**vehicle detection and speed detection using opencv and python**

**1.INTRODUCTION:**

Government of India aims to develop 100 smart cities in future. A Smart city delivers smart services like smart traffic management, traffic surveillance etc. To deliver these smart services, various information and communication technologies are used. The smart traffic monitoring is incomplete without a system that is capable of automatic detection of traffic rule violations. Automatic traffic surveillance system is the need of the smart traffic management. In urban areas, the detection of red light violations, speed limit violations and stop look and go protocol violations are the issues that usually arises. The detection of red light violation is generally manual process in India barring few cities where CCTV footage of traffic cameras is used for this. To detect the speed limit protocol violation, the speed guns are used. For smart city development, these issues need to be resolved. Smart city traffic surveillance system is the right solution to these issues. For detection of moving vehicles, detection of vehicles speed and automation recognition of number plates of the vehicles, various techniques have been proposed by many researchers but a comprehensive and cost effective solution is still missing. In the present era of computer vision, the detection of moving objects is intrinsic need of many image processing applications like traffic surveillance, vehicle classification, collision detection (such as accidents on roads) etc. There exist wide variety of methodologies for moving vehicle detection and tracking but efficient technique with higher accuracy and economy needs to be developed. In smart city traffic surveillance system1, these techniques may play important roles. This paper proposed an efficient and novel approach for detection of the moving vehicles and their speed. The proposed approach can be integrated with existing traffic monitoring based on cameras system without major modifications.

**1.1 Objective of the project:**

An intelligent traffic management and surveillance is the basic need for the smart city development in India. This includes the detection of moving vehicles, estimation of their speed and detection of the speed limit violation and its registration number. This paper proposes an efficient and novel approach for the detection of moving vehicles as well as estimation of their speeds by using a single camera in daylight or properly illuminated environment. The proposed approach detects and tracks the vehicle passing through the surveillance area and keeps the record of vehicles position. In this paper vehicles tracking is based on the relative positions of the vehicles in consecutive frames. This information may be used in the Automatic Number Plate Recognition (ANPR) System for selection of those key frames where speed limit violation occurs. The average detection accuracy achieved by proposed approach is about 87.7%. The proposed approach uses cropping operation to minimize the scope of any false positive detection on both sides of road.

**2.LITERATURE SURVEY:**

**Smart City Traffic Management and Surveillance System for Indian Scenario**

In India, speed and red light violation are the main cause of accidents. Over speed reduces our reaction time and it makes difficult to control a vehicle during obstacle moves into our path. Violating the red light increases our chances of crashing. At present in India to detect the over speed vehicles and red light violation there should be an efficient and automated system. In this paper we proposed the framework of smart city traffic management and surveillance system for automatic over speeding vehicle detection and red light violation detection system. The proposed model will help to detect the vehicles having over speed and red light violation. This proposed system is based on pattern recognition and image processing techniques.

**Scene Modeling and Change Detection in Dynamic Scenes: A Subspace Approach**

Background modeling and subtraction are core components in video processing. To this end, one aims to recover and continuously update a representation of the scene that is compared with the current input to perform subtraction. Most of the existing methods treat each pixel independently and attempt to model the background perturbation through statistical modeling such as a mixture of Gaussians. While such methods have satisfactory performance in many scenarios, they do not model the relationships and correlation amongst nearby pixels. Such correlation between pixels exists both in space and across time especially when the scene consists of image structures moving across space. Waving trees, beach, [escalators](https://www.sciencedirect.com/topics/engineering/escalators) and natural scenes with rain or snow are examples of such scenes. In this paper, we propose a method for differentiating between image structures and motion that are persistent and repeated from those that are “new”. Towards capturing the appearance characteristics of such scenes, we propose the use of an appropriate [subspace](https://www.sciencedirect.com/topics/engineering/subspace) created from image structures. Furthermore, the dynamical characteristics are captured by the use of a prediction mechanism in such subspace. Since the model must adapt to long-term changes in the background, an [incremental](https://www.sciencedirect.com/topics/engineering/incremental) method for fast online adaptation of the model parameters is proposed. Given such adaptive models, robust and meaningful measures for detection that consider both structural and motion changes are considered. Promising experimental results that include qualitative and [quantitative comparisons](https://www.sciencedirect.com/topics/computer-science/quantitative-comparison) with existing background modeling/subtraction techniques demonstrate the very promising performance of the proposed framework when dealing with complex backgrounds.

**Detection and Classification of Vehicles**

This paper presents algorithms for vision-based detection and classification of vehicles in monocular image sequences of traffic scenes recorded by a stationary camera. Processing is done at three levels: raw images, region level, and vehicle level. Vehicles are modeled as rectangular patches with certain dynamic behavior. The proposed method is based on the establishment of correspondences between regions and vehicles, as the vehicles move through the image sequence. Experimental results from highway scenes are provided which demonstrate the effectiveness of the method. We also briefly describe an interactive camera calibration tool that we have developed for recovering the camera parameters using features in the image selected by the user.

**Real Time Speed Estimation of Moving Vehicles from Side View Images from an Uncalibrated Video Camera**

In order to estimate the speed of a moving vehicle with side view camera images, velocity vectors of a sufficient number of reference points identified on the vehicle must be found using frame images. This procedure involves two main steps. In the first step, a sufficient number of points from the vehicle is selected, and these points must be accurately tracked on at least two successive video frames. In the second step, by using the displacement vectors of the tracked points and passed time, the velocity vectors of those points are computed. Computed velocity vectors are defined in the video image coordinate system and displacement vectors are measured by the means of pixel units. Then the magnitudes of the computed vectors in image space should be transformed to the object space to find the absolute values of these magnitudes. This transformation requires an image to object space information in a mathematical sense that is achieved by means of the calibration and orientation parameters of the video frame images. This paper presents proposed solutions for the problems of using side view camera images mentioned here.

**ViBe: A Universal Background Subtraction Algorithm for Video Sequences**

This paper presents a technique for motion detection that incorporates several innovative mechanisms. For example, our proposed technique stores, for each pixel, a set of values taken in the past at the same location or in the neighborhood. It then compares this set to the current pixel value in order to determine whether that pixel belongs to the background, and adapts the model by choosing randomly which values to substitute from the background model. This approach differs from those based upon the classical belief that the oldest values should be replaced first. Finally, when the pixel is found to be part of the background, its value is propagated into the background model of a neighboring pixel. We describe our method in full details (including pseudo-code and the parameter values used) and compare it to other background subtraction techniques. Efficiency figures show that our method outperforms recent and proven state-of-the-art methods in terms of both computation speed and detection rate. We also analyze the performance of a downscaled version of our algorithm to the absolute minimum of one comparison and one byte of memory per pixel. It appears that even such a simplified version of our algorithm performs better than mainstream techniques.

**Image Segmentation in Video Sequences: A Probabilistic Approach**

Background subtraction" is an old technique for finding moving objects in a video sequence for example, cars driving on a freeway. The idea is that subtracting the current image from a timeaveraged background image will leave only nonstationary objects. It is, however, a crude approximation to the task of classifying each pixel of the current image; it fails with slow-moving objects and does not distinguish shadows from moving objects. The basic idea of this paper is that we can classify each pixel using a model of how that pixel looks when it is part of different classes. We learn a mixture-of-Gaussians classification model for each pixel using an unsupervised technique- an efficient, incremental version of EM. Unlike the standard image-averaging approach, this automatically updates the mixture component for each class according to likelihood of membership; hence slow-moving objects are handled perfectly. Our approach also identifies and eliminates shadows much more effectively than other techniques such as thresholding. Application of this method as part of the Roadwatch traffic surveillance project is expected to result in significant improvements in vehicle identification and tracking.

**OR-PCA with Dynamic Feature Selection for Robust Background Subtraction**

Background modeling and foreground object detection is the first step in visual surveillance system. The task becomes more difficult when the background scene contains significant variations, such as water surface, waving trees and sudden illumination conditions, etc. Recently, subspace learning model such as Robust Principal Component Analysis (RPCA) provides a very nice framework for separating the moving objects from the stationary scenes. However, due to its batch optimization process, high dimensional data should be processed. As a result, huge computational complexity and memory problems occur in traditional RPCA based approaches. In contrast, Online Robust PCA (ORPCA) has the ability to process such large dimensional data via stochastic manners. OR-PCA processes one frame per time instance and updates the subspace basis accordingly when a new frame arrives. However, due to the lack of features, the sparse component of OR-PCA is not always robust to handle various background modeling challenges. As a consequence, the system shows a very weak performance, which is not desirable for real applications. To handle these challenges, this paper presents a multi-feature based ORPCA scheme. A multi-feature model is able to build a robust low-rank background model of the scene. In addition, a very nice feature selection process is designed to dynamically select a useful set of features frame by frame, according to the weighted sum of total features. Experimental results on challenging datasets such as Wallflower, I2R and BMC 2012 show that the proposed scheme outperforms the state of the art approaches for the background subtraction task.

**Model-Based Vehicle Detection and Classification using Orthographic Approximations**

This paper reports the current state of work to simplify our previous model-based methods for visual tracking of vehicles for use in a real-time system intended to provide continuous monitoring and classification of traffic from a fixed camera on a busy multi-lane motorway. The main constraints of the system design were: (i) all low level processing is to be carried out by low-cost auxiliary hardware; (ii) all 3-D reasoning is to be carried out automatically off-line, at set-up time. The system developed uses three main stages: (i) pose and model hypothesis using 1-D templates, (ii) hypothesis tracking, and (iii) hypothesis verification, using 2-D templates. Stages (i) and (iii) have radically different computing performance and computational costs, and need to be carefully balanced for efficiency. Together, they provide an effective way to locate, track and classify vehicles.

**An Image Change Detection Algorithm Based on Markov Random Field Models**

This paper addresses the problem of image change detection (ICD) based on Markov random field (MRF) models. MRF has long been recognized as an accurate model to describe a variety of image characteristics. Here, we use the MRF to model both noiseless images obtained from the actual scene and change images (CIs), the sites of which indicate changes between a pair of observed images. The optimum ICD algorithm under the maximum a posteriori (MAP) criterion is developed under this model. Examples are presented for illustration and performance evaluation.

**Automatic Traffic Surveillance System for Vehicle Tracking and Classification**

This paper presents an automatic traffic surveillance system to estimate important traffic parameters from video sequences using only one camera. Different from traditional methods that can classify vehicles to only cars and noncars, the proposed method has a good ability to categorize vehicles into more specific classes by introducing a new "linearity" feature in vehicle representation. In addition, the proposed system can well tackle the problem of vehicle occlusions caused by shadows, which often lead to the failure of further vehicle counting and classification. This problem is solved by a novel line-based shadow algorithm that uses a set of lines to eliminate all unwanted shadows. The used lines are devised from the information of lane-dividing lines. Therefore, an automatic scheme to detect lane-dividing lines is also proposed. The found lane-dividing lines can also provide important information for feature normalization, which can make the vehicle size more invariant, and thus much enhance the accuracy of vehicle classification. Once all features are extracted, an optimal classifier is then designed to robustly categorize vehicles into different classes. When recognizing a vehicle, the designed classifier can collect different evidences from its trajectories and the database to make an optimal decision for vehicle classification. Since more evidences are used, more robustness of classification can be achieved. Experimental results show that the proposed method is more robust, accurate, and powerful than other traditional methods, which utilize only the vehicle size and a single frame for vehicle classification.

**3. SYSTEM ANALYSIS**

**3.1 Existing System:**

In recent times, there has been a drastic change in people’s lifestyles and with an increase in incomes and lower cost of automobiles there is a huge increment in the number of cars on the roads which has led to traffic and commotion. The manual efforts to keep people from breaking traffic rules such as the speed limit are not enough. There is not enough police and man force available to track the traffic and vehicles on roads and check them for speed control. Hence, we require technologically advanced speed calculators installed that effectively detect cars on the road and calculate their speeds.

**Disadvantage:**

1.Less Accuracy.

**3.2. Proposed System:**

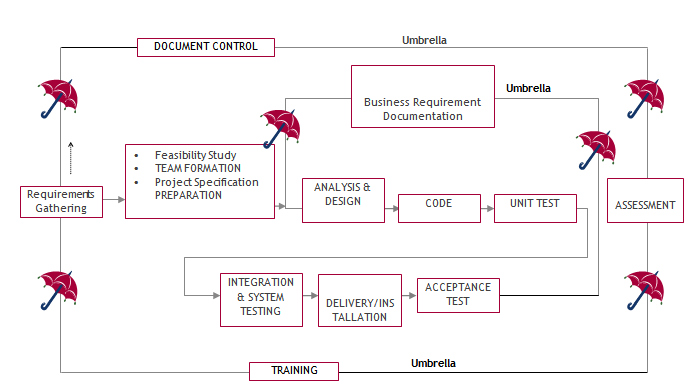
we can use OpenCV software which uses the Haar cascade to train our machine To implement the above idea two basic requirements, need to be met which are the effective detection of the cars on roads and their velocity measurement. For this purpose to detect the object, in this case the car. we have developed a Haar cascade to detect cars on the roads, whose velocities are then measured using a python script. The real-time application of this project proves to be much useful as it is easy to implement, fast to process and efficient with low cost development. Also, the tool might be useful to apply in simulation tools to measure velocities of cars. This can be further developed to identify all kinds of vehicles as well as to check anyone who breaks a traffic light.

**Advantage:**

1.More Accuracy.

**3.3. PROCESS MODEL USED WITH JUSTIFICATION**

**SDLC (Umbrella Model):**

****

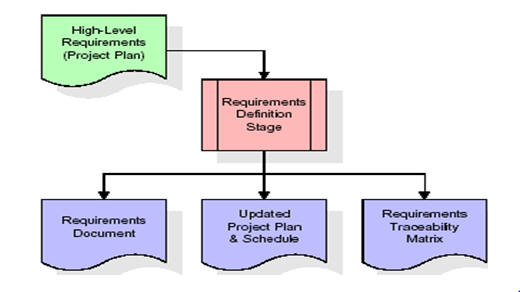
The requirements gathering process takes as its input SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

**Stages in SDLC:**

* Requirement Gathering
* Analysis
* Designing
* Coding
* Testing
* Maintenance

**Requirements Gathering** **stage:**

the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as mission critical inputs, outputs and reports. A user class hierarchy is developed and associated with these major functions, data areas, and data entities. Each of these definitions is termed a Requirement. Requirements are identified by unique requirement identifiers and, at minimum, contain a requirement title and textual description.



These requirements are fully described in the primary deliverables for this stage: the Requirements Document and the Requirements Traceability Matrix (RTM). The requirements document contains complete descriptions of each requirement, including diagrams and references to external documents as necessary. Note that detailed listings of database tables and fields are *not* included in the requirements document.

The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan. The purpose of the RTM is to show that the product components developed during each stage of the software development lifecycle are formally connected to the components developed in prior stages.

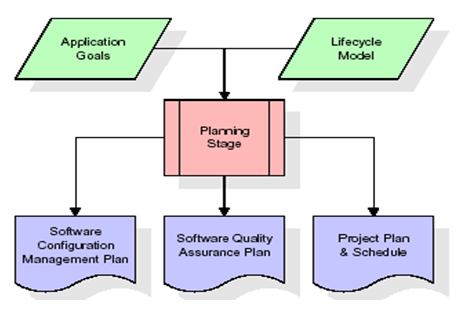
In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term requirements traceability.

The outputs of the requirements definition stage include the requirements document, the RTM, and an updated project plan.

* Feasibility study is all about identification of problems in a project.
* No. of staff required to handle a project is represented as Team Formation, in this case only modules are individual tasks will be assigned to employees who are working for that project.
* Project Specifications are all about representing of various possible inputs submitting to the server and corresponding outputs along with reports maintained by administrator.

**Analysis Stage:**

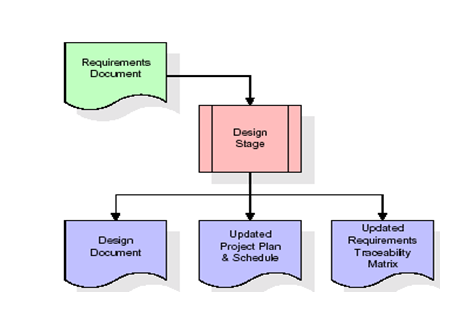
The planning stage establishes a bird's eye view of the intended software product, and uses this to establish the basic project structure, evaluate feasibility and risks associated with the project, and describe appropriate management and technical approaches.



The most critical section of the project plan is a listing of high-level product requirements, also referred to as goals. All of the software product requirements to be developed during the requirements definition stage flow from one or more of these goals. The minimum information for each goal consists of a title and textual description, although additional information and references to external documents may be included. The outputs of the project planning stage are the configuration management plan, the quality assurance plan, and the project plan and schedule, with a detailed listing of scheduled activities for the upcoming Requirements stage, and high level estimates of effort for the out stages.

**Designing Stage:**

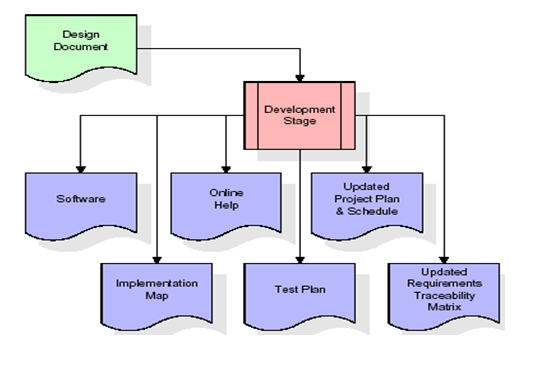
The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.



When the design document is finalized and accepted, the RTM is updated to show that each design element is formally associated with a specific requirement. The outputs of the design stage are the design document, an updated RTM, and an updated project plan.

**Development (Coding) Stage:**

The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artifacts will be produced. Software artifacts include but are not limited to menus, dialogs, and data management forms, data reporting formats, and specialized procedures and functions. Appropriate test cases will be developed for each set of functionally related software artifacts, and an online help system will be developed to guide users in their interactions with the software.



The RTM will be updated to show that each developed artifact is linked to a specific design element, and that each developed artifact has one or more corresponding test case items. At this point, the RTM is in its final configuration. The outputs of the development stage include a fully functional set of software that satisfies the requirements and design elements previously documented, an online help system that describes the operation of the software, an implementation map that identifies the primary code entry points for all major system functions, a test plan that describes the test cases to be used to validate the correctness and completeness of the software, an updated RTM, and an updated project plan.

**Integration & Test Stage:**

During the integration and test stage, the software artifacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability. During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into the Production Initiation Plan.



The outputs of the integration and test stage include an integrated set of software, an online help system, an implementation map, a production initiation plan that describes reference data and production users, an acceptance plan which contains the final suite of test cases, and an updated project plan.

* **Installation & Acceptance Test:**

During the installation and acceptance stage, the software artifacts, online help, and initial production data are loaded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer.

After customer personnel have verified that the initial production data load is correct and the test suite has been executed with satisfactory results, the customer formally accepts the delivery of the software.



The primary outputs of the installation and acceptance stage include a production application, a completed acceptance test suite, and a memorandum of customer acceptance of the software. Finally, the PDR enters the last of the actual labor data into the project schedule and locks the project as a permanent project record. At this point the PDR "locks" the project by archiving all software items, the implementation map, the source code, and the documentation for future reference.

**Maintenance:**

Outer rectangle represents maintenance of a project, Maintenance team will start with requirement study, understanding of documentation later employees will be assigned work and they will undergo training on that particular assigned category. For this life cycle there is no end, it will be continued so on like an umbrella (no ending point to umbrella sticks).

**3.4. Software Requirement Specification**

**3.4.1. Overall Description**

A Software Requirements Specification (SRS) – a [requirements specification](http://en.wikipedia.org/wiki/Requirements_specification) for a [software system](http://en.wikipedia.org/wiki/Software_system) is a complete description of the behavior of a system to be developed. It includes a set of [use cases](http://en.wikipedia.org/wiki/Use_case) that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. [Nonfunctional requirements](http://en.wikipedia.org/wiki/Non-functional_requirements) are requirements which impose constraints on the design or implementation (such as [performance engineering](http://en.wikipedia.org/wiki/Performance_engineering) requirements, [quality](http://en.wikipedia.org/wiki/Quality_%28business%29) standards, or design constraints).

System requirements specification: A structured collection of information that embodies the requirements of a system. A [business analyst](http://en.wikipedia.org/wiki/Business_analyst), sometimes titled [system analyst](http://en.wikipedia.org/wiki/System_analyst), is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Within the [systems development lifecycle](http://en.wikipedia.org/wiki/Systems_development_life_cycle) domain, the BA typically performs a liaison function between the business side of an enterprise and the information technology department or external service providers. Projects are subject to three sorts of requirements:

* [Business requirements](http://en.wikipedia.org/wiki/Business_requirements) describe in business terms *what* must be delivered or accomplished to provide value.
* Product requirements describe properties of a system or product (which could be one of several ways to accomplish a set of business requirements.)
* Process requirements describe activities performed by the developing organization. For instance, process requirements could specify .Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:
* **ECONOMIC FEASIBILITY**

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs. The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economical feasibility for certain.

* **Operational Feasibility**

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization’s operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits. The well-planned design would

ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

* **TECHNICAL FEASIBILITY**

Earlier no system existed to cater to the needs of ‘Secure Infrastructure Implementation System’. The current system developed is technically feasible. It is a web based user interface for audit workflow at NIC-CSD. Thus it provides an easy access to .the users. The database’s purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security.

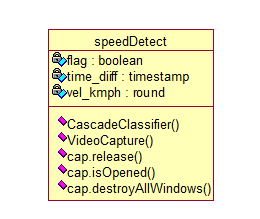
**4. SYSTEM DESIGN**

**UML Diagram:**

**Class Diagram:**

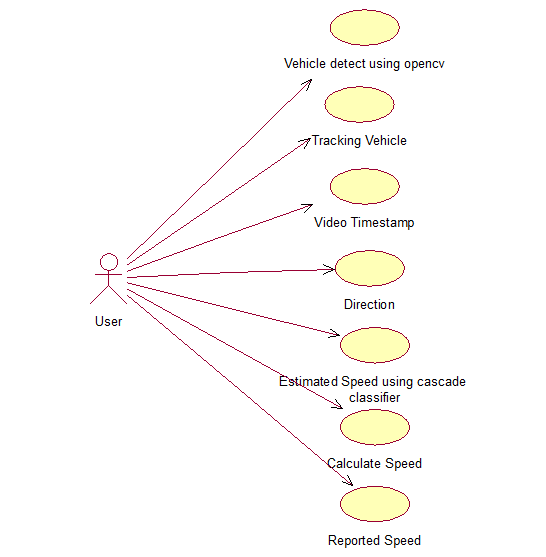
The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed. In the diagram, classes are represented with boxes which contain three parts:

* The upper part holds the name of the class
* The middle part contains the attributes of the class
* The bottom part gives the methods or operations the class can take or undertake.



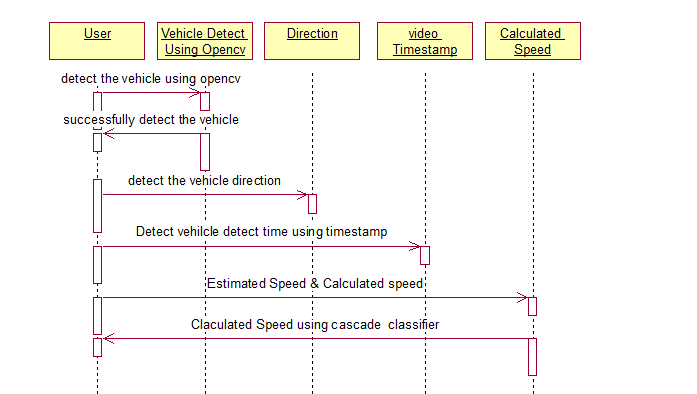
**Use case Diagram:**

A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.



**Sequence Diagram:**

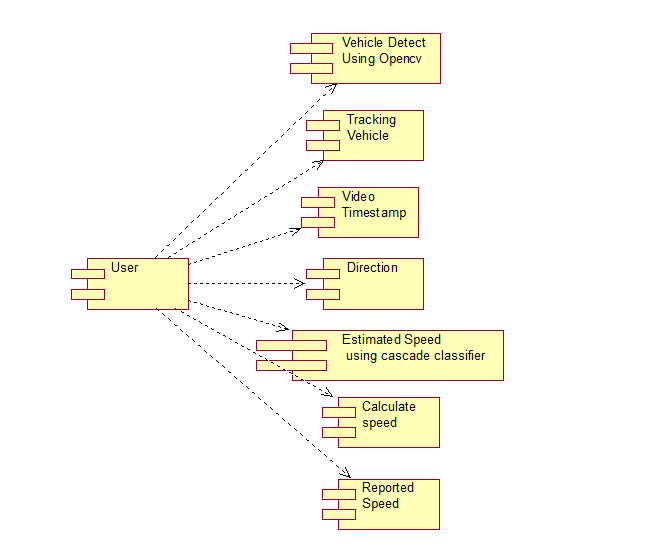
A sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



**Component Diagram:**

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems.

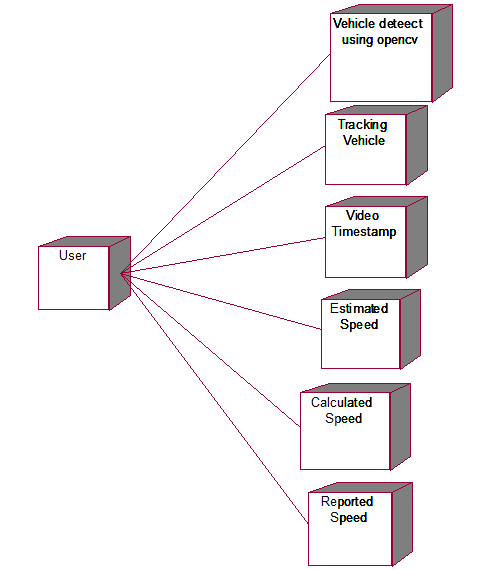
Components are wired together by using an assembly connector to connect the required interface of one component with the provided interface of another component. This illustrates the service consumer - service provider relationship between the two components.



**Deployment Diagram:**

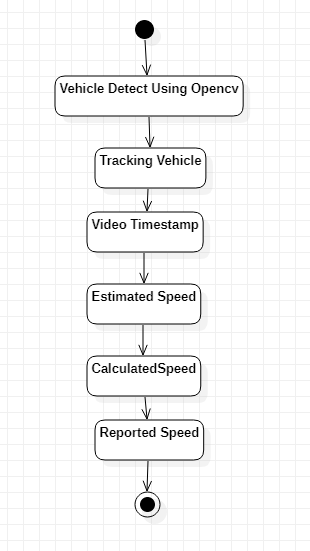
A deployment diagram in the Unified Modeling Language models the *physical* deployment of artifacts on nodes. To describe a web site, for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server), what software components ("artifacts") run on each node (e.g., web application, database), and how the different pieces are connected (e.g. JDBC, REST, RMI).

The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have sub nodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers.



**Activity Diagram:**

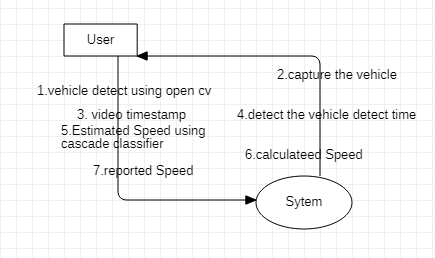
Activity diagram is another important diagram in UML to describe dynamic aspects of the system. It is basically a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent.



**Data Flow Diagram:**

Data flow diagrams illustrate how data is processed by a system in terms of inputs and outputs. Data flow diagrams can be used to provide a clear representation of any business function. The technique starts with an overall picture of the business and continues by analyzing each of the functional areas of interest. This analysis can be carried out in precisely the level of detail required. The technique exploits a method called top-down expansion to conduct the analysis in a targeted way.

As the name suggests, Data Flow Diagram (DFD) is an illustration that explicates the passage of information in a process. A DFD can be easily drawn using simple symbols. Additionally, complicated processes can be easily automated by creating DFDs using easy-to-use, free downloadable diagramming tools. A DFD is a model for constructing and analyzing information processes. DFD illustrates the flow of information in a process depending upon the inputs and outputs. A DFD can also be referred to as a Process Model. A DFD demonstrates business or technical process with the support of the outside data saved, plus the data flowing from the process to another and the end results.



**5. IMPLEMETATION:**

**5.1 Python**

Python is a general-purpose language. It has wide range of applications from Web development (like: Django and Bottle), scientific and mathematical computing (Orange, SymPy, NumPy) to desktop graphical user Interfaces (Pygame, Panda3D). The syntax of the language is clean and length of the code is relatively short. It's fun to work in Python because it allows you to think about the problem rather than focusing on the syntax.

**History of Python:**

Python is a fairly old language created by Guido Van Rossum. The design began in the late 1980s and was first released in February 1991.

**Why Python was created?**

In late 1980s, Guido Van Rossum was working on the Amoeba distributed operating system group. He wanted to use an interpreted language like ABC (ABC has simple easy-to-understand syntax) that could access the Amoeba system calls. So, he decided to create a language that was extensible. This led to design of a new language which was later named Python.

**Why the name Python?**

No. It wasn't named after a dangerous snake. Rossum was fan of a comedy series from late seventies. The name "Python" was adopted from the same series "Monty Python's Flying Circus".

**Features of Python:**

**A simple language which is easier to learn**

Python has a very simple and elegant syntax. It's much easier to read and write Python programs compared to other languages like: C++, Java, C#. Python makes programming fun and allows you to focus on the solution rather than syntax.

If you are a newbie, it's a great choice to start your journey with Python.

**Free and open-source**

You can freely use and distribute Python, even for commercial use. Not only can you use and distribute softwares written in it, you can even make changes to the Python's source code.

Python has a large community constantly improving it in each iteration.

**Portability**

You can move Python programs from one platform to another, and run it without any changes.

It runs seamlessly on almost all platforms including Windows, Mac OS X and Linux.

**Extensible and Embeddable**

Suppose an application requires high performance. You can easily combine pieces of C/C++ or other languages with Python code.

This will give your application high performance as well as scripting capabilities which other languages may not provide out of the box.

**A high-level, interpreted language**

Unlike C/C++, you don't have to worry about daunting tasks like memory management, garbage collection and so on.

Likewise, when you run Python code, it automatically converts your code to the language your computer understands. You don't need to worry about any lower-level operations.

**Large standard libraries to solve common tasks**

Python has a number of standard libraries which makes life of a programmer much easier since you don't have to write all the code yourself. For example: Need to connect MySQL database on a Web server? You can use MySQLdb library using import MySQLdb .

Standard libraries in Python are well tested and used by hundreds of people. So you can be sure that it won't break your application.

**Object-oriented**

Everything in Python is an object. Object oriented programming (OOP) helps you solve a complex problem intuitively.

With OOP, you are able to divide these complex problems into smaller sets by creating objects.

**Applications of Python:**

**1. Simple Elegant Syntax**

Programming in Python is fun. It's easier to understand and write Python code. Why? The syntax feels natural. Take this source code for an example:

a = 2

b = 3

sum = a + b

print(sum)

**2. Not overly strict**

You don't need to define the type of a variable in Python. Also, it's not necessary to add semicolon at the end of the statement.

Python enforces you to follow good practices (like proper indentation). These small things can make learning much easier for beginners.

**3. Expressiveness of the language**

Python allows you to write programs having greater functionality with fewer lines of code. Here's a link to the source code of Tic-tac-toe game with a graphical interface and a smart computer opponent in less than 500 lines of code. This is just an example. You will be amazed how much you can do with Python once you learn the basics.

**4. Great Community and Support**

Python has a large supporting community. There are numerous active forums online which can be handy if you are stuck.

**5.2 Sample Code:**

**speedDetect.py**

import numpy as np

import cv2

import time

car\_cascade = cv2.CascadeClassifier('hand.xml')

cap = cv2.VideoCapture('car.mp4')

wide=0.1 #depends upon size of car(~2.5)

flag=True

start=end=0

time\_diff=0

while(cap.isOpened()):

ret, img = cap.read()

height,width,chan=img.shape

#print(height,width,chan)

gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

cars = car\_cascade.detectMultiScale(gray, 1.3, 5)

#crp=gray[0:480,0:int(width/2)+10]

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

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

center\_x=(2\*x+w)/2

center\_y=(2\*y+h)/2

#print(center\_x,center\_y)

dist1=((wide\*668.748634441)/w)

#print("Distance from car:",round(dist1,2),"m")

roi\_gray = gray[y:y+h,x:x+w]

roi\_color = img[y:y+h,x:x+w]

dist0=((wide\*668.748634441)/w)

actual\_dist=dist0\*(width/2)/668.748634441

#print("Actual Distance:",actual\_dist)

if flag is True and int(round(center\_x)) in (range(0,80) or range(400,480)):

start=time.time()

flag=False

#print("Start:",start)

if flag is False and int(round(center\_x)) in range(int(round(width/2))-10,int(round(width/2))+10):

end=time.time()

time\_diff=end-start

#print("End:",end)

flag=True

s\_flag=True

#print("Time Difference:",time\_diff)

if time\_diff>0 and s\_flag==True:

velocity=actual\_dist/time\_diff

#print(round(start),round(end))

vel\_kmph=round(velocity\*3.6,2)

print("Speed:",vel\_kmph,"kmph")

print("Distance from car:",round(dist1,2),"m")

s\_flag=False

cv2.line(img,(int(width/2),0),(int(width/2),height),(255,0,0),2)

cv2.imshow('frame',img)

if cv2.waitKey(1) & 0xFF == ord('q'):

break

cap.release()

cv2.destroyAllWindows()

**6.TESTING:**

**Implementation and Testing:**

Implementation is one of the most important tasks in project is the phase in which one has to be cautions because all the efforts undertaken during the project will be very interactive. Implementation is the most crucial stage in achieving successful system and giving the users confidence that the new system is workable and effective. Each program is tested individually at the time of development using the sample data and has verified that these programs link together in the way specified in the program specification. The computer system and its environment are tested to the satisfaction of the user.

**Implementation:**

The implementation phase is less creative than system design. It is primarily concerned with user training, and file conversion. The system may be requiring extensive user training. The initial parameters of the system should be modifies as a result of a programming. A simple operating procedure is provided so that the user can understand the different functions clearly and quickly. The different reports can be obtained either on the inkjet or dot matrix printer, which is available at the disposal of the user. The proposed system is very easy to implement. In general implementation is used to mean the process of converting a new or revised system design into an operational one.

## Testing

Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system property functions as a unit. The test data should be chosen such that it passed through all possible condition. Actually testing is the state of implementation which aimed at ensuring that the system works accurately and efficiently before the actual operation commence. The following is the description of the testing strategies, which were carried out during the testing period.

**System Testing**

Testing has become an System integral part of any system or project especially in the field of information technology. The importance of testing is a method of justifying, if one is ready to move further, be it to be check if one is capable to with stand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to user the software must be tested whether it is solving the purpose for which it is developed. This testing involves various types through which one can ensure the software is reliable. The program was tested logically and pattern of execution of the program for a set of data are repeated. Thus the code was exhaustively checked for all possible correct data and the outcomes were also checked.

**Module Testing**

To locate errors, each module is tested individually. This enables us to detect error and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus all the modules are individually tested from bottom up starting with the smallest and lowest modules and proceeding to the next level. Each module in the system is tested separately. For example the job classification module is tested separately. This module is tested with different job and its approximate execution time and the result of the test is compared with the results that are prepared manually. The comparison shows that the results proposed system works efficiently than the existing system. Each module in the system is tested separately. In this system the resource classification and job scheduling modules are tested separately and their corresponding results are obtained which reduces the process waiting time.

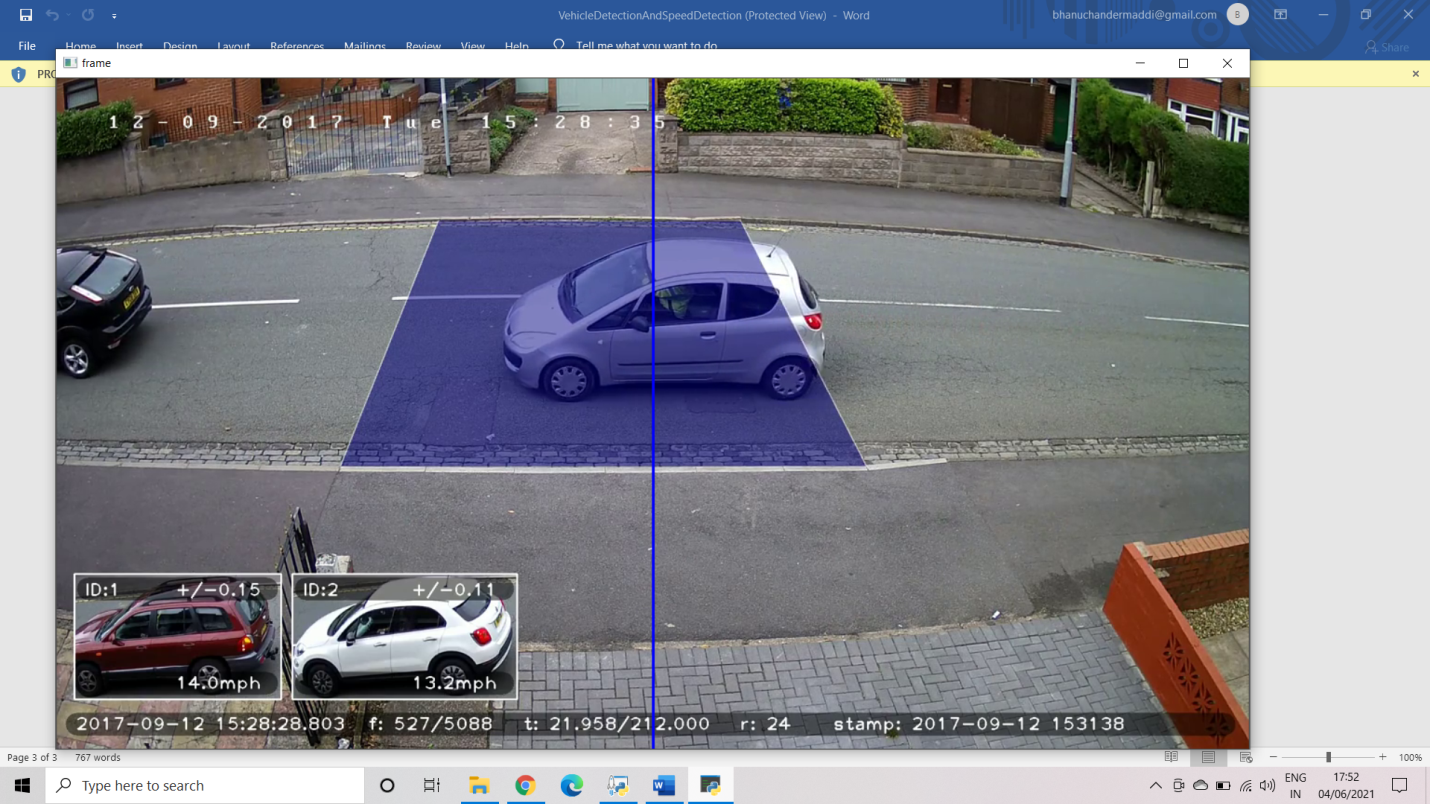
**Integration Testing**

After the module testing, the integration testing is applied. When linking the modules there may be chance for errors to occur, these errors are corrected by using this testing. In this system all modules are connected and tested. The testing results are very correct. Thus the mapping of jobs with resources is done correctly by the system.

**Acceptance Testing**

When that user fined no major problems with its accuracy the system passers through a final acceptance test. This test confirms that the system needs the original goals, objectives and requirements established during analysis without actual execution which elimination wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptable and ready for the operation.

## 7.SCREEN SHOTS:

****

**8.CONCLUSION:**

This paper proposes an approach to detect and track the moving vehicles and estimation of their speeds. The innovation of the approach lies in the selection of the Region of Interest for the vehicle detection. The approach proposed in this paper is verified and tested on four different videos. The average detection accuracy achieved by proposed approach is 87.7%. The proposed approach uses cropping operation to minimize the scope of any false positive detection on both sides of road. The average false positive detection in the proposed approach is lower than average false positive detection in leading approaches such as STA12. Maximum tracking accuracy achieved by the proposed technique is up to 98.3% in the afternoon session, but the average tracking accuracy of the proposed approach is about 92.2% that is improvement to other methods. In the proposed method, detection and tracking of the moving vehicles utilizes parameters such as position, height and width of vehicle instead of features extraction. This requires lesser computation and memory. The proposed approach stores vehicles parameters, estimated speed of the detected vehicles in the database. The proposed system can be adopted easily in existing traffic management system.

**9.REFERENCES:**

[1] T. Kumar, R. Sachan and D. S. Kushwaha, Smart City Traffic Management and Surveillance System for Indian Scenario, Proceedings of International Conference on Recent Advances in Mathematics, Statistics and Computer Science (ICRAMSCS), (2015).

[2] A. Mittal, A. Monnet and N. Paragios, Scene Modeling and Change Detection in Dynamic Scenes: A Subspace Approach, Computer Vision and Image Understanding, vol. 113(1), pp. 63–79, (2009).

[3] S. Gupte, O. Masoud, R. F. K. Martin and N. P. Papanikolopoulos, Detection and Classification of Vehicles, IEEE Transactions on Intelligent Transportation Systems, vol. 3(1), pp. 37–47, (2002).

[4] A. Hussain, K. Shahzad and C. Tang, Real Time Speed Estimation of Vehicles, WasetAcNz, vol. 6(1), pp. 726–730, (2012).

[5] O. Barnich and M. V. Droogenbroeck, ViBe: A Universal Background Subtraction Algorithm for Video Sequences, IEEE Transactions on Image Processing 2015, vol. 20, pp. 1709–1724, July (2011).

[6] N. Friedman and S. Russell, Image Segmentation in Video Sequences: A Probabilistic Approach, In UAI’97 Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence, pp. 175–181, (1997).

[7] S. Javed, U. D. L. Rochelle, S. K. Jung and L. Mia, OR-PCA with Dynamic Feature Selection for Robust Background Subtraction, Proceedings of the 30th Annual ACM Symposium on Applied Computing-SAC’15, pp. 86–91, (2015).

[8] G. Sullivan, K. Baker, A. Worrall, C. Attwood and P. Remagnino, Model-Based Vehicle Detection and Classification using Orthographic Approximations, Image and Vision Computing, vol. 15(8), pp. 649–654, (1997).

[9] T. Kasetkasem and P. K. Varshney, An Image Change Detection Algorithm Based on Markov Random Field Models, IEEE Transactions on Geoscience and Remote Sensing, vol. 40(8), pp. 263–1823, (2002). [10] J. W. Hsieh, S. H. Yu, Y. S. Chen and W. F. Hu, Automatic Traffic Surveillance System for Vehicle Tracking and Classification, IEEE Transactions on Intelligent Transportation Systems, vol. 7(2), pp. 175–187, (2006).

[11] K. Padmavathi and K. Thangadurai, Implementation of RGB and Grayscale Images in Plant Leaves Disease Detection–Comparative Study, Indian Journal of Science and Technology, vol. 9(6), (2016).

[12] Picomixer, http://picomixer.com/documents/Introduction of Picomixer STA.pdf/