

PROFESSIONAL TRAINING REPORT
at
Sathyabama Institute of Science and Technology
(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award
of Bachelor of Technology Degree in

Information Technology

By

AARTHY G
(Reg. No.38120001)



DEPARTMENT OF INFORMATION TECHNOLOGY
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

JUNE 2020



SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY

(Established under Section 3 of UGC Act, 1956)
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai -
600119 www.sathyabama.ac.in



SCHOOL OF COMPUTING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of
AARTHY G (Reg.no.38120001) who carried out the project entitled
“**VEHICLE COUNTING USING MACHINE LEARNING**”
under our supervision from June 2020 to December 2020.

Internal Guide

Dr. SENDURU SRINIVASULU M.E., Ph.D.,

Head of the Department

Dr. R. SUBHASHINI M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I AARTHY G(Reg No: 38120001) hereby declare that the Project Report entitled “VEHICLE COUNTING USING MACHINE LEARNING” done by me under the guidance of Dr. /Prof./ Mr. Dr. SENDURU SRINIVASULU, M.E., Ph.D., is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Information Technology.

DATE:21/01/2021

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge our 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 our thanks to **Dr. T. SASIKALA M.E., Ph.D.**, Dean, School of Computing and **Dr. R. SUBHASHINI M.E., Ph.D.**, Head of the Department, Department of Information Technology for providing us the 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. SENDURU SRINIVASULU M.E., Ph.D.**, for his 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 **INFORMATION TECHNOLOGY** who were helpful in many ways for the completion of the project.

TRAINING CERTIFICATE



RETECH
SOLUTIONS PVT. LTD.
An ISO 9001:2015 certified company

INT: RE20S066

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **AARTHY G** student of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY, BTECH/IT** has undergone the Internship Training through online in our concern entitled **PYTHON** from **19th Aug 2020** to **28th Aug 2020** in relevant departments related to their academic studies.

During the above period, the performance was good & we wish great success in all your future endeavours.

J. mere

Technology drives the world. We are the people who are behind the technology.
Authorised Signature



No-31, 1st Floor, Alagesan Street, Tambaram West, Chennai-45.

Office : 044-42077204 | +91-8122266653

www.retechsolutions.in

ABSTRACT

This project is about counting the vehicle using Machine Learning. Vehicle counting is considered one of the most important applications in traffic control and management. Vehicle counting process provides appropriate information about traffic flow, vehicle crash occurrences and traffic peak times in roadways. If you have a good sense of the volume of traffic moving along a given road or network of roads, you can better understand congestion and then manage and/or make plans to reduce/eliminate it. Then, a small decrease in classification performance can have serious economic losses. Therefore, the accuracy and the time complexity become critical for the traffic system. The algorithm used here will count number of vehicles. However, it requires a large labeled datasets, and has a limitations when multiple vehicles are in the scene. Herein, we propose machine learning techniques. The experiments show that our method is performing as accurately as the existing model with significantly lower labeled datasets We apply Haar Cascade algorithm to count the number of vehicles in traffic signals. when it is found that density of vehicles is more in traffic signal it will detect and create an alert.

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE No
1.	INTRODUCTION	1
	1.1 INTRODUCTION TO PROJECT	1
	1.2 DOMAIN DESCRIPTION	2
2.	AIM AND SCOPE OF THE PRESENT INVESTIGATION	8
	2.1 AIM	8
	2.2 PROBLEM STATEMENT	8
	2.3 PROPOSE SYSTEM	8
	2.4 SCOPE	
3.	EXPERIMENTAL METHODS AND ALGORITHMS USED	7
	3.1 ALGORITHMS	7
	3.2 FEASIBILITY STUDY	8
	3.3 SUMMARY	8
4.	RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS	19
	4.1 REQUIREMENTS	19
	4.2 ABOUT OPENCV PACKAGE	19
	4.3 RESULTS AND DISCUSSION	22
5.	CONCLUSION AND FUTURE WORK	27
	REFERENCES	28

APPENDIX	30
A.SCREENSHOTS	30
B. SOURCE CODE	32

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
3.1	HAAR-LIKE FEATURE	9
3.3	SYSTEM ARCHITECTURE	14
3.3	USE CASE DIAGRAM	16
3.3	SEQUENCE DIAGRAM	17
3.3	ACTIVITY DIAGRAM	18
4.2	ANACONDA DISTRIBUTION	21

CHAPTER 1

INTRODUCTION

In recent years there is increase in computing power have brought computer vision to consumer-grade applications. As computers offer more and more processing power, the goal of real-time traffic sign detection and recognition is becoming feasible. Traffic management has become an important daily routine in cities today with the exponential growth of traffic on roads. Automatic vehicle detection and counting from traffic scenes and extracting essential parameters related to vehicular traffic can help better management of traffic on busy highways and road intersections. The article is dedicated to the presentation of a vision-based system for road vehicle counting and classification. The system is able to achieve counting with a very good accuracy.

1.1 INTRODUCTION TO PROJECT

Due to the increasing of road networks and the number of vehicles, traffic monitoring becomes very important. This leads to a quick development of the Intelligent Transportation System (ITS) technologies. Compared to conventional techniques such as microwave detectors, vision-based systems acquire more information about the vehicles. One of this information is vehicle counting and determining the traffic flow. This can be very useful for traffic monitoring and guide the drivers for better driving routes through bypassing the congested roads. Why would you want to count vehicles on the road? Here are some reasons:

Traffic management and planning:

If you have a good sense of the volume of traffic moving along a given road or network of roads, you can better understand congestion and then manage and/or make plans to reduce/eliminate it. Vehicle count data is very useful to urban city planners and transport authorities.

Traffic control: No one likes to be stuck behind a red light especially when the road is free. Vehicle counting systems can be integrated with traffic light control software to intelligently direct vehicles based on the current traffic situation in real time.

Parking management: A VCS can be installed at the entrance of a parking lot to monitor vehicles coming in and going out in order to determine whether there are slots available at any given time. It can also be used to ensure the number of vehicles in a given place (such as a hotel or events center) does not exceed its capacity by controlling Automatic Barrier Gates as opposed to issuing tags.

1.2 DOMAIN DESCRIPTION

Machine Learning

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves. The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

Machine Learning Methods

Machine learning algorithms are often categorized as supervised or unsupervised.

Supervised machine learning algorithms can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.

In contrast, **unsupervised machine learning algorithms** are used when the

information used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system doesn't figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.

Semi-supervised machine learning algorithms fall somewhere in between supervised and unsupervised learning, since they use both labeled and unlabeled data for training – typically a small amount of labeled data and a large amount of unlabeled data. The systems that use this method are able to considerably improve learning accuracy. Usually, semi-supervised learning is chosen when the acquired labeled data requires skilled and relevant resources in order to train it / learn from it. Otherwise, acquiring unlabeled data generally doesn't require additional resources.

Reinforcement machine learning algorithms is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behaviour within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which action is best; this is known as the reinforcement signal.

Advantages of Machine Learning

Machine Learning undoubtedly helps people to work more creatively and efficiently. Basically, you too can delegate quite complex or monotonous work to the computer through Machine Learning - starting with scanning, saving and filing paper documents such as invoices up to organizing and editing images.

In addition to these rather simple tasks, self-learning machines can also perform complex tasks. These include, for example, the recognition of error patterns. This is a major advantage, especially in areas such as the manufacturing industry: the industry relies on continuous and error-free production. While even experts often cannot be sure where and by which correlation a production error in a plant fleet arises, Machine Learning offers the possibility to identify the error early this saves down times and money. Self-learning programs are now also used in the medical field. In the future, after "consuming" huge amounts of data (medical publications, studies, etc.), apps will be able to warn a in case his doctor wants to prescribe a drug that he cannot tolerate.

This "knowledge" also means that the app can propose alternative options which for example also take into account the genetic requirements of the respective patient.

Applications of Machine Learning

1. Virtual Personal Assistants

Siri, Alexa, Google Now are some of the popular examples of virtual personal assistants. As the name suggests, they assist in finding information, when asked over voice. All you need to do is activate them and ask "What is my schedule for today?", "What are the flights from Germany to London", or similar questions. For answering, your personal assistant looks out for the information, recalls your related queries, or send a command to other resources (like phone apps) to collect info. You can even instruct assistants for certain tasks like "Set an alarm for 6 AM next morning", "Remind me to visit Visa Office day after tomorrow". Machine learning is an important part of these personal assistants as they collect and refine the information on the basis of your previous involvement with them. Later, this set of data is utilized to render results that are tailored to your preferences. Virtual Assistants are integrated to a variety of platforms. For example: Smart Speakers: Amazon Echo and Google Home Smartphones: Samsung Bixby on Samsung S8 Mobile Apps: Google Allo.

2. Predictions while Commuting

Traffic Predictions: We all have been using GPS navigation services. While we do that, our current locations and velocities are being saved at a central server for managing traffic. This data is then used to build a map of current traffic. While this helps in preventing the traffic and does congestion analysis, the underlying problem is that there are less number of cars that are equipped with GPS. Machine learning in such scenarios helps to estimate the regions where congestion can be found on the basis of daily experiences.

Online Transportation Networks: When booking a cab, the app estimates the price of the ride. When sharing these services, how do they minimize the detours? The answer is machine learning. Jeff Schneider, the engineering lead at Uber ATC reveals in an interview that they use ML to define price surge hours by predicting the rider demand. In the entire cycle of the services, ML is playing a major role.

3. Videos Surveillance

Imagine a single person monitoring multiple video cameras! Certainly, a difficult job to do and boring as well. This is why the idea of training computers to do this job makes sense. The video surveillance system nowadays are powered by AI that makes it possible to detect crime before they happen. They track unusual behaviour of people like standing motionless for a long time, stumbling, or napping on benches etc. The system can thus give an alert to human attendants, which can ultimately help to avoid mishaps. And when such activities are reported and counted to be true, they help to improve the surveillance services. This happens with machine learning doing its job at the backend.

4. Social Media Services

From personalizing your news feed to better ads targeting, social media platforms are utilizing machine learning for their own and user benefits. Here are a few examples that you must be noticing, using, and loving in your social media accounts, without realizing that these wonderful features are nothing but the applications of ML. People You May Know: Machine learning works on a simple concept: understanding with experiences. Facebook continuously notices the friends that you connect with, the profiles that you visit very often, your interests, workplace, or a group that you share with someone etc. On the basis of continuous learning, a list of Facebook users are suggested that you can become friends with. Face Recognition: You upload a picture of you with a friend and Facebook instantly recognizes that friend. Facebook checks the poses and projections in the picture, notice the unique features, and then match them with the people in your friend list. The entire process at the backend is complicated and takes care of the precision factor but seems to be a simple application of ML at the front end. Similar Pins: Machine learning is the core element of Computer Vision, which is a technique to extract useful information from images and videos. Pinterest uses computer vision to identify the objects (or pins) in the images and recommend similar pins accordingly.

5. Email Spam and Malware Filtering

There are a number of spam filtering approaches that email clients use. To ascertain that these spam filters are continuously updated, they are powered by machine

learning. When rule-based spam filtering is done, it fails to track the latest tricks adopted by spammers. Multi Layer Perceptron, C 4.5 Decision Tree Induction are some of the spam filtering techniques that are powered by ML. Over 325, 000 malwares are detected everyday and each piece of code is 90–98% similar to its previous versions. The system security programs that are powered by machine learning understand the coding pattern. Therefore, they detect new malware with 2–10% variation easily and offer protection against them.

6. Online Customer Support

A number of websites nowadays offer the option to chat with customer support representative while they are navigating within the site. However, not every website has a live executive to answer your queries. In most of the cases, you talk to a chatbot. These bots tend to extract information from the website and present it to the customers. Meanwhile, the chatbots advance with time. They tend to understand the user queries better and serve them with better answers, which is possible due to its machine learning algorithms.

7. Search Engine Result Refining

Google and other search engines use machine learning to improve the search results for you. Every time you execute a search, the algorithms at the backend keep a watch at how you respond to the results. If you open the top results and stay on the web page for long, the search engine assumes that the results it displayed were in accordance to the query. Similarly, if you reach the second or third page of the search results but do not open any of the results, the search engine estimates that the results served did not match requirement. This way, the algorithms working at the backend improve the search results.

8. Product Recommendations

You shopped for a product online few days back and then you keep receiving emails for shopping suggestions. If not this, then you might have noticed that the shopping website or the app recommends you some items that somehow matches with your taste. Certainly, this refines the shopping experience but did you know that it's machine learning doing the magic for you? On the basis of your behaviour with the website/app, past purchases, items liked or added to cart, brand preferences etc., the product recommendations are made.

9. Online Fraud Detection

Machine learning is proving its potential to make cyberspace a secure place and tracking monetary frauds online is one of its examples. For example: Paypal is using ML for protection against money laundering. The company uses a set of tools that helps them to compare millions of transactions taking place and distinguish between legitimate or illegitimate transactions taking place between the buyers and sellers.

CHAPTER 2

AIM AND SCOPE OF THE PRESENT SYSTEM

2.1 AIM

In existing system a vehicle counting method based on Single Shot Multi Box Detector (SSD). We perform vehicle detection, vehicle tracking and counting tasks via SSD. The SSD has the capacity to extract meaningful features of blob of vehicles, and predicting the location and classification of vehicles.

2.2 PROBLEM STATEMENT

This system need of complex data augmentation also suggests it needs a large number of data to train.

The information flows unidirectional and the classifier network cannot utilize other directional information.

2.3 PROPOSED SYSTEM:

- 1.Detection and counting of the vehicles in traffic signals..
- 2.Predicting the scene class based on the spatial relationships among the vehicles of interest and contextually important objects.

2.4 SCOPE

This uses machine learning techniques to get a high degree of accuracy from what is called “training data”. Haar Cascades use the Adaboost learning algorithm which selects a small number of important features from a large set to give an efficient result

CHAPTER 3

EXPERIMENTAL METHODS AND ALGORITHMS

3.1 ALGORITHM

Haar Cascade Algorithm

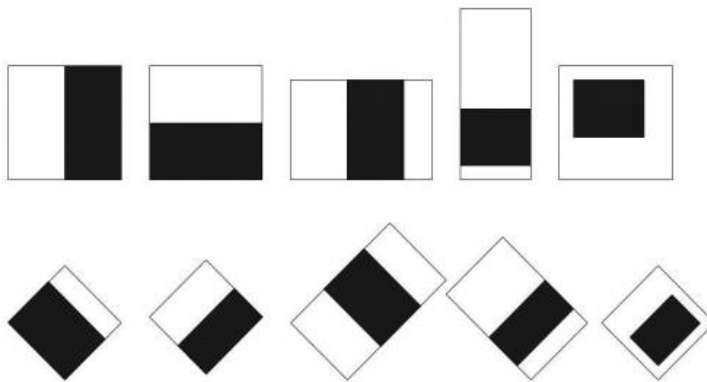
Haar Cascade is a machine learning object detection algorithm used to identify objects in an image or video.

The algorithm has four stages:

1. Haar Feature Selection
2. Creating Integral Images
3. Adaboost Training
4. Cascading Classifier

Haar-Like Feature

Haar-Like is a rectangular simple feature that is used as an input feature for cascaded classifier. In there are some filters based on Haar-Like feature. By applying every one of these filters into one special area of the image, the pixel sums under white areas are subtracted from the pixel sums under the black areas. That is the weight of white and black area can be considered as "1" and "-1", respectively.



AdaBoost Algorithm

AdaBoost algorithm (a machine learning meta-algorithm) in choosing features and improving the performance is repeatedly used. AdaBoost in order to construct a strong classifier combined many weak classifiers.

AdaBoost in the way that mixes a series of AdaBoost classifiers as a filter chain. Each

filter is a separate AdaBoost classifier which consists of a few weak classifiers. If each of these filters in the acceptance region of the image shows the vehicle fails, this area is immediately classified as a non-vehicle. When a filter accepts an area of image as a vehicle, the area enters the next filter in the chain. If this area of the image passes all the chain filters successfully, it is classified as vehicle. In this algorithm, each cycle of boosting a feature among all other potential features is selected and in the end, the final classification will be a linear combines of the initial weak classification.

Integral Image

Integral image is a quick method for calculating the Haar-Like feature. has used this technique and they recognized which Haar-Like feature among the other image.

Integral image is sum of all pixel values. With integral image Haar-Like feature can be calculated quickly by simple addition and subtraction.

Integral images can be defined as two-dimensional lookup tables in the form of a matrix with the same size of the original image.

Cascaded Classifier

Cascaded classifier is used for fast rejection of error windows and improving the processing speed.

In every node of trees there is a non-vehicle branching, it means that the image will not be vehicle. By this technique the false negative rate is at least.

The cascade classifier consists of a collection of stages, where each stage is an ensemble of weak learners. If the label is positive, the classifier passes the region to the next stage. The detector reports an object found at the current window location when the final stage classifies the region as positive.

Advantage of haar cascade

- It achieves a good performance for both accuracy and time-complexity.
- calculation speed due to the use of integral images
- Quick detection and high efficiency

3.2 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

1. Economic feasibility
2. Technical feasibility
3. Social feasibility

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands being placed on the client. A feasibility study evaluates the project's potential for success.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

3.3 SUMMARY

Python Language

Python is an object-oriented programming language created by Guido Rossum in 1989. It is ideally designed for rapid prototyping of complex applications. It has interfaces to many OS system calls and libraries and is extensible to C or C++. Many large companies use the Python programming language include NASA, Google, YouTube, BitTorrent, etc. Python programming is widely used in Artificial Intelligence, Natural Language Generation, Neural Networks and other advanced fields of Computer Science. Python had deep focus on code readability & this class will teach you python from basics.

Python Programming Characteristics

- It provides rich data types and easier to read syntax than any other programming languages
- It is a platform independent scripted language with full access to operating system API's
- Compared to other programming languages, it allows more run-time flexibility
- It includes the basic text manipulation facilities of Perl and Awk
- A module in Python may have one or more classes and free functions
- Libraries in Python are cross-platform compatible with Linux, Macintosh, and Windows
- For building large applications, Python can be compiled to byte-code
- Python supports functional and structured programming as well as OOP
- It supports interactive mode that allows interacting [Testing](#) and debugging of snippets of code
- In Python, since there is no compilation step, editing, debugging and testing is fast.

Applications of Python Programming

Web Applications

You can create scalable Web Apps using frameworks and CMS (Content Management System) that are built on Python. Some of the popular platforms for creating Web Apps are: Django, Flask, Pyramid, Plone, Django CMS. Sites like Mozilla, Reddit, Instagram and PBS are written in Python.

Scientific and Numeric Computing

There are numerous libraries available in Python for scientific and numeric computing. There are libraries like: SciPy and NumPy that are used in general purpose computing. And, there are specific libraries like: EarthPy for earth science, AstroPy for Astronomy and so on. Also, the language is heavily used in machine learning, data mining and deep learning.

Creating software Prototypes

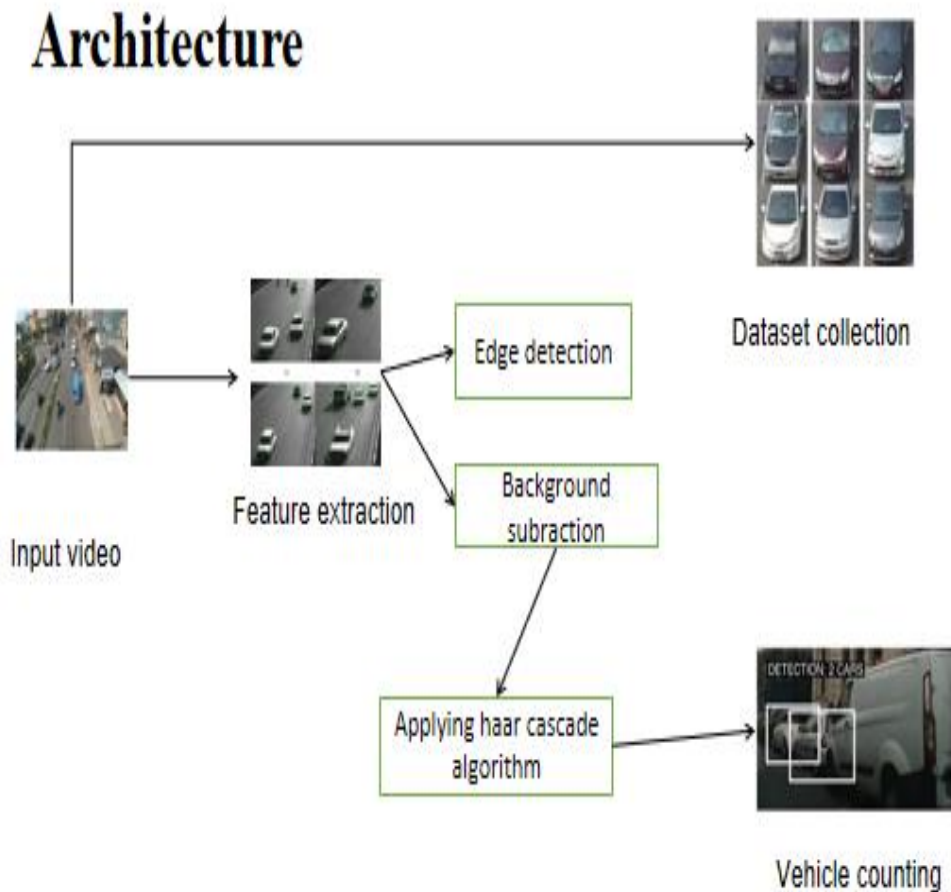
Python is slow compared to compiled languages like C++ and Java. It might not be a good choice if resources are limited and efficiency is a must. However, Python is a great language for creating prototypes. For example: You can use Pygame (library for creating games) to create your game's prototype first. If you like the prototype, you can use language like C++ to create the actual game.

Good Language to Teach Programming

Python is used by many companies to teach programming to kids and newbies. It is a good language with a lot of features and capabilities. Yet, it's one of the easiest language to learn because of its simple easy-to-use syntax.

SYSTEM DESIGN

SYSTEM ARCHITECTURE



UML DIAGRAMS

- UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.
- The goal is for UML to become a common language for creating models of

object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

- The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.
- The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.
- The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

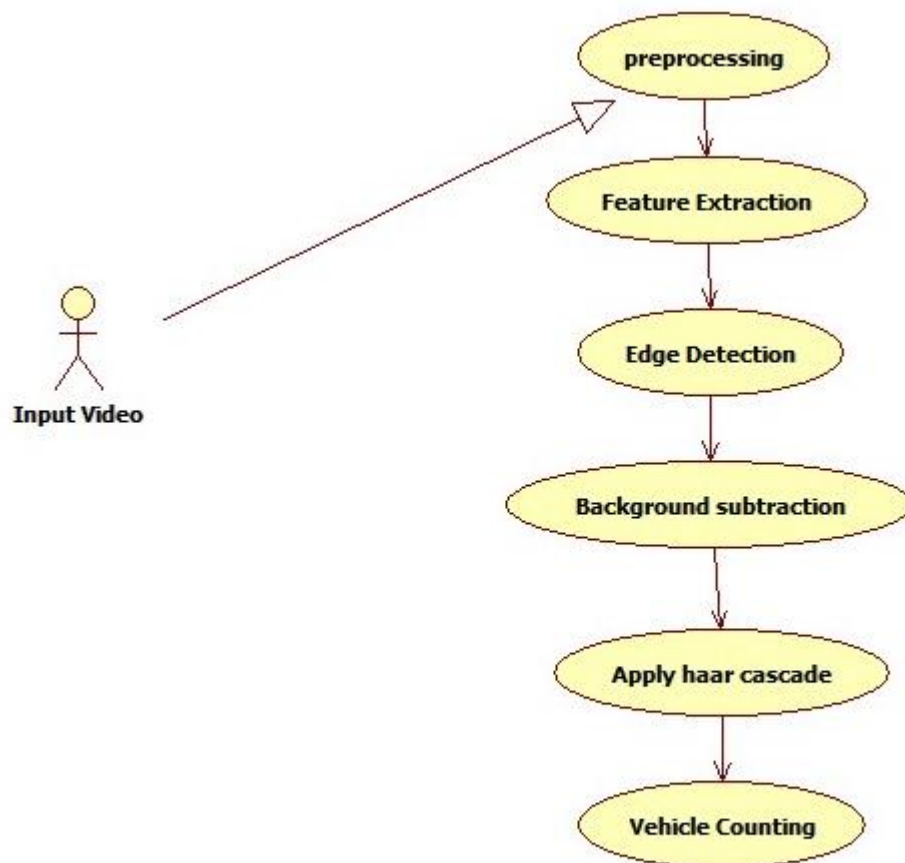
The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

USE CASE DIAGRAM:

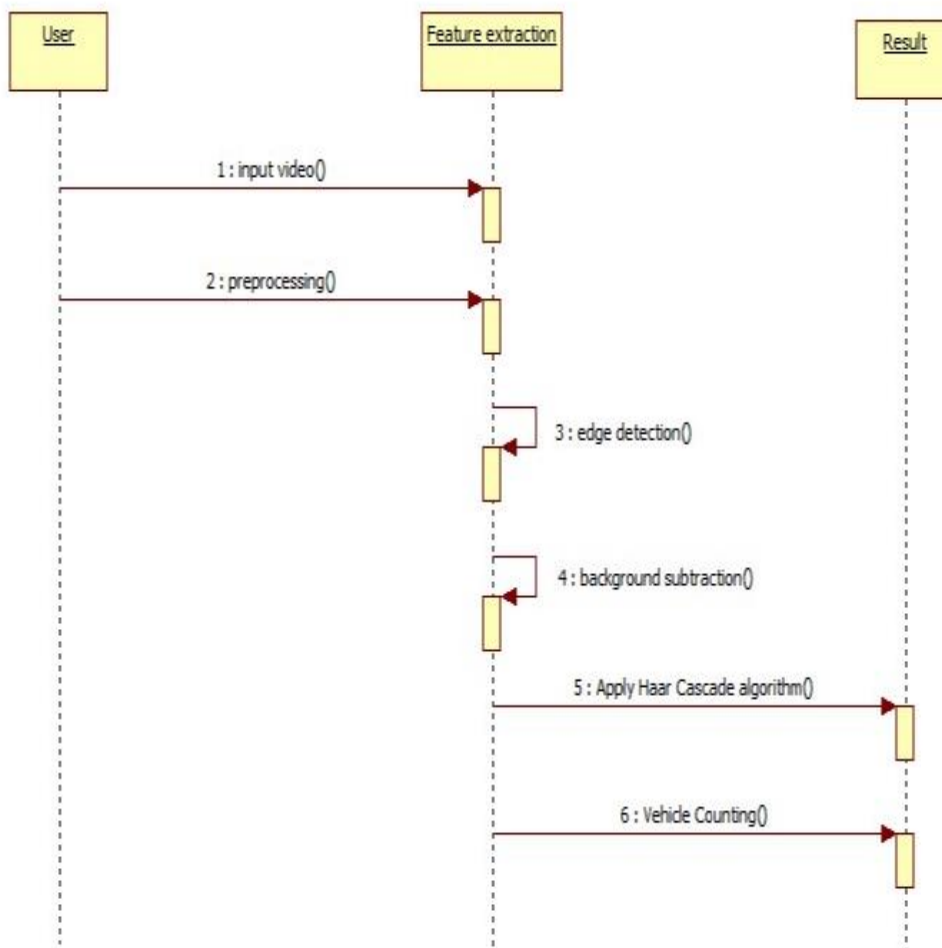
A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions

are performed for which actor. Roles of the actors in the system can be depicted.



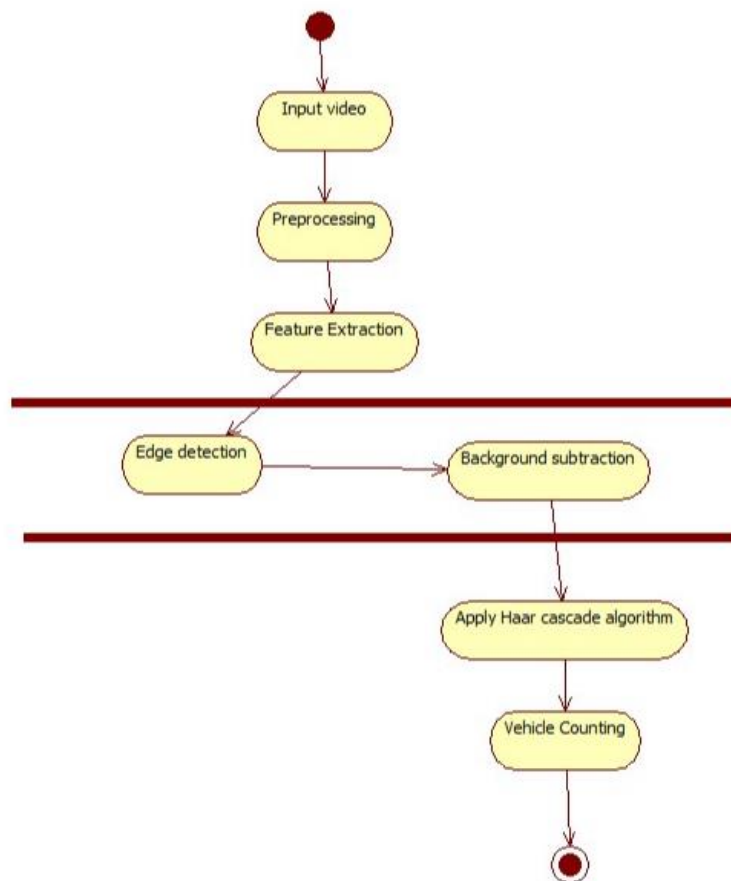
SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



CHAPTER 4

SYSTEM IMPLEMENTATION

4.1 REQUIREMENTS

I. HARDWARE REQUIREMENTS

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list, especially in case of operating systems. The minimal hardware requirements are as follows,

- ☐PROCESSOR : PENTIUM IV
- ☐RAM : 8 GB
- ☐PROCESSOR : 2.4 GHZ
- ☐MAIN MEMORY : 8GB RAM
- ☐PROCESSING SPEED : 600 MHZ
- ☐HARD DISK DRIVE : 1TB
- ☐KEYBOARD :104 KEYS

II. SOFTWARE REQUIREMENTS

Software requirements deals with defining resource requirements and prerequisites that needs to be installed on a computer to provide functioning of an application. These requirements are need to be installed separately before the software is installed. The minimal software requirements are as follows,

- ☐FRONT END :PYTHON
- ☐IDE : ANACONDA
- ☐OPERATING SYSTEM :WINDOWS 10

4.2 ABOUT OPENCV PACKAGE

Python is a general purpose programming language started by Guido van Rossum, which became very popular in short time mainly because of its simplicity and code readability. It enables the programmer to express his ideas in fewer lines of code without reducing any readability.

Compared to other languages like C/C++, Python is slower. But another important feature of Python is that it can be easily extended with C/C++. This feature helps us to write computationally intensive codes in C/C++ and create a Python wrapper for it so that we can use these wrappers as Python modules. This gives us two advantages: first, our code is as fast as original C/C++ code (since it is the actual C++ code working in background) and second, it is very easy to code in Python. This is how OpenCV-Python works, it is a Python wrapper around original C++ implementation.

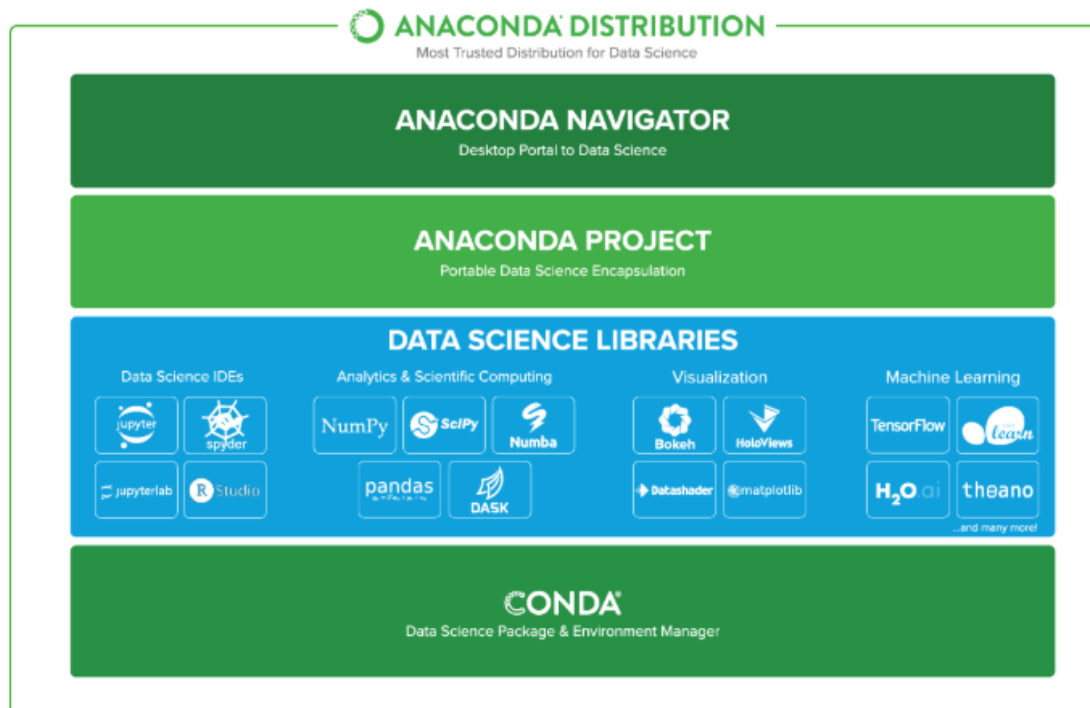
And the support of Numpy makes the task more easier. Numpy is a highly optimized library for numerical operations. It gives a MATLAB-style syntax. All the OpenCV array structures are converted to-and-from Numpy arrays. So whatever operations you can do in Numpy, you can combine it with OpenCV, which increases number of weapons in your arsenal. Besides that, several other libraries like SciPy, Matplotlib which supports Numpy can be used with this.

So OpenCV-Python is an appropriate tool for fast prototyping of computer vision problems.

FEATURES OF ANACONDA NAVIGATOR

Anaconda is a free and open source, easy to install distribution of Python and R programming languages. Anaconda provides a working environment which is used for scientific computing, data science, statistical analysis and machine learning.

The latest distribution of Anaconda is Anaconda 5.3 and is released in October, 2018. It has the conda package, environment manager and a collection of 1000+ open source packages long with free community support.



What is Anaconda Navigator?

Anaconda Navigator is a desktop graphical user interface (GUI) included in the Anaconda distribution. It allows us to launch applications provided in the Anaconda distribution and easily manage conda packages, environments and channels without the use of command-line commands. It is available for Windows, macOS and Linux.

4.3 RESULTS AND DISCUSSION

Data Preprocessing

It is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis.

Data goes through a series of steps during preprocessing:

Data Cleaning: Data is cleansed through processes such as filling in missing values, smoothing the noisy data, or resolving the inconsistencies in the data.

Data Integration: Data with different representations are put together and conflicts within the data are resolved.

Data Transformation: Data is normalized, aggregated and generalized.

Data Reduction: This step aims to present a reduced representation of the data in a data warehouse.

Feature extraction

It is the process of transforming the raw pixel values from an image, to a more meaningful and useful information that can be used in other techniques, such as point matching or machine learning.

In feature extraction module, we will characterize the boxes associated with vehicles by a set of

parameters. This step aims to perform two important tasks:

- Classify vehicles according to their types.
- Detect partial occlusions of vehicles.

Classification process is based on preliminary study applied to a vehicle image database in order to determine feature values associated with vehicle types. Thus, each vehicle detected in the first phase is classified depending on the values of the calculated parameters. In the last module, these vehicles are counted during their movement in the scene.

Background Subtraction

Background subtraction is a concept that belongs to areas like image processing and

computer vision. Background subtraction is also known as the foreground detection. This technique is widely used to detect moving objects from video. In this technique, foreground of an image is extracted for further processing. It is the process of separating out foreground object of an image from the background in frames of video sequence.

Edges essentially separate two various regions which are static region (the roadway) and dynamic region (moving vehicles). The static background is then deleted to locate moving objects in each frame. The result zone leaves only vehicles and some details as moving objects in sequential images which are changing frame to frame.

Edge Detection

Each image (video frame) has three significant features to achieve detection goals. These features include: edges, contours and points. Among mentioned features, an appropriate option is to use edge pixels. Processing of image pixels enables us to find edge pixels, which are the main features of passing vehicles in a roadway video frame. Edge detection process is demonstrated in a binary image (threshold) with the detected edge pixels.

The next step is to extract moving edges from sequential video frames and process the resulting edge information to obtain quantitative geometric measurements of passing vehicles.

Sliding Windows

Object detection using sliding window machine learning algorithm, first requirement of sliding window algorithm is to prepare labeled training set. Sliding window is a problem solving technique for problems involves arrays/lists.

The idea is to train a binary classifier, which determines if the presented object is “positive” or “negative.” The trained classifier can then be used to “inspect” a target image by sampling it, starting from the top-left corner

SOFTWARE TESTING

General

In a generalized way, we can say that the system testing is a type of testing in which the main aim is to make sure that system performs efficiently and seamlessly. The process of testing is applied to a program with the main aim to discover an unprecedented error, an error which otherwise could have damaged the future of the software. Test cases which brings up a high possibility of discovering and error is considered successful. This successful test helps to answer the still unknown errors.

TEST CASE

Testing, as already explained earlier, is the process of discovering all possible weak-points in the finalized software product. Testing helps to counter the working of sub-assemblies, components, assembly and the complete result. The software is taken through different exercises with the main aim of making sure that software meets the business requirement and user-expectations and doesn't fails abruptly. Several types of tests are used today. Each test type addresses a specific testing requirement.

Testing Techniques

A test plan is a document which describes approach, its scope, its resources and the schedule of aimed testing exercises. It helps to identify almost other test item, the features which are to be tested, its tasks, how will everyone do each task, how much the tester is independent, the environment in which the test is taking place, its technique of design plus the both the end criteria which is used, also rational of choice of theirs, and whatever kind of risk which requires emergency planning. It can be also referred to as the record of the process of test planning. Test plans are usually prepared with signification input from test engineers.

(I) UNIT TESTING

In unit testing, the design of the test cases is involved that helps in the validation of the internal program logic. The validation of all the decision branches and internal code takes place. After the individual unit is completed it takes place. Plus it is taken into account after the individual unit is completed before integration. The unit test thus performs the basic level test at its component stage and test the particular business process, system configurations etc. The unit test ensures that the particular unique path of the process gets performed precisely to the documented specifications and

contains clearly defined inputs with the results which are expected.

(II) INTEGRATION TESTING

These tests are designed to test the integrated software items to determine whether if they really execute as a single program or application. The testing is event driven and thus is concerned with the basic outcome of field. The Integration tests demonstrate that the components were individually satisfaction, as already represented by successful unit testing, the components are apt and fine. This type of testing is specially aimed to expose the issues that come-up by the components combination.

(III) FUNCTIONAL TESTING

The functional tests help in providing the systematic representation that functions tested are available and specified by technical requirement, documentation of the system and the user manual.

(IV) SYSTEM TESTING

System testing, as the name suggests, is the type of testing in which ensure that the software system meet the business requirements and aim. Testing of the configuration is taken place here to ensure predictable result and thus analysis of it. System testing is relied on the description of process and its flow, stressing on pre driven process and the points of integration.

(V) WHITE BOX TESTING

The white box testing is the type of testing in which the internal components of the system software is open and can be processed by the tester. It is therefore a complex type of testing process. All the data structure, components etc. are tested by the tester himself to find out a possible bug or error. It is used in situation in which the black box is incapable of finding out a bug. It is a complex type of testing which takes more time to get applied.

(VI) BLACK BOX TESTING

The black box testing is the type of testing in which the internal components of the software is hidden and only the input and output of the system is the key for the tester to find out a bug. It is therefore a simple type of testing. A programmer with basic knowledge can also process this type of testing. It is less time consuming as compared to the white box testing. It is very successful for software which are less complex are

straight-forward in nature. It is also less costly than white box testing.

(V) ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

CHAPTER 5

CONCLUSION AND FUTUREWORK

5.1 Conclusion

Vehicle counting is considered one of the most important applications in traffic control and management. To count vehicles, synchronous vehicle detection and tracking should be carried out. Preliminary results for developing an automated system for counting and classification of vehicles in motion, based on machine learning techniques were presented in this paper. The developed system was able to count vehicles and classify them with a reasonable accuracy.

5.2 Future Work

In Future sensors, processors and CCTV cameras can be implemented with increased image resolution provides the basis for a continuous growth of this field.

REFERENCES

1.Accurate Vehicle Counting Approach Based on Deep Neural Networks, Mohamed A. Abdelwahab, 2019 International Conference on Innovative Trends in Computer Engineering (ITCE)

2.Implementation of Pixel Based Adaptive Segmenter method for tracking and counting vehicles in visual surveillance, Muhammad Brilliant Subaweh ; Eri Prasetyo Wibowo, 2016 International Conference on Informatics and Computing (ICIC)

3.Vehicle Count System based on Time Interval Image Capture Method and Deep Learning Mask R-CNN, Eduardo Jr Piedad ; Tuan-Tang Le ; Kimberly Aying ; Fhenyl Kristel Pama ; Ianny Tabale, TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)

4.Vehicle counting system in real-time, Salma Bouaich ; Mohamed Adnane Mahraz ; Jamal Riffi ; Hamid Tairi, 2018 International Conference on Intelligent Systems and Computer Vision (ISCV)

5.An SSD Algorithm Based on Vehicle Counting Method, Qiaoqian Chen ; Na Huang ; Jieming Zhou ; Zhao Tan, 2018 37th Chinese Control Conference (CCC)

6.Fast approach for efficient vehicle countingM.A. Abdelwahab, Electronics Letters,Year: 2019 | Volume: 55, Issue: 1 | Journal Article | Publisher: IET

7.A real time vehicle counting based on adaptive tracking approach for highway videos, Muhamad Soleh ; Grafika Jati ; Ananto Tri Sasongko ; Wisnu Jatmiko ; M H Hilman, 2017 International Workshop on Big Data and Information Security (IWBIS)

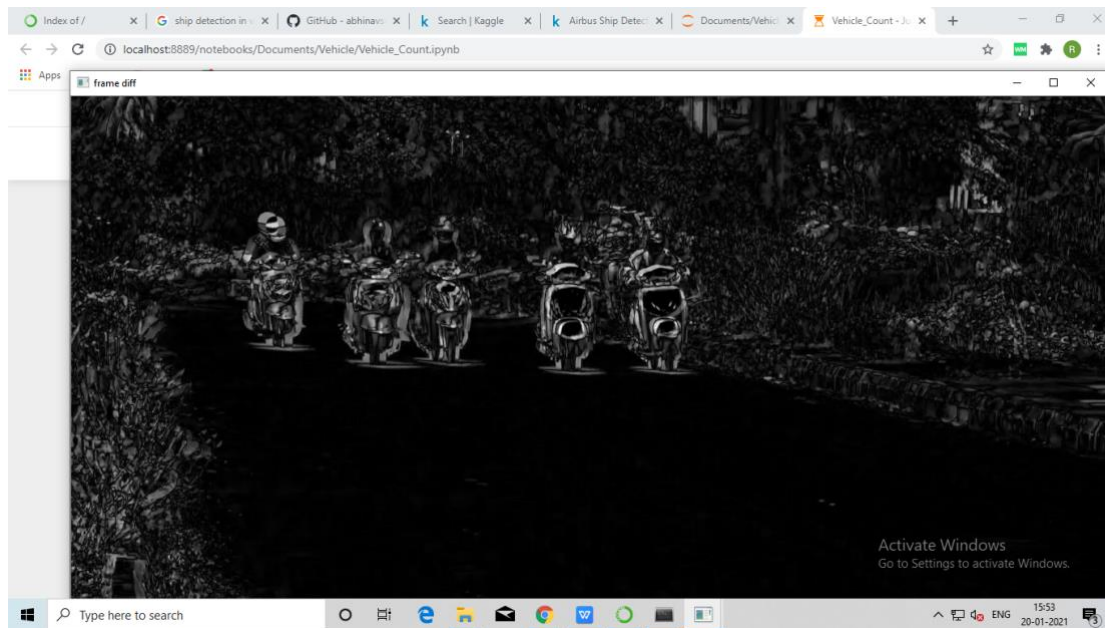
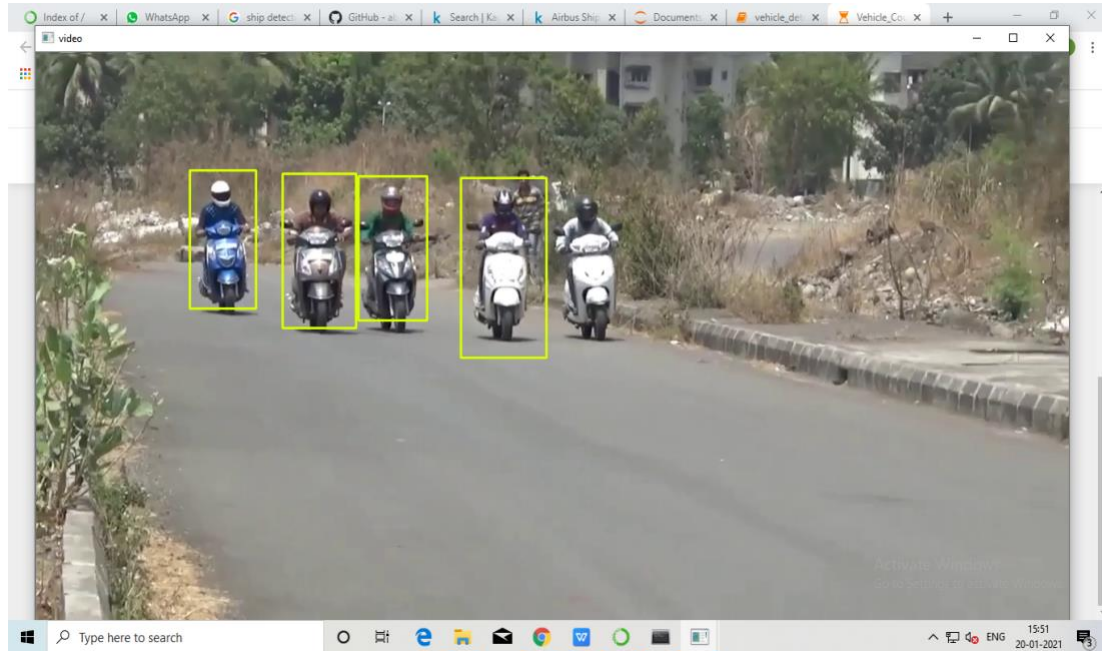
8.Video-Based Vehicle Counting Framework, Zhe Dai ; Huansheng Song ; Xuan Wang ; Yong Fang ; Xu Yun ; Zhaoyang Zhang ; Huaiyu Li, IEEE Access, Year: 2019

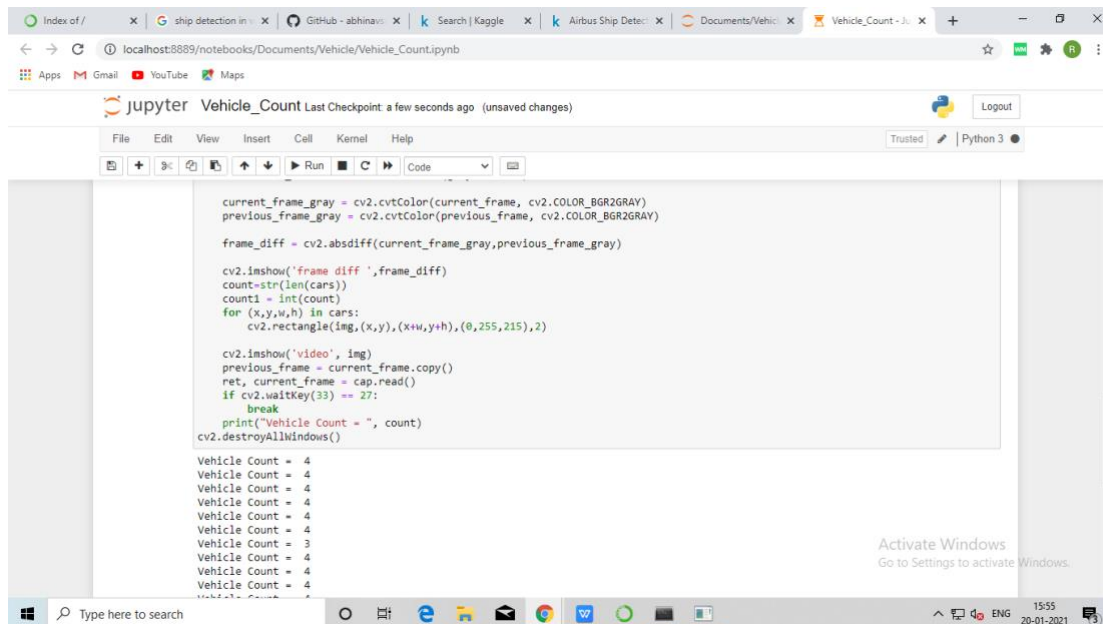
9. Kalman Filter-based Vehicle Count Estimation Approach Using Probe Data: A Multi-lane Road Case Study, Mohammad A. Aljamal ; Hossam M. Abdelghaffar ; Hesham A. Rakha, 2019 IEEE Intelligent Transportation Systems Conference (ITSC)

10. Compressed-Domain Highway Vehicle Counting by Spatial and Temporal Regression, Zilei Wang ; Xu Liu ; Jiashi Feng ; Jian Yang ; HongSheng Xi , 2019 IEEE Transactions on Circuits and Systems for Video Technology

APPENDIX

A.SCREENSHOTS





B.SOURCE CODE

```
import cv2

cascade_src = r'C:\Users\RE cs 2\Documents\Vehicle\Bike
Detection\two_wheeler.xml'
video_src = r'C:\Users\RE cs 2\Documents\Vehicle\Bike
Detection\two_wheeler2.mp4'

cap = cv2.VideoCapture(video_src)
ret, current_frame = cap.read()
car_cascade = cv2.CascadeClassifier(cascade_src)
previous_frame = current_frame
fgbg = cv2.createBackgroundSubtractorMOG2()
while True:
    ret, img = cap.read()
    fgmask = fgbg.apply(img)
    cv2.imshow('Backgrond Subtraction',fgmask)
    if (type(img) == type(None)):
        break

    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    cars = car_cascade.detectMultiScale(gray,1.01, 1)

    current_frame_gray = cv2.cvtColor(current_frame, cv2.COLOR_BGR2GRAY)
    previous_frame_gray = cv2.cvtColor(previous_frame, cv2.COLOR_BGR2GRAY)

    frame_diff = cv2.absdiff(current_frame_gray,previous_frame_gray)

    cv2.imshow('frame diff ',frame_diff)
    count=str(len(cars))
    count1 = int(count)
    for (x,y,w,h) in cars:
        cv2.rectangle(img,(x,y),(x+w,y+h),(0,255,215),2)

    cv2.imshow('video', img)
    previous_frame = current_frame.copy()
    ret, current_frame = cap.read()
    if cv2.waitKey(33) == 27:
        break
    if count1>5:
        print("Density Detected = ", count1)
cv2.destroyAllWindows()

import cv2
```

```
cascade_src = r'C:\Users\RE cs 2\Documents\Shahana 2019-2020\ML Project  
Correct\Vehicle\Bike Detection\two_wheeler.xml'
```

```
video_src = r'C:\Users\RE cs 2\Documents\Shahana 2019-2020\ML Project  
Correct\Vehicle\Bike Detection\two_wheeler2.mp4'
```

```
cap = cv2.VideoCapture(video_src)
```

```
car_cascade = cv2.CascadeClassifier(cascade_src)
```

```
while True:
```

```
    ret, img = cap.read()
```

```
    if (type(img) == type(None)):  
        break
```

```
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

```
    cars = car_cascade.detectMultiScale(gray,1.01, 1)
```

```
    for (x,y,w,h) in cars:  
        cv2.rectangle(img,(x,y),(x+w,y+h),(0,255,215),2)
```

```
    cv2.imshow('video', img)
```

```
    if cv2.waitKey(33) == 27:  
        break
```

```
cv2.destroyAllWindows()
```

```
#bus detection
```

```
import cv2
```

```
cascade_src = r'C:\Users\RETECH-CSC 04\Downloads\Vehicle-And-Pedestrian-  
Detection-Using-Haar-Cascades-master\Main Project\Main Project\Bus  
Detection\Bus_front.xml'
```

```
video_src = r'C:\Users\RETECH-CSC 04\Downloads\Vehicle-And-Pedestrian-  
Detection-Using-Haar-Cascades-master\Main Project\Main Project\Bus  
Detection\bus1.mp4'
```

```
cap = cv2.VideoCapture(video_src)
```

```
car_cascade = cv2.CascadeClassifier(cascade_src)
```

```
while True:
```

```

ret, img = cap.read()

if (type(img) == type(None)):
    break

gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

cars = car_cascade.detectMultiScale(gray, 1.16, 1)

for (x,y,w,h) in cars:
    cv2.rectangle(img,(x,y),(x+w,y+h),(0,0,255),2)

cv2.imshow('video', img)

if cv2.waitKey(33) == 27:
    break

cv2.destroyAllWindows()
#car detection
# -*- coding: utf-8 -*-

import cv2

cascade_src = r'C:\Users\RETECH-CSC 04\Downloads\Vehicle-And-Pedestrian-
Detection-Using-Haar-Cascades-master\Main Project\Main Project\Car
Detection\cars.xml'

video_src = r'C:\Users\RETECH-CSC 04\Downloads\Vehicle-And-Pedestrian-
Detection-Using-Haar-Cascades-master\Main Project\Main Project\Car
Detection\video.avi'

cap = cv2.VideoCapture(video_src)

car_cascade = cv2.CascadeClassifier(cascade_src)

while True:
    ret, img = cap.read()

    if (type(img) == type(None)):
        break
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    cars = car_cascade.detectMultiScale(gray, 1.1, 2)

    for (x,y,w,h) in cars:

```

```
    cv2.rectangle(img,(x,y),(x+w,y+h),(0,255,255),2)

cv2.imshow('video', img)

if cv2.waitKey(33) == 27:
    break

cv2.destroyAllWindows()
```