newly introduced object) is done.

Background subtraction is process of extracting foreground objects from maintained background model. A foreground object is any entity that detected by producing difference of the every frame of sequence to background model. This result can be further used for tracking targets, motion detection. Background subtraction further divides into parametric and non-parametric background subtraction. There are different background subtractions techniques have been proposed in literature. The background model can be static or dynamic. The flowchart for Background subtraction is shown in figure 4. Dynamic background model is one in which the background of scene may contain moving objects in outdoor environment, Pixel-based and block based are two major kind of approached are for background Subtraction. To construct a statistical representation of background scene non-parametric statistical Modelling of pixel process is used. The different challenges that have to face to construct a good background subtraction algorithm are robustness against the changes in illumination and shadow

detection.

Background subtraction is a widely used approach for detecting moving objects in videos from static cameras. The rationale in the approach is that of detecting the moving objects from the difference between the current frame and a reference frame, often called the “background image”, or “background model”. As a basic, the background image must be a representation of the scene with no moving objects and must be kept regularly updated so as to adapt to the varying luminaries conditions and geometry settings. More complex models have extended the concept of “background subtraction” beyond its literal meaning. Several methods for performing background subtraction have been proposed in the recent literary. All of these methods try to effectively estimate the background model from the temporal sequence of the frames. However, there is a wide variety of techniques and both the expert and the newcomer to this area can be confused about the benefits and limitations of each method. This paper provides a thorough review of the main methods (with inevitable exclusions due to space restrictions) and an original categorization based on speed, memory requirements and accuracy.

Background subtraction is one of the key techniques for automatic video analysis, especially in the domain of video surveillance. Although its importance, evaluations of recent background subtraction methods with respect to the challenges of video surveillance suffer from various shortcomings. To address this issue, we first identify the main challenges of background subtraction in the field of video surveillance. We then compare the performance of nine background subtraction methods with post-processing according to their ability to meet those challenges. Therefore, we introduce a new evaluation data set with accurate ground truth annotations and shadow masks. This enables us to provide precise in depth evaluation of the strengths and drawbacks of background subtraction methods. A common method for search-space reduction and focus of attention modeling in video analysis is background subtraction (BS). The term BS covers a set of methods that aim to distinguish between foreground and background areas in video sequences utilizing a background model.

CHAPTER 5

RESULTS AND

DISCUSSIONS

Once the design aspect of the system is finalized, the system enters into the coding and testing phase. The coding phase brings the actual system into action by converting the design of the system into the code in a given programming language. Therefore, a good coding style has to be taken whenever changes are required, it easily screws into the system. Coding standards are guidelines to programming that focus on the physical structure and appearance of the program. They make the code easier to read, understand and maintain. This phase of the system actually implements the blueprint developed during the design phase. The coding specification should be in such a way that any programmer must be able to understand the code and can bring about changes whenever felt necessary. Some of the standards needed to achieve the above-mentioned objectives are as follows:

Program should be simple, clear and easy to

understand. Naming conventions

Value conventions

Script and comment

procedure Message box

format Exception and error

handling

NAMING CONVENTIONS

Naming conventions of classes, data member, member functions, procedures etc., should be **self-descriptive**. One should even get the meaning and scope of the

variable by its name. The conventions are adopted for **easy understanding** of the intended

message by the user. So it is customary to follow the conventions. These conventions are as follows:

Class names

Class names are problem domain equivalence and begin with capital letter and have mixed cases.

Member Function and Data Member name

Member function and data member name begins with a lowercase letter with each subsequent letters of the new words in uppercase and the rest of letters in lowercase.

VALUE CONVENTIONS

Value conventions ensure values for variable at any point of time. This involves the following:

Proper default values for the variables.

Proper validation of values in the field.

Proper documentation of flag values.

SCRIPT WRITING AND COMMENTING STANDARD

Script writing is an art in which indentation is utmost important. Conditional and looping statements are to be properly aligned to facilitate easy understanding. Comments are included to minimize the number of surprises that could occur when going through the code.

MESSAGE BOX FORMAT

When something has to be prompted to the user, he must be able to understand it properly. To achieve this, a specific format has been adopted in displaying messages to the user. They are as follows:

X – User has performed illegal operation.

! – Information to the user

TEST PROCEDURE

SYSTEM TESTING

Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. The goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself. For example the design must not have any logic faults in the design is detected before coding commences, otherwise the cost of fixing the faults will be considerably higher as reflected. Detection of design faults can be achieved by means of inspection as well as walkthrough.

Testing is one of the important steps in the software development phase. Testing checks for the errors, as a whole of the project testing involves the following test cases:

Static analysis is used to investigate the structural properties of the Source code.

Dynamic testing is used to investigate the behavior of the source code by executing the program on the test data.

TEST DATA AND OUTPUT

UNIT TESTING

Unit testing is conducted to verify the functional performance of each modular component of the software. Unit testing focuses on the smallest unit of the software design (i.e.), the module. The white-box testing techniques were heavily employed for unit testing.

FUNCTIONAL TESTS

Functional test cases involved exercising the code with nominal input values for which the expected results are known, as well as boundary values and special values, such as logically related inputs, files of identical elements, and empty files.

Three types of tests in Functional test:

Performance Test

Stress Test

Structure Test

PERFORMANCE TEST

It determines the amount of execution time spent in various parts of the unit, program throughput, and response time and device utilization by the program unit.

STRESS TEST

Stress Test is those test designed to intentionally break the unit. A Great deal can be learned about the strength and limitations of a program by examining the manner in which a programmer in which a program unit breaks.

STRUCTURED TEST

Structure Tests are concerned with exercising the internal logic of a program and traversing particular execution paths. The way in which White-Box test strategy was employed to ensure that the test cases could Guarantee that all independent paths within a module have been have been exercised at least once.

Exercise all logical decisions on their true or false sides.

Execute all loops at their boundaries and within their operational bounds.

Exercise internal data structures to assure their validity.

Checking attributes for their correctness.

Handling end of file condition, I/O errors, buffer problems and textual errors in output information

INTEGRATION TESTING

Integration testing is a systematic technique for construction the program structure while at the same time conducting tests to uncover errors associated with interfacing. i.e., integration testing is the complete testing of the set of modules which makes up the product. The objective is to take untested modules and build a program structure tester should identify critical modules. Critical modules should be tested as early as possible. One approach is to wait until all the units have passed testing, and then combine them and then tested. This approach is evolved from unstructured testing of small programs. Another strategy is to construct the product in increments of tested units. A small set of modules are integrated together and tested, to which another module is added and tested in combination. And so on. The advantages of this approach are that, interface dispenses can be easily found and corrected.

The major error that was faced during the project is linking error. When all the modules are combined the link is not set properly with all support files. Then we checked out for interconnection and the links. Errors are localized to the new module and its intercommunications. The product development can be staged, and modules integrated in as they complete unit testing. Testing is completed when the last module is integrated and tested.

TESTING TECHNIQUES / TESTING

STRATERGIES TESTING

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet –undiscovered error. A successful test is one that uncovers an as-yet- undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It verifies that the whole set of programs hang together. System testing requires a test consists of several key activities and steps for run program, string, system and is important in adopting a

successful new system. This is the last chance to detect and correct errors before the system is installed for user acceptance testing.

The software testing process commences once the program is created and the documentation and related data structures are designed. Software testing is essential for correcting errors. Otherwise the program or the project is not said to be complete. Software testing is the critical element of software quality assurance and represents the ultimate review of specification design and coding. Testing is the process of executing the program with the intent of finding the error. A good test case design is one that has a probability of finding a yet undiscovered error. A successful test is one that uncovers a yet undiscovered error. Any engineering product can be tested in one of the two ways:

WHITE BOX TESTING

This testing is also called as Glass box testing. In this testing, by knowing the specific functions that a product has been designed to perform test can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases. Basis path testing is a white box testing.

Basis path testing:

- Flow graph notation

- Cyclomatic complexity

- Deriving test cases

- Graph matrices Control

BLACK BOX TESTING

In this testing by knowing the internal operation of a product, test can be conducted to ensure that “all gears mesh”, that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

The steps involved in black box test case design are:

- Graph based testing methods
- Equivalence partitioning
- Boundary value analysis
- Comparison testing

SOFTWARE TESTING STRATEGIES:

A software testing strategy provides a road map for the software developer. Testing is a set activity that can be planned in advance and conducted systematically. For this reason a template for software testing a set of steps into which we can place specific test case design methods should be strategy should have the following characteristics:

- Testing begins at the module level and works “outward” toward the integration of the entire computer based system.

- Different testing techniques are appropriate at different points in time.

- The developer of the software and an independent test group conducts testing.

- Testing and Debugging are different activities but debugging must be accommodated in any testing strategy.

INTEGRATION TESTING:

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with. Individual modules, which are highly prone to interface errors, should not be assumed to work instantly when we put them together. The problem of course, is “putting them together”- interfacing. There may be the chances of data lost across on another’s sub functions, when combined may not produce the desired major function; individually acceptable impression may be magnified to unacceptable levels; global data structures can present problems.

PROGRAM TESTING:

The logical and syntax errors have been pointed out by program testing. A syntax error is an error in a program statement that in violates one or more rules of the language in which it is written. An improperly defined field dimension or omitted keywords are common syntax error. These errors are shown through error messages generated by the computer. A logic error on the other hand deals with the incorrect data fields, out-off-range items and invalid combinations. Since the compiler s will not deduct logical error, the programmer must examine the output. Condition testing exercises the logical conditions contained in a module. The possible types of elements in a condition include a Boolean operator, Boolean variable, a pair of Boolean parentheses A relational operator or on arithmetic expression. Condition testing method focuses on testing each condition in the program the purpose of condition test is to deduct not only errors in the condition of a program but also other a errors in the program.

SECURITY TESTING:

Security testing attempts to verify the protection mechanisms built in to a system well, in fact, protect it from improper penetration. The system security must be tested for invulnerability from frontal attack must also be tested for invulnerability from rear attack. During security, the tester places the role of individual who desires to penetrate system.

VALIDATION TESTING

At the culmination of integration testing, software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test-validation testing begins. Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in manner that is reasonably expected by the customer. Software validation is achieved through a series of black box tests that demonstrate conformity with requirement. After validation test has been conducted, one of two conditions exists.

* The function or performance characteristics confirm to specifications and are accepted.

- * A validation from specification is uncovered and a deficiency created.

Deviation or errors discovered at this step in this project is corrected prior to completion of the project with the help of the user by negotiating to establish a method for resolving deficiencies. Thus the proposed system under consideration has been tested by using validation testing and found to be working satisfactorily. Though there were deficiencies in the system they were not catastrophic.

USER ACCEPTANCE TESTING

User acceptance of the system is key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system and user at the time of developing and making changes whenever required. This is done in regarding to the following points.

- Input screen design.
- Output screen design.

CHAPTER 6

CONCLUSION:

This have a look at suggested a parking space detection technique based totally on an AVM device and related emptiness analysis algorithms too facilitate indoor parking wherein numerous different lights assets are gift, and out of doors parking, where environmental versions can be vast. we suggest a aircraft-primarily based method which adopts an operational three-D parking lot characteristic such as abundant planar aspects. The aircraft-based totally three-D segment model performs a key element in conduct inter-item occlusion and point of view alteration. however, to relieve the interference of unpredictable illumination deviations and sun shades, we endorse aircraft-based class system. furthermore, by using supplying a Bayesian categorized framework to participate the three-D prototypical with the aircraft-based organisation technique, we methodically infer the space popularity. final, to triumph over the insufficient lighting fixtures within the night-time, we additionally gift a pre-processing step to improve picture excellence. further, a incredibly sturdy evaluation approach was suggested to classify parking area occupancy. Perspectives have been used to detect capabilities, categorized as avenue or impediment capabilities, inside the parking area. Road capabilities were used to offer information concerning thee viable vacancies of a parking spaces' and obstacles capabilities had been used to offer records regarding the feasible occupancy of a parking area. sooner or later, those two varieties of information had been incorporated to determine whether or not a selected parking space is occupied.

REFERENCES:

1. Groh, B.H.; Friedl, M.; Linarth, A.G.; Angelopoulou, E. Advanced real-time indoor parking localization based on semi-static objects. In Proceedings of the 2014 17th International Conference on Information Fusion (FUSION), Salamanca, Spain, 7–10 July 2014; pp. 1–7. [Google Scholar]z'
2. Ata, K.M.; Soh, A.C.; Ishak, A.J.; Jaafar, H.; Khairuddin, N.A. Smart Indoor Parking System Based on Dijkstra's Algorithm. IJEEAS 2019, 2, 13–20. [Google Scholar]
3. Jeong, S.H.; Choi, C.G.; Oh, J.N.; Yoon, P.J.; Kim, B.S.; Kim, M.; Lee, K.H. Low cost design of parallel parking assist system based on an ultrasonic sensor. Int. J. Autom. Technol. 2010, 11, 409–416. [Google Scholar] [CrossRef]
4. Pohl, J.; Sethsson, M.; Degerman, P.; Larsson, J. A semi-automated parallel parking system for passenger cars. Proc. Inst. Mech. Eng. D J. Automob. Eng. 2006, 220, 53–65. [Google Scholar] [CrossRef]
5. Jung, H.G.; Cho, Y.H.; Yoon, P.J.; Kim, J. Scanning laser radar-based target position designation for parking aid system. IEEE Trans. Intell. Transp. Syst. 2008, 9, 406–424. [Google Scholar] [CrossRef]z'
6. Suhr, J.K.; Jung, H.G.; Bae, K.; Kim, J. Automatic free parking space detection by using motion stereo-based 3D reconstruction. Mach. Vis. Appl. 2010, 21, 163–176. [Google Scholar] [CrossRef]
7. Jung, H.G.; Kim, D.S.; Yoon, P.J.; Kim, J. Parking slot markings recognition for automatic parking assist system. In Proceedings of the IEEE Intelligent Vehicles Symposium, Tokyo, Japan, 13–15 June 2006; pp. 106–113. [Google Scholar]
8. Lee, S.; Seo, S.W. Available parking slot recognition based on slot context analysis. IET Intell. Transp. Syst. 2016, 10, 594–604. [Google Scholar] [CrossRef]
9. Suhr, J.K.; Jung, H.G. Full-automatic recognition of various parking slot

markings using a hierarchical tree structure. Opt. Eng. 2013, 52, 037203.
[Google Scholar] [CrossRef]

10. Sevellano, X.; Màrmol, E.; Fernandez-Arguedas, V. Towards smart traffic management systems: Vehicle on-street parking spot detection based on video analytics. In Proceedings of the 17th International Conference on Information Fusion, Salamanca, Spain, 7–10 July 2014. [Google Scholar]z'

SOURCE CODE

```
clc;
clear
all;
close
all;

delete('./db/*.png')

%%

[f p]=uigetfile('*.');
vob = VideoReader([p
f]); frame =
vob.read(inf);
video=ExportMovement(vob,frame); frame=read(vob);

video=VideoReader(video);
frames=read(video);
num=get(video,'numberOfFrames'); cr=frames(:,:,1);

if
    f=='video1
    .avi' l=12;
    l1=53;
    ss=[9 13 346 197];
else
    l=9
    ;
    l1=
```

52;

ss=[9.5 1.5 1149.98 77.98];

en

d

l2=

57;

```

iii=1;
for i=1:num
    image=frames(:,:,i
    );
    image=imcrop(image,[ss(1) ss(2) ss(3) ss(4)]);
% %
figure,imshow(image,[]);
bw=im2bw(image,0.1);
% %    figure,imshow(bw,[]);

```

```

    se =
    strel('line',l,90); bw
    = imdilate(bw,se);

```

```

    [imlabel]=bwlabel(
    bw);
    stats=regionprops(i
    m);
    area=[stats.Area];

```

```

    loc=find(area>280);
    ima=zeros(size(im));

```

```

    for ii=1:length(loc)
% %
    ima=zeros(size(im));
    ima(im==loc(ii))=1;
%
    figure,imshow(ima
    ) if ii==length(loc)
    path=['./db/' num2str(iii)

```

```
'.png']; imwrite(ima,path)
```

```
    num(iii
```

```
)=i;
```

```
    iii=iii+1;
```

```
end
```

```

end
close
all;
end

```

```

addpath('.\db\');
Dire=dir('.\db\');
names =
{Dire.name};
Fol_len =
length(Dire);
Len_of_folder = 0;
for i = 1: Fol_len
    if not(strcmp(names(i),'.')| strcmp(names(i) , '..') | strcmp(names(i) , 'Thumbs.db'))
Len_of_folder = Len_of_folder + 1;
names{Len_of_folder}=name
    s{i}; end
end
% data_send = serial('COM3');
%
fopen(data_send)
count=0;
cout=
0;
area1
=0;
for i=1:Len_of_folder
    path=['./db/' num2str(i)
'.png']; new=num(i);
    input=imcrop(frame(:,:,new),[ss(1) ss(2) ss(3) ss(4)]);
    line= [ 1 l1 size(input,2) l2-l1];

```

```
image=imread(path);  
figure,imshow(input)  
[imlabel]=bwlabel(image);  
  
for i=1:label
```

```

        im1=zeros(size(i
        m));
        centroids=zeros(1,
        2 );
bbb=zeros(4,1);
        im1(find(im==i))=1;
        s=regionprops(im1,'centroid','BoundingBox','Area');
        centroids = ceil(cat(1, s.Centroid));
bbb=cat(1,
        s.BoundingBox);
        area=cat(1, s.Area);
        res=sprintf('No of vehicle
        %d',count); hold on
        rectangle('Position',line,'EdgeColor','g','LineWidth',2)
        plot(centroids(:,1), centroids(:,2), 'b*')
xlabel(res)
        if l1<=centroids(1,2) &
            centroids(1,2)<=57 if 700<area1 |
            area1<400 | area>800
                count=count+1
cout=cout+1;
        hold on
        rectangle('Position',bbb,'EdgeColor','r','LineWidth',
        2) area1=area;
        end
e
n
d
e
n
d
        if cout==3
%         fprintf(data_send,'%s\r\n','24');

```

```
%      pause(2)
%
    cout=0;
    else if
    cout>=6
%      fprintf(data_send,'%s\r\n','55');
%
    pause(3)
end
```

```
    end
    pause(0
.4);
    close
all; end
```

```
% fclose(data_send);
```

```
function video=ExportMovement(vob,
frame) render_output = 1;
draw_diff = 1;
%Control
parameters
pause_on = 0;
delay_on = 0;
delay_length = 0.1;
show_background = 0;
video_chunk_size = 10;
```

```
% input_video =
[p f]; downsample
= 1;
movement_thresh = 0.2;
use_grayscale = 0;
frame_rate = 30;
dt =
downsample/frame_rate;
bk_downsample = 20;
```

```
%% Open video
disp('Opening
```

video...')

```

% % vob = VideoReader(input_video);
% % frame =
vob.read(inf); vidHeight
= vob.Height; vidWidth
= vob.Width;
nFrames = vob.NumberOfFrames;

% Preallocate movie structure.
disp('Preallocating video...')
sub_mov(1:(nFrames/downsample))
= ...
struct('cdata', zeros(vidHeight, vidWidth, 3,
    'uint8'),... 'colormap', []);

%% Determine background frame: Update this to use the recursive background estimator
background_frame = frame*0;
disp('Calculating
background...') for k =
1:bk_downsample:nFrames
background_frame = background_frame + read(vob, k)/(nFrames/bk_downsample);
disp(k/(nFrames)*100)
end
main_image =
imshow(background_frame); if
show_background
    pa
use
end

if render_output
vw =
VideoWriter('cars.avi');

```

```
open(vw);  
sub_mov(1).cdata = imabsdiff(read(vob, 1),  
background_frame); [vidHeightvidWidthvidDepth] =  
size(sub_mov(1).cdata);  
end
```

```

%% Read one frame at a
time. downsample_index =
0;
buffer_index = 0;
frame_buffer = read(vob,[1 video_chunk_size]);

for k = 1 : nFrames-1

downsample_index = downsample_index + 1;
    movement = imabsdiff(read(vob,k),background_frame);
    pause(delay_length)

    if use_grayscale
im =
        rgb2gray(movement)
    ; else
im = im2bw(movement,
    movement_thresh); end

    if render_output
clf
drawing_image = read(vob, k +
    1); if draw_diff == 1
movement_old =
        imabsdiff(drawing_image,background_frame); if
        use_grayscale
drawing_image =
            rgb2gray(movement_old); else
drawing_image = im2bw(movement_old,
            movement_thresh); end
end

```

```
imshow(drawing_image) drawnow
    hold on
```

```
drawnow
currFrame = getframe;
currFrame.cdata = imresize(currFrame.cdata,
[vidHeightvidWidth]); writeVideo(vw,currFrame);
    end
```

```
disp(k/(nFrames)*1
```

```
00);
```

```
    if pause_on
waitforbuttonpr
ess
    e
nd
en
d
```

```
if
render_outp
ut close(vw);
end
video='cars.
avi';
```

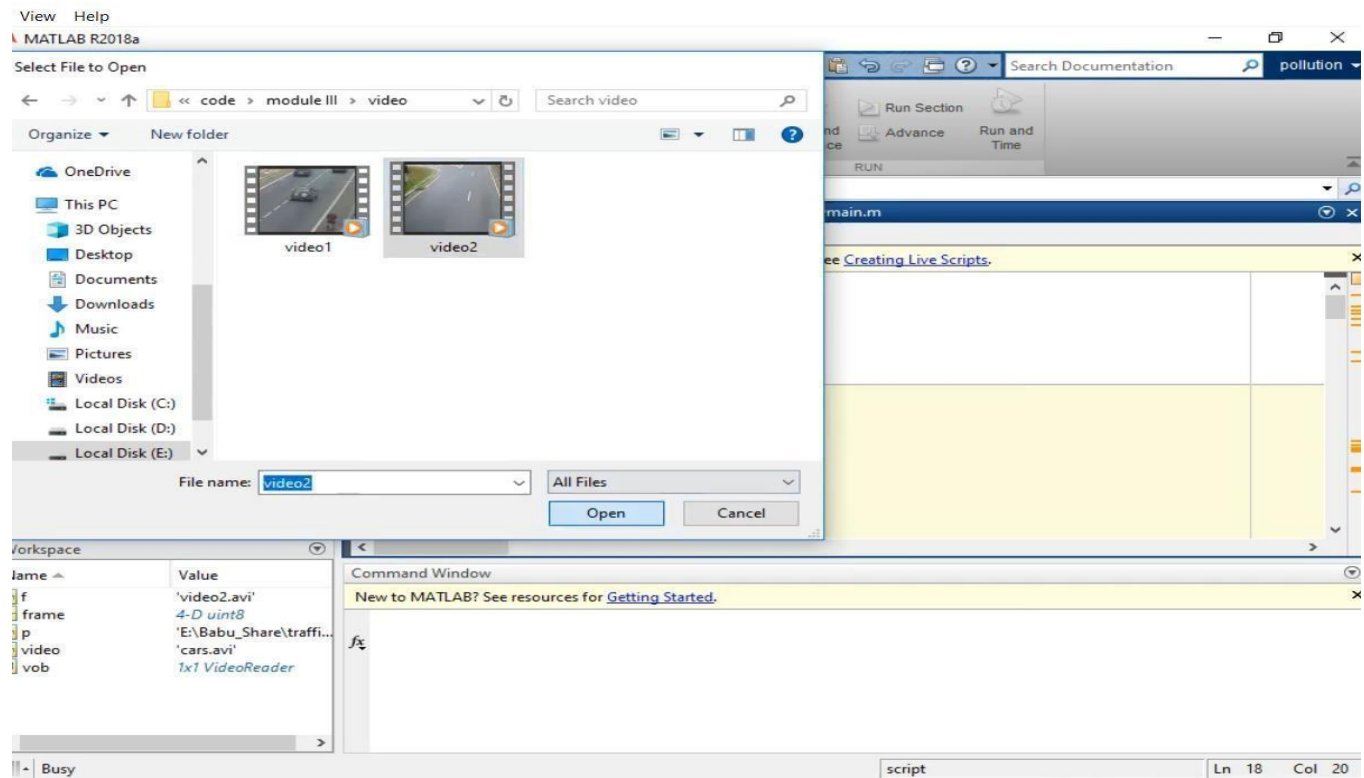
```
function [x y ] = region_interst(i_image,i)
```

```
% n_of_frames = get(vid,'NumberOfFrames');  
% input_video = read(vid);
```

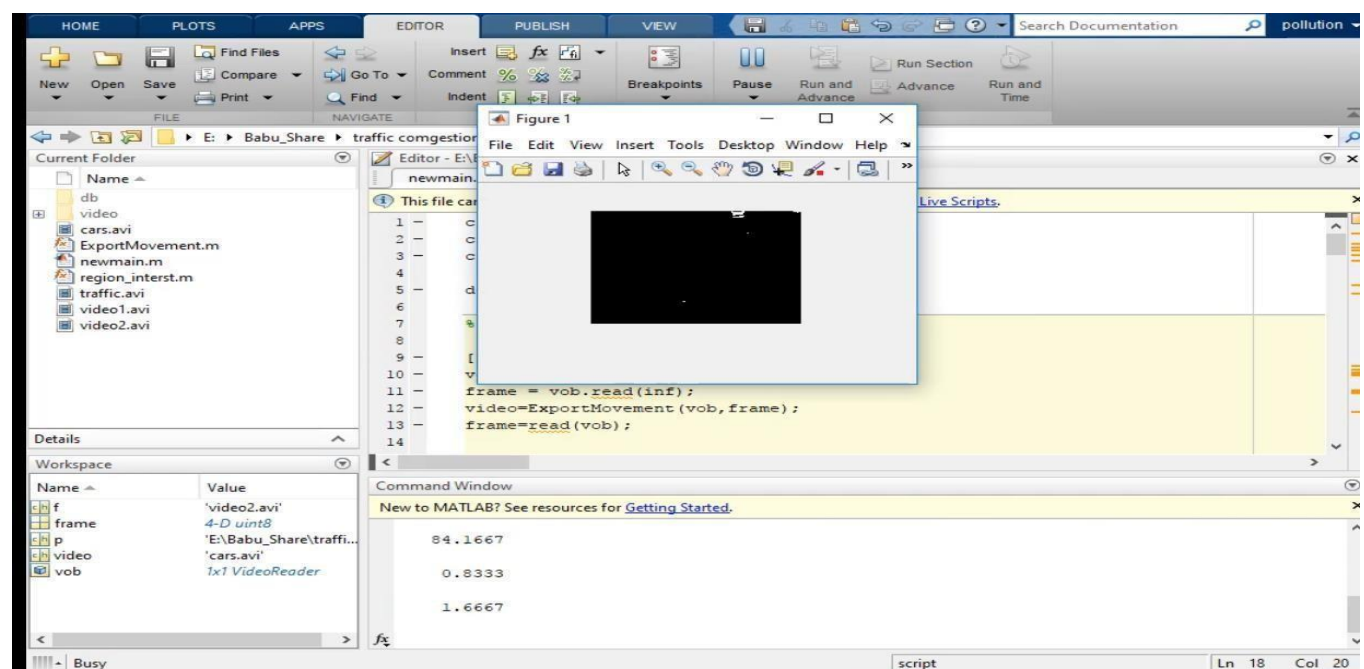
```
frames = i_image(:,:,i);
```

```
% %  
imshow(frames);  
[x  
y]=imcrop(frames);
```

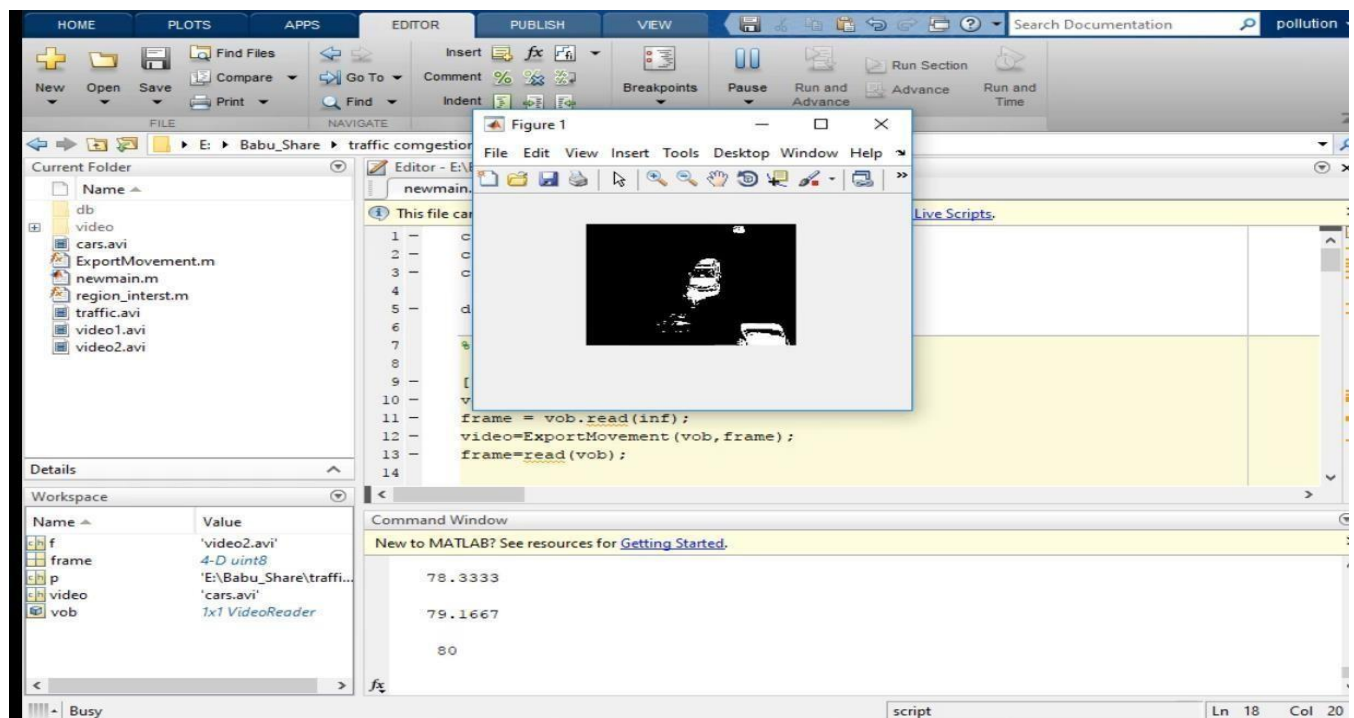
SCREENSHOTS



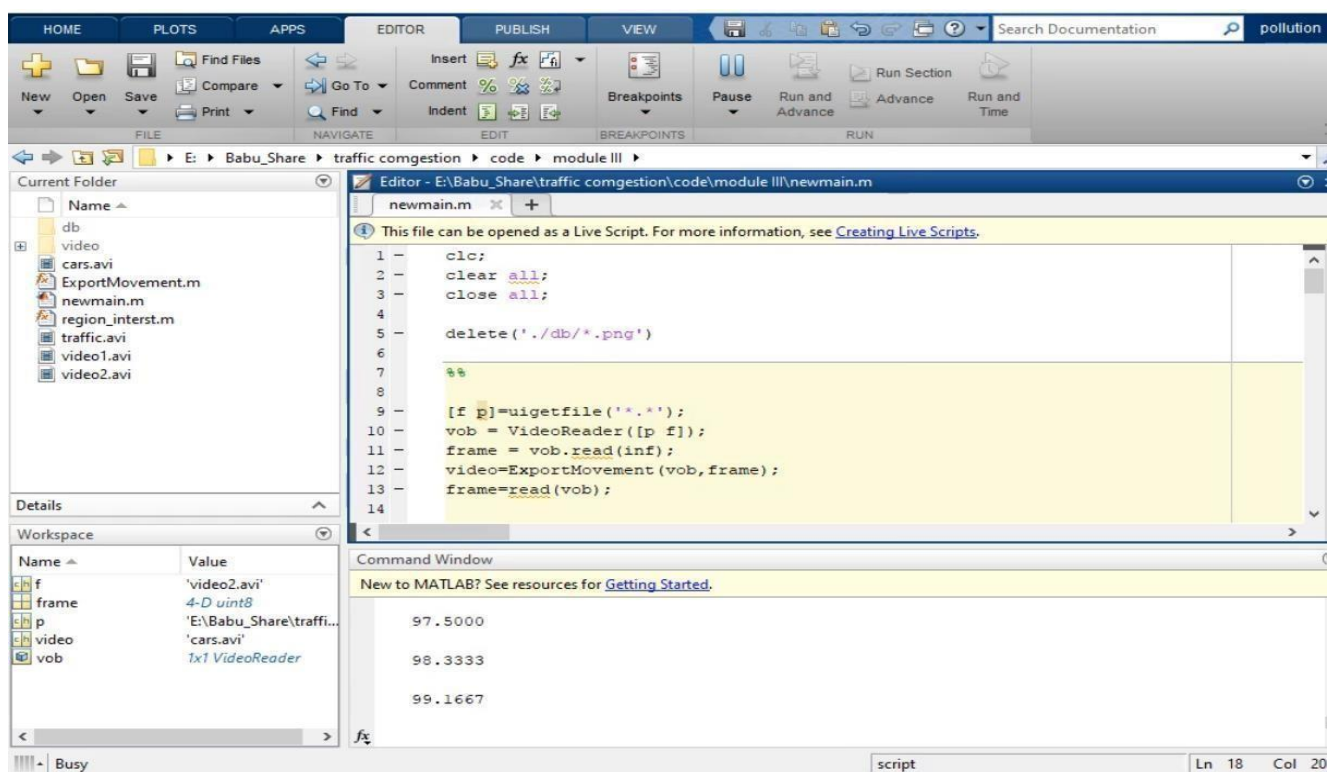
Input the video



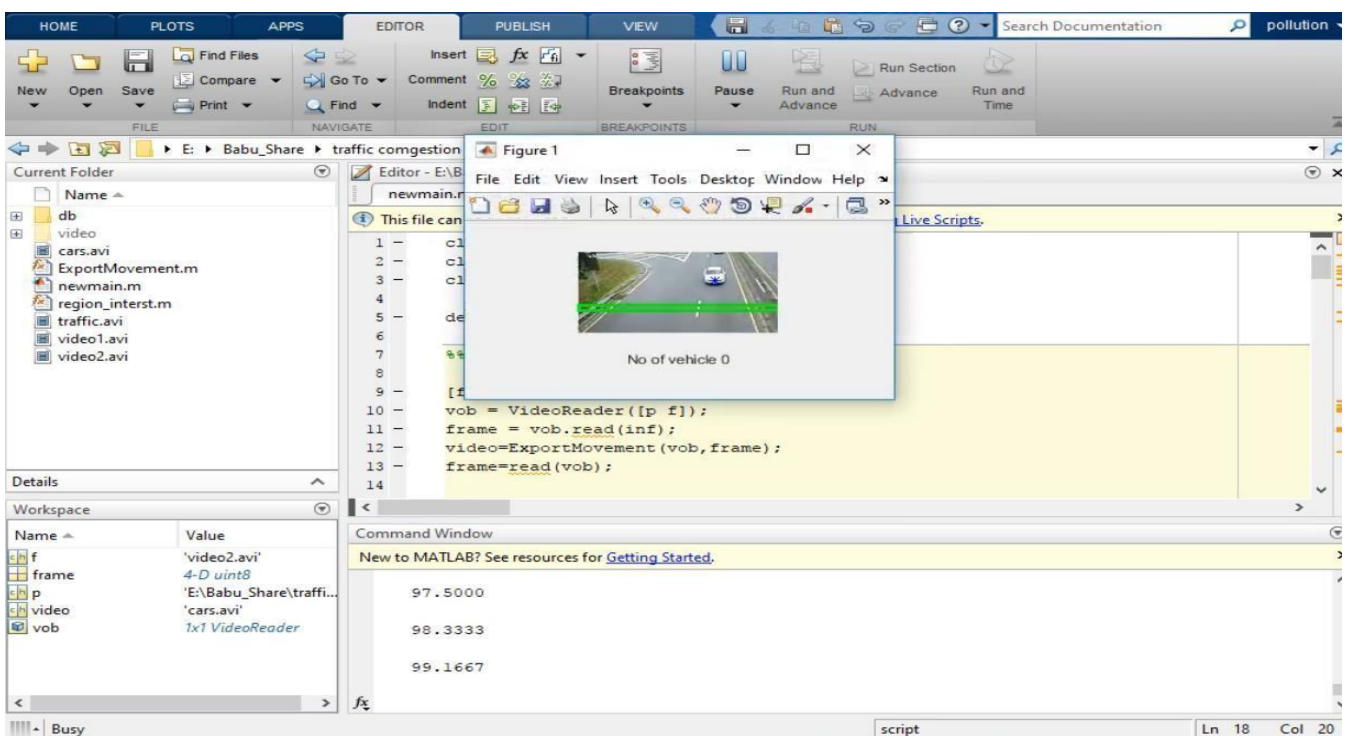
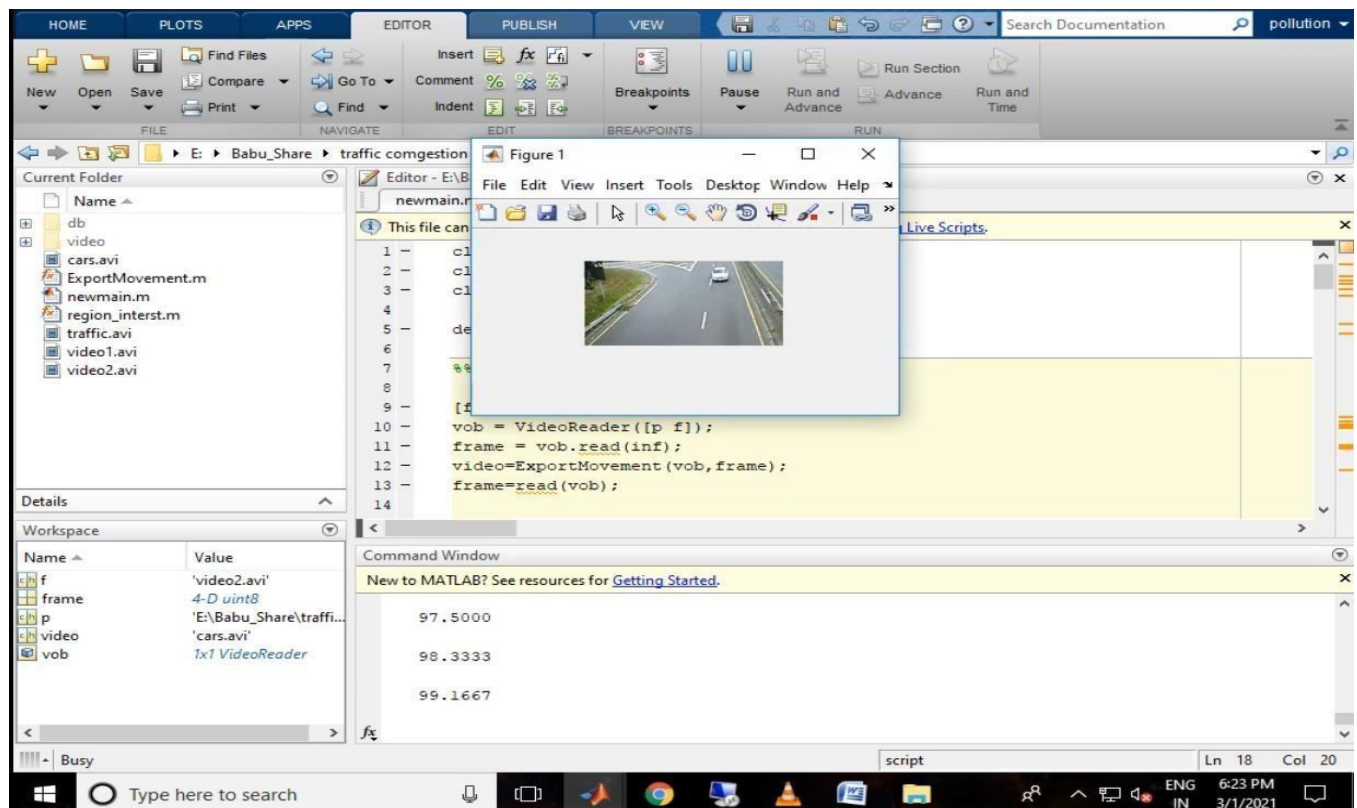
Background subtraction takes place .



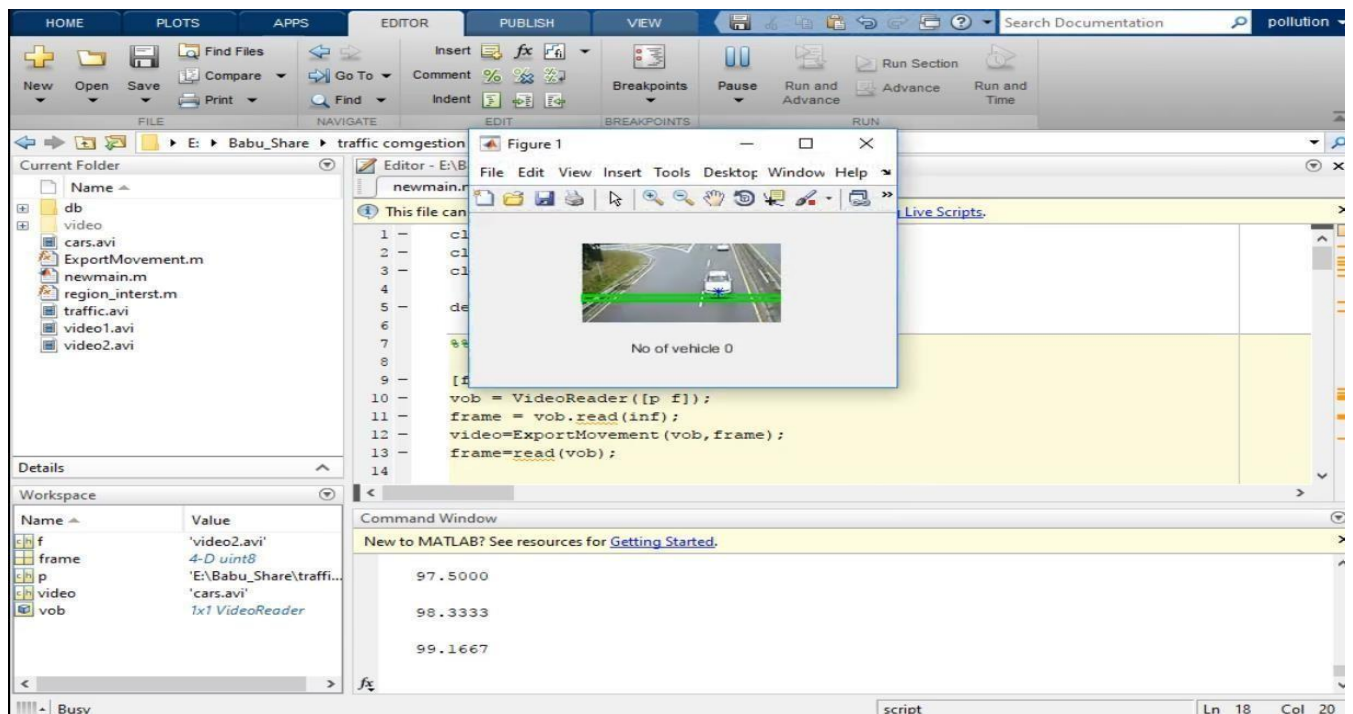
It removes all the unwanted background and turns background into black colour and highlights the important data into white colour.



Video is converted into frames conversion into binary image.

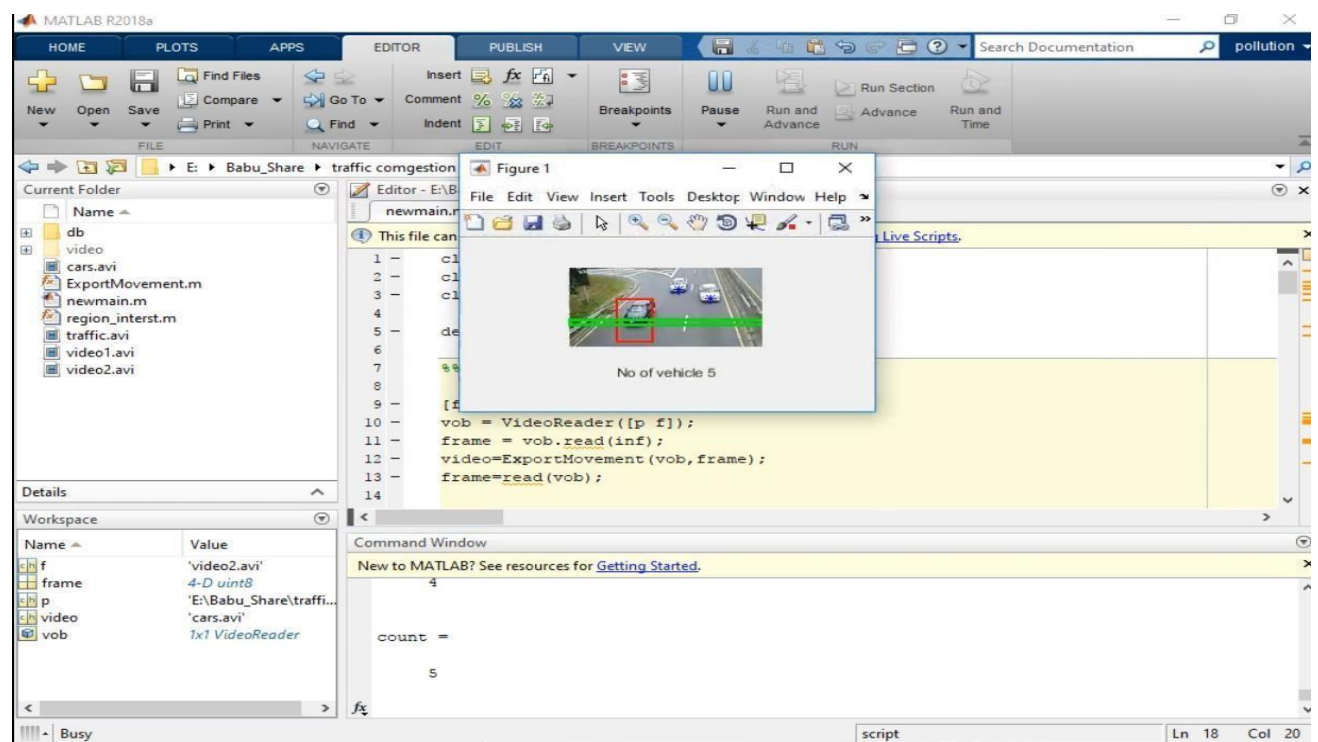


Lane detection takes place.



Like it detects when ever a car entering into a lane.

B



Bounding Box takes place like it creates a box over the vehicle and simultaneously counting car takes place.