Smart Control of Traffic Light Using Artificial Intelligence

1. INTRODUCTION

With the increasing number of vehicles in urban areas, many road networks are facing problems with the capacity drop of roads and the corresponding Level of Service. Many traffic-related issues occur because of traffic control systems on intersections that use fixed signal timers. They repeat the same phase sequence and its duration with no changes. Increased demand for road capacity also increases the need for new solutions for traffic control that can be found in the field of Intelligent Transport Systems. Let us take the case study of Mumbai and Bangalore. Traffic flow in Bangalore is the worst in the world while Mumbai is close behind in fourth position, according to a report detailing the traffic situation in 416 cities across 57 countries. In Bangalore, a journey during rush-hour takes 71% longer. In Mumbai, it is 65% longer [1]. There are three standard methods for traffic control that are being used currently: 1) Manual Controlling: As the name suggests, it requires manpower to control the traffic. The traffic police are allotted for a required area to control traffic. The traffic police carry signboard, sign light, and whistle to control the traffic. 2) Conventional traffic lights with static timers: These are controlled by fixed timers. A constant numerical value is loaded in the timer. The lights are automatically switching to red and green based on the timer value. 3) Electronic Sensors: Another advanced method is placing some loop detectors or proximity sensors on the road. This sensor gives data about the traffic on the road. According to the sensor data, the traffic signals are controlled. These conventional methods face certain drawbacks. The manual controlling system requires a large amount of manpower. As there is poor strength of traffic police, we cannot have them controlling traffic manually in all areas of a city or town. So a better system to control the traffic is needed. Static traffic controlling uses a traffic light with a timer for every phase, which is fixed and does not adapt according to the real-time traffic on that road. While using electronic sensors i.e., proximity sensors or loop detectors, the accuracy and coverage are often in conflict because the collection of high-quality information is usually based on sophisticated and expensive technologies, and thus limited budget will reduce the number of facilities. Moreover, due to the limited effective range of most sensors, the total coverage on a network of facilities usually requires a lot of sensors. In recent years, video monitoring and surveillance systems have been extensively used in traffic management for security, ramp metering, and providing information and updates to travellers in real-time. The traffic density estimation and vehicle classification can also be achieved using video monitoring systems, which can then be used to control the timers of the traffic signals so as to optimize traffic flow and minimize congestion. Our proposed system aims to design a traffic light controller based on Computer Vision that can adapt to the current traffic situation. It uses live images from the CCTV cameras at traffic junctions for real-time traffic density calculation by detecting the number of vehicles at the signal and setting the green signal time accordingly. The vehicles are classified as a car, bike, bus/truck, or rickshaw to obtain an accurate estimate of the green signal time. It uses YOLO in order to detect the number of vehicles and then set the timer of the traffic signal according to vehicle density in the corresponding direction. This helps to optimize the green signal times, and traffic is cleared at a much faster rate than a static system, thus reducing the unwanted delays, congestion, and waiting time, which in turn will reduce the fuel consumption and pollution.

1.1 Objective of the Project

Traffic congestion is becoming one of the critical issues with increasing population and automobiles in cities. Traffic jams not only cause extra delay and stress for the drivers, but also increase fuel consumption and air pollution. Although it seems to pervade everywhere, megacities are the ones most affected by it. And its ever-increasing nature makes it necessary to calculate the road traffic density in real-time for better signal control and effective traffic management. The traffic controller is one of the critical factors affecting traffic flow. Therefore, the need for optimizing traffic control to better accommodate this increasing demand arises. Our proposed system aims to utilize live images from the cameras at traffic junctions for traffic density calculation using image processing and AI. It also focuses on the algorithm for switching the traffic lights based on the vehicle density to reduce congestion, thereby providing faster transit to people and reducing pollution.

2. LITERATURE SURVEY

“Smart Control of Traffic Light System using Image Processing”

The congestion of the urban traffic is becoming one of critical issues with increasing population and automobiles in cities. Traffic jams not only cause extra delay and stress for the drivers, but also increase fuel consumption, add transportation cost, and increase carbon dioxide air pollution. The traffic controller is one of critical factors affecting the traffic flow. The conventional traffic patterns are nonlinear and complex and time dependent rather than traffic dependent. This paper proposes a traffic control system based on image processing using MATLAB code which changes the time of green, amber and red light with respect to the traffic density and traffic count. Two Arduino UNO is used, one for controlling green and amber lights and other for controlling red light. This is a continuous process.

"Improving Traffic Light Control by Means of Fuzzy Logic,"

In urban areas, the traffic demand grows every year due to the constantly increasing number of vehicles. The consequence is a capacity drop of the roads followed by traffic problems like congestion, reduced travel time, increased fuel consumption, etc. This paper presents an adaptive traffic light controller based on fuzzy logic for improving the traffic flow on an isolated intersection. A set of fuzzy rules has been made that, using the collected information from road detectors (queue length, arrival flow, and exit flow), computes the amount of time for which the next phase should be shortened or extended. The proposed fuzzy control system is constituted of two parts: one for the primary driveway (with a higher volume of vehicles) and for the secondary driveway (with a lower volume of vehicles). The proposed controller is compared with a fixed signal program in three scenarios with different traffic demand proofing the effectiveness of the developed decision rules.

"Smart controlling for traffic light time,"

Traffic jam and traffic accidents become serious problems especially in crowded cities, which wasting time and money. Traffic light is basic element in control traffic flow through specify waiting and going time, fixed traffic light time systems is bad control way, since number of cars is not consistency with each traffic light, thus lead to imbalance system. Intelligent transportation system including smart way to control traffic light time based on number of cars in each traffic light, this paper develops an automatic algorithm to control traffic light time based on artificial intelligent techniques and image for cars on traffic lights, this algorithm is validated by compare its results with manual results. Applying following proposed algorithm in transportation system will regulate traffic flow and reduce traveling and waiting time wasted in roads.

"Traffic Light Control and Violation Detection Using Image Processing”

Now a days as the population increases the transportation demands are increased. The total number of vehicles required to fulfil the transportation needs too are increased just as population have increased. Increase of the vehicle usage leads to heavy traffic in the road. It happens due to the current traffic control method and the road infrastructure. It will affect the human as well as fuel resources by wasting time in the road because of the heavy traffic. In order to reduce the wastage of time as well as the wastage of fuel in the road traffic an effective and a smart traffic control strategy is required. The traditional way of traffic control provides a time slot for each direction of road. Another advanced method is placing some proximity sensors on the road. This sensor gives the data about the traffic on the road. According to the sensor data the traffic signals are controlled. This project proposes a new way of traffic control. A digital camera installed with traffic signal light used to capture the live road images. Then the captured images are fed into digital image processor to find the traffic density on the road then the traffic signals are controlled. The proposed system helps to use the time and fuel resources efficiently by avoiding the time wasted on the empty road.

"Smart traffic lights switching and traffic density calculation using video processing,"

Congestion in traffic is a serious problem nowadays. Although it seems to pervade everywhere, mega cities are the ones most affected by it. And it's ever increasing nature makes it imperative to know the road traffic density in real time for better signal control and effective traffic management. There can be different causes of congestion in traffic like insufficient capacity, unrestrained demand, large Red Light delays etc. While insufficient capacity and unrestrained demand are somewhere interrelated, the delay of respective light is hard coded and not dependent on traffic. Therefore the need for simulating and optimizing traffic control to better accommodate this increasing demand arises. In recent years, video monitoring and surveillance systems have been widely used in traffic management for traveler's information, ramp metering and updates in real time. The traffic density estimation and vehicle classification can also be achieved using video monitoring systems. This paper presents the method to use live video feed from the cameras at traffic junctions for real time traffic density calculation using video and image processing. It also focuses on the algorithm for switching the traffic lights according to vehicle density on road, thereby aiming at reducing the traffic congestion on roads which will help lower the number of accidents. In turn it will provide safe transit to people and reduce fuel consumption and waiting time. It will also provide significant data which will help in future road planning and analysis. In further stages multiple traffic lights can be synchronized with each other with an aim of even less traffic congestion and free flow of traffic.

‘Overview of the YOLO Object Detection Algorithm’

Object detection has been known as the core of computer vision and attracted much research attention in recent years especially because of its close relationship with video analysis and image understanding. According to the abundant research on object detection, many traditional object detection methods have been proposed. This paper introduces some famous traditional methods, which are based on SIFT, HOG, SURF, and ORB. However, due to the characteristics of large amount of computation and simple training structure, the traditional detection method has low detection speed. With the fast rise of deeper learning, stronger devices are implemented to address the problems that exist in conventional architectures. In the architecture of the network, training and optimization functions etc., these models are special. In this paper, we review the frameworks for object detection based on deep learning. We begin our review with the methods based on Convolutional Neural Networks. Then typical methods of object detection and some helpful modification to improve detection performance are introduced. Moreover, the methods based on YOLO and SSD are introduced. In fact, despite of the same basis of algorithm or features, the performance and features of different methods are various. Thus in this paper we analyze the features and architecture of each method. This also offers some research to equate various approaches and draw some concrete conclusions. In this paper, we also introduce some typical datasets used for testing or training the object detection model. This paper made a systematic classification and summary in the object detection field, which can be meaningful and useful for the scholars who started to learn about it.

‘Traffic Signal Synchronization’

The delay, which decides the level of service and gauges the performance of a facility, is one of the most concerned aspects of transport planners and road users in the field of transportation. Because of the issue of rapid urbanization and vehicular growth, urban traffic is getting over congested which increases travel time, fuel utilization, environmental loss etc. It is a severe worry in developing nation like India. Signal Coordination is one of the most cost effective ways to overcome the delay proposed in this research. Signal Coordination improves traffic flow by arranging a green time of downstream traffic signal to coincide with the appearance of the platoon.

Different techniques for homogeneous traffic conditions and strict lane conditions have been developed, according to the literature. Very limited studies have been done on heterogeneous traffic conditions and lane change behavior, like in India. The main intension of signal synchronization is to focus on reducing delay experienced by the vehicles while traveling through the signalized intersection corridor and developed a co-ordinate signalized intersection network. The present paper deal with the various aspect of signal synchronization based on the past experience of various researches. It is also tried to identify the appropriate parameters along with the appropriate methodology correlated with Indian traffic scenario. Several important parameters have been found to be traffic composition, cycle length, traffic volume, phase sequence, offsets, saturation flow etc. It has been observed that sign at synchronization and they related aspect should be while designing the signal system in urban areas.

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

In India traffic is enlarge four times faster than population. Nowadays, so many countries suffer from the traffic congestion issues that affect the transportation method in cities and cause serious trouble. Even though replacing traffic officers and custodian by automatic traffic systems, the optimization of the heavy traffic jam is still a big issue to be faced, especially with several junction nodes. Traffic jams also build many other critical issues and problems which straightly affect the human routine lives and sometime reason for death for example if there is an emergency vehicle like ambulance on the roadway going with critical patient. In that situation if an ambulance gets stuck in a large traffic j am then there are high chances that the patient can't reach the hospital on time. It is very key to design an advanced traffic system which controls traffic intelligently to avoid accidents, collisions and traffic jams.

Disadvantage:

1. Less Accuracy

PROPOSED SYSTEM:

Our proposed system aims to present a traffic light controller based on Computer Vision that can adapt to the current traffic situation. It uses live video feed from the CCTV cameras at traffic junctions for real-time traffic density calculation by detecting the vehicles at the signal and setting the green signal time accordingly. The vehicles are classified as car, bus/truck, or rickshaw to obtain a more accurate estimate of the green signal time. We have used object detection like computer vision in order to detect the number of vehicles for each direction. We then set the timers of these traffic signals according to vehicle density in each direction and hence the system becomes adaptive. This helps to optimize the green signal times, and traffic is cleared at a far quicker rate than a static system, therefore reducing the unwanted delays, congestion, and waiting time, which in turn will reduce the fuel consumption and pollution

Advantage:

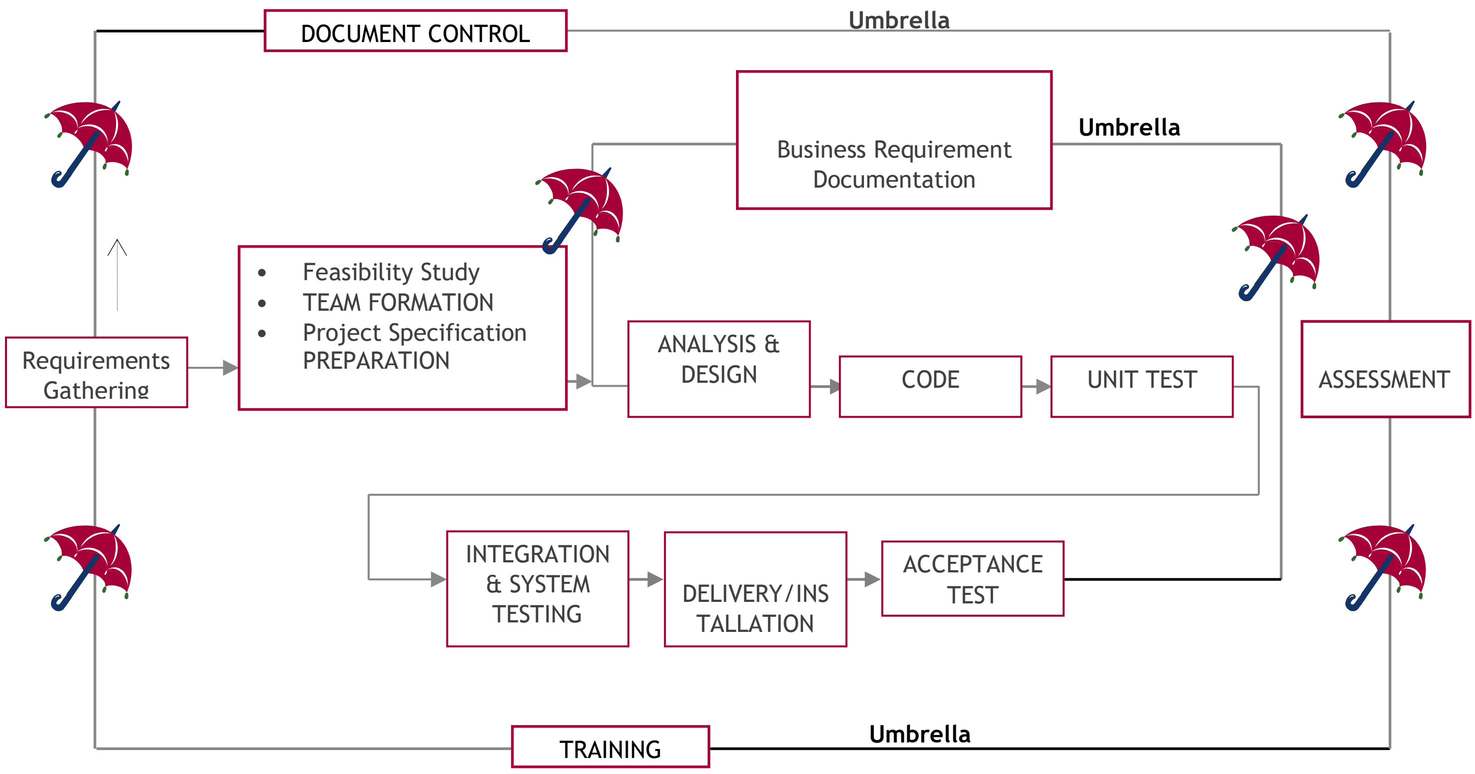
1. More Accuracy.

MODULES

- Run traffic simulation
- Run yolo traffic detection and tracking
- EXIT

3.3. PROCESS MODEL USED WITH JUSTIFICATION

SDLC (Umbrella Model):



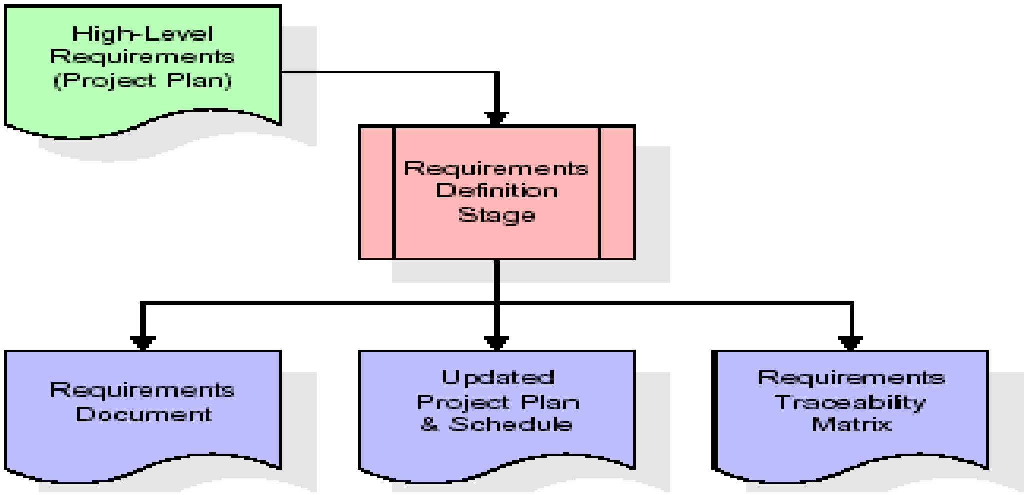
SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

<u>Stages in SDLC</u>:

- Requirement Gathering
- Analysis
- Designing
- Coding
- Testing
- Maintenance

<u>Requirements Gatheringstage</u>:

The requirements gathering process takes as its input 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.


High-Level Requirements (Project Plan)
Requirements Definition Stage
Requirements Document
Updated Project Plan & Schedule
Requirements Traceability Matrix

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.

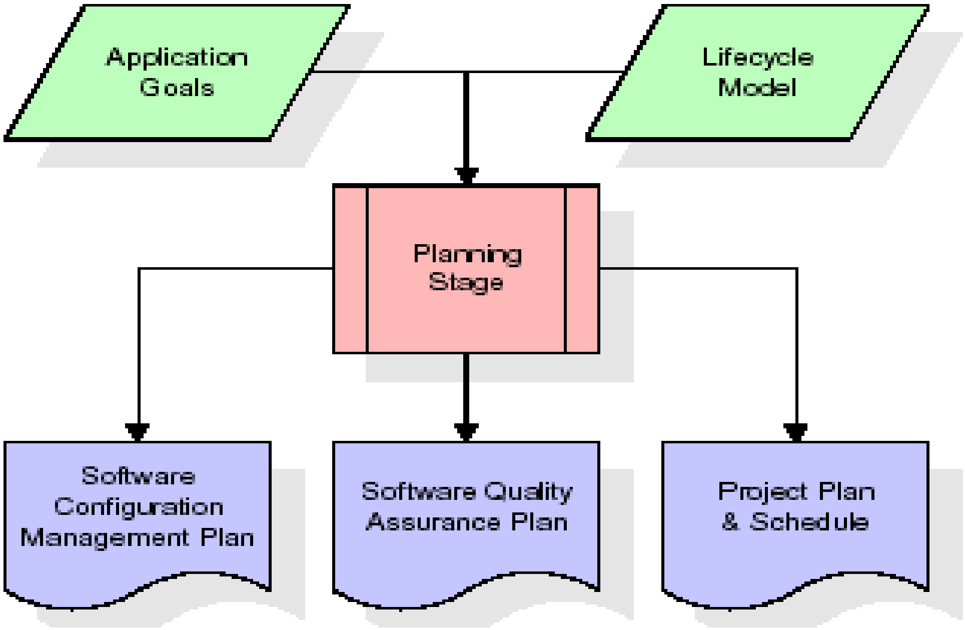
In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked 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:

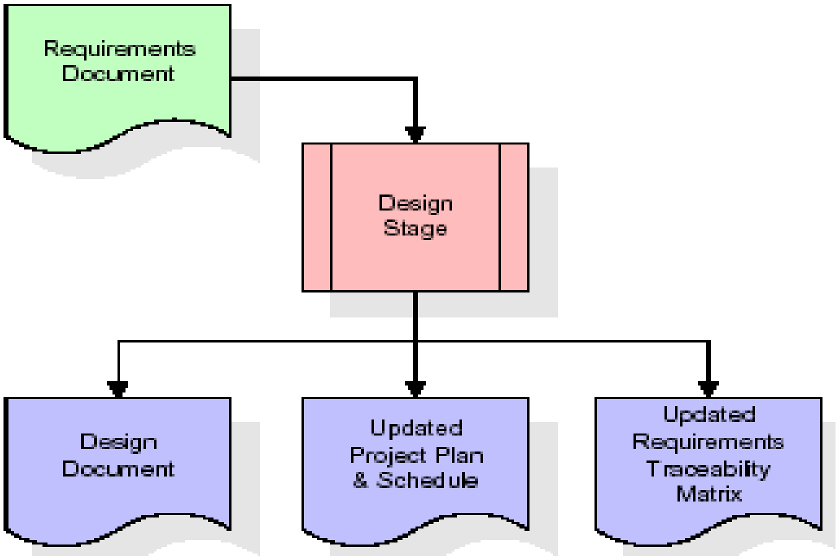
The planning stage establishes a bird's eye view of the intended software product, and uses this to establish the basic project structure, evaluate feasibility and risks associated with the project, and describe appropriate management and technical approaches.


Application Goals
Lifecycle Model
Planning Stage
Software Configuration Management Plan
Software Quality Assurance Plan
Project Plan & Schedule

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:

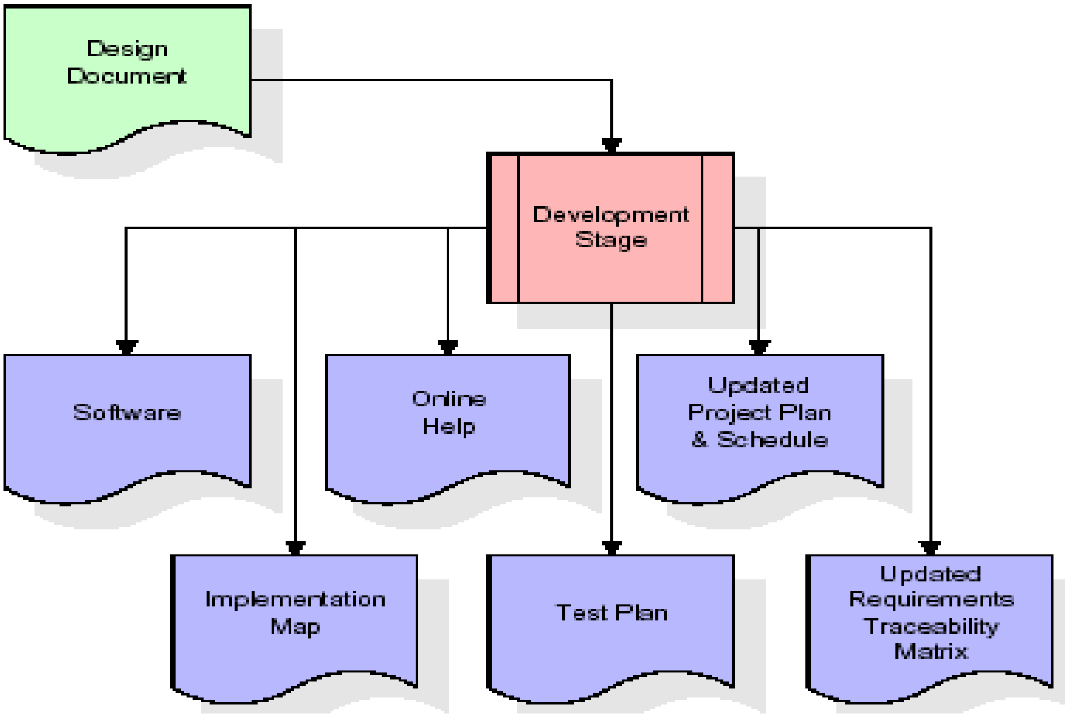
The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features 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.


Requirements Document
Design Stage
Design Document
Updated Project Plan & Schedule
Updated Requirements Traceability Matrix

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:

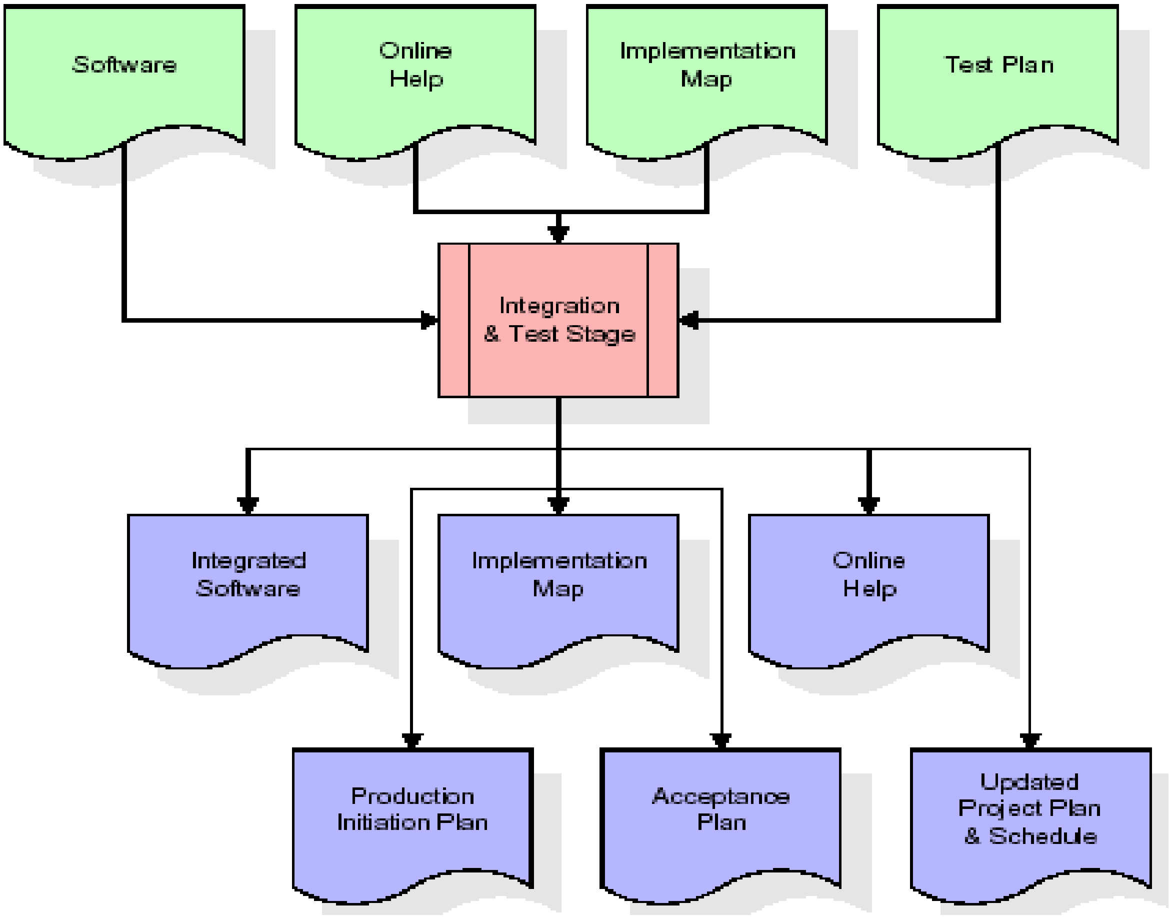
The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artefacts will be produced. Software artefacts include but are not limited to menus, dialogs, and data management forms, data reporting formats, and specialized procedures and functions. Appropriate test cases will be developed for each set of functionally related software artefacts, 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 artefact is linked to a specific design element, and that each developed artefact 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.

<u>Integration & Test Stage</u>:

During the integration and test stage, the software artefacts, 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.

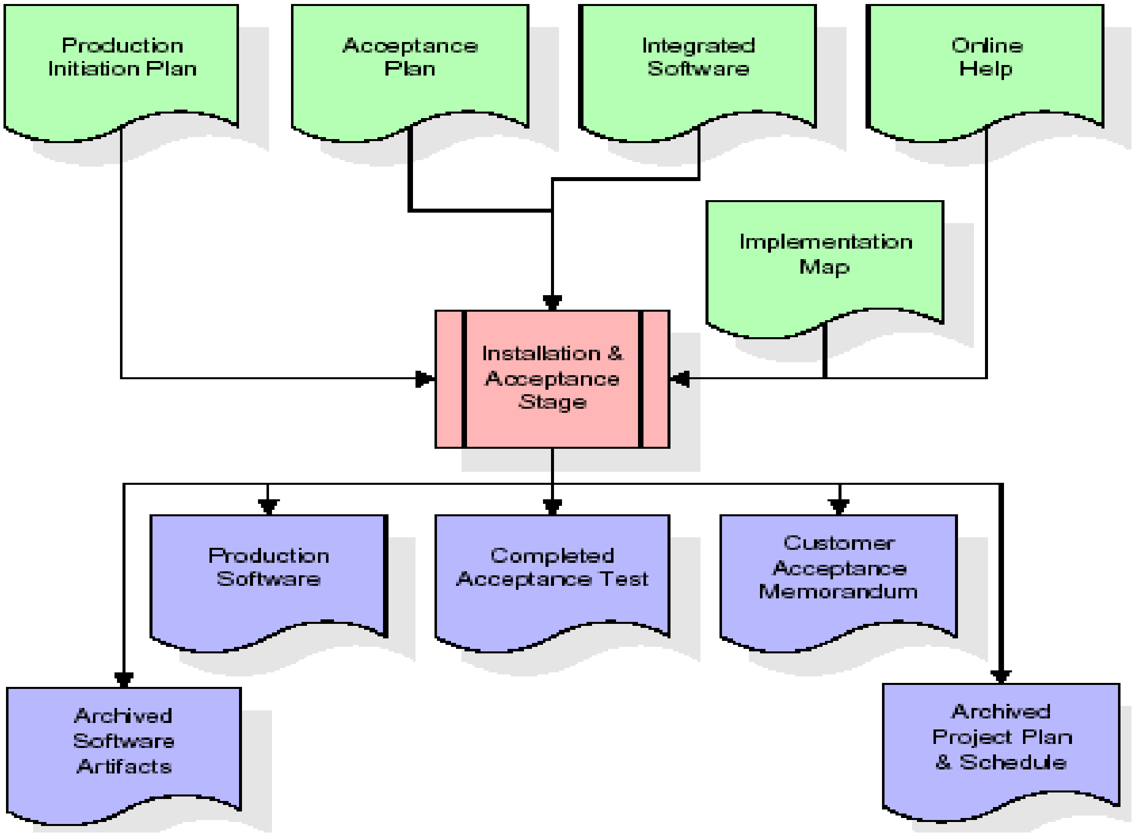


The outputs of the integration and test stage include an integrated set of software, an online help system, an implementation map, a production initiation plan that describes reference data and production users, an acceptance plan which contains the final suite of test cases, and an updated project plan.

- **Installation & Acceptance Test:**

During the installation and acceptance stage, the software artefacts, 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.


Production Initiation Plan
Acceptance Plan
Integrated Software
Online Help
Implementation Map
Installation & Acceptance Stage
Production Software
Completed Acceptance Test
Customer Acceptance Memorandum
Archived Software Artifacts
Archived Project Plan & Schedule

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 for a software system is a complete description of the behaviour of a system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. Non-functional requirements are requirements which impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

System requirements specification: A structured collection of information that embodies the requirements of a system. A business analyst, sometimes titled 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 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 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.

3.4.2. External Interface Requirements

User Interface

The user interface of this system is a user friendly python Graphical User Interface.

Hardware Interfaces

The interaction between the user and the console is achieved through python capabilities.

Software Interfaces

The required software is python.

HARDWARE REQUIREMENTS:

• Processor	-	Pentium –IV
• Speed	-	1.1 GHz
• RAM	-	4GB(min)
• Hard Disk	-	500GB
• Key Board	-	Standard Windows Keyboard
• Mouse	-	Two or Three Button Mouse
• Monitor	-	SVGA

SOFTWARE REQUIREMENTS:

• Operating System	-	Windows 10/above
• Programming Language	-	Python 3.7 /above

4. SYSTEM DESIGN

UML Diagram:

The Unified Modelling Language allows the software engineer to express an analysis model using the modelling notation that is governed by a set of syntactic semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

- **User Model View**

i. This view represents the system from the users perspective.

ii. The analysis representation describes a usage scenario from the end-users perspective.

- **Structural Model view**

  i. In this model the data and functionality are arrived from inside the system.

  ii. This model view models the static structures.

- **Behavioural Model View**

  It represents the dynamic of behavioural as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

- **Implementation Model View**

  In this the structural and behavioural as parts of the system are represented as they are to be built.

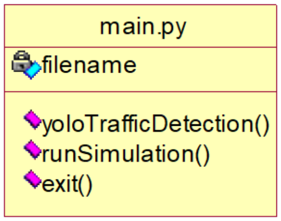
- **Environmental Model View**

  In this the structural and behavioural aspects of the environment in which the system is to be implemented are represented.

Class Diagram:

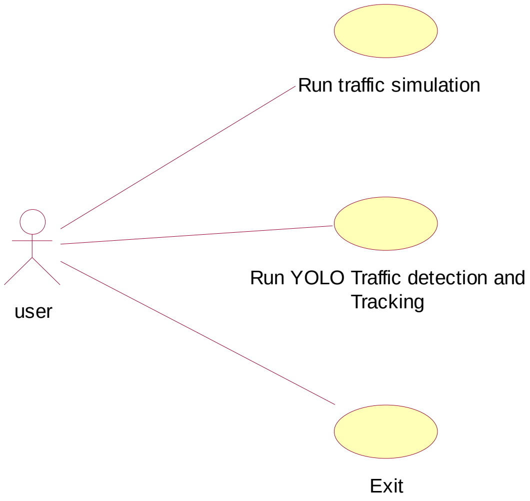
The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class 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



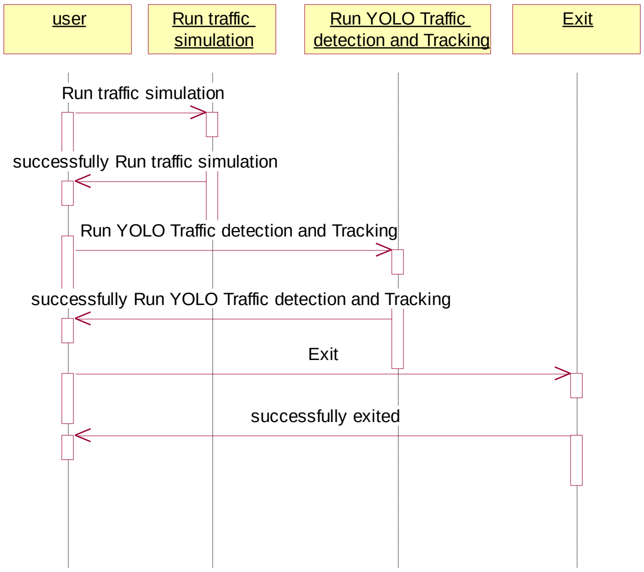
Use case Diagram:

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.



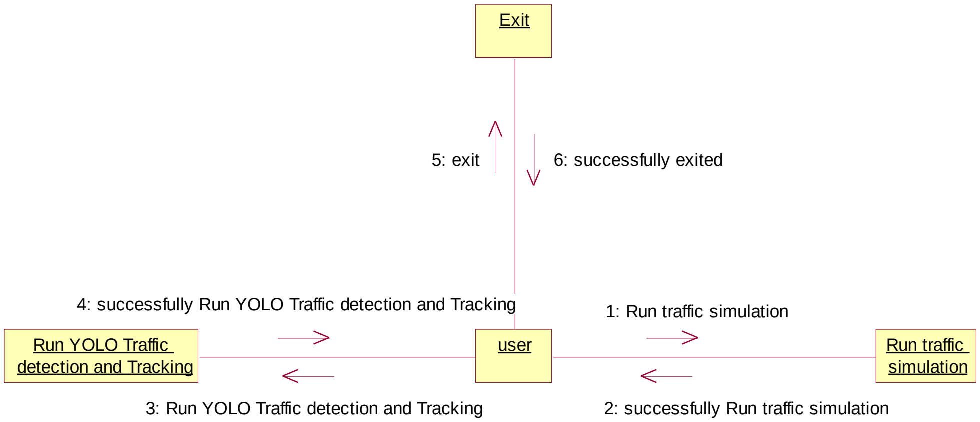
Sequence diagram:

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.


user
Run traffic simulation
Run YOLO Traffic detection and Tracking
Exit
Run traffic simulation
successfully Run traffic simulation
Run YOLO Traffic detection and Tracking
successfully Run YOLO Traffic detection and Tracking
Exit
successfully exited

Collaboration diagram:

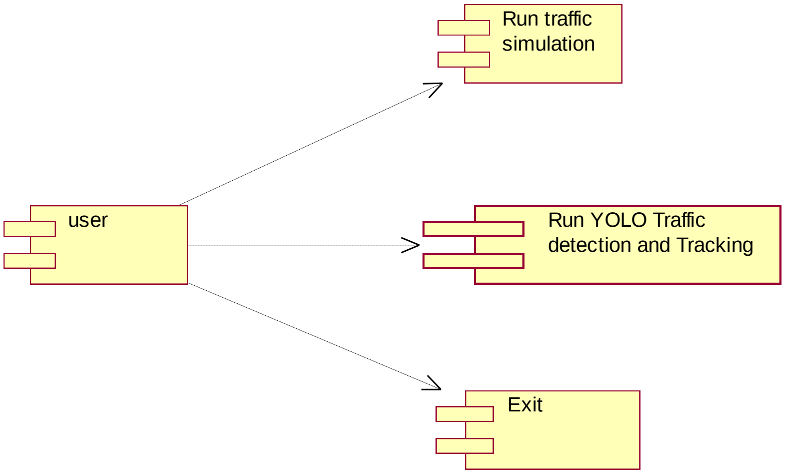
A collaboration diagram describes interactions among objects in terms of sequenced messages. Collaboration diagrams represent a combination of information taken from class, sequence, and use case diagrams describing both the static structure and dynamic behaviour of a system.


Exit
5: exit
6: successfully exited
4: successfully Run YOLO Traffic detection and Tracking
1: Run traffic simulation
Run YOLO Traffic detection and Tracking
user
Run traffic simulation
3: Run YOLO Traffic detection and Tracking
2: successfully Run traffic simulation

Component Diagram:

In the Unified Modelling 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.

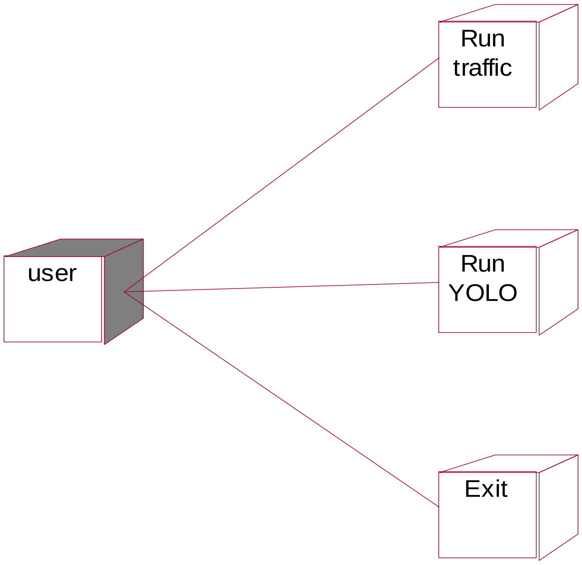
Components are wired together by using an assembly connector to connect the required interface of one component with the provided interface of another component. This illustrates the service consumer - service provider relationship between the two components.



Deployment Diagram:

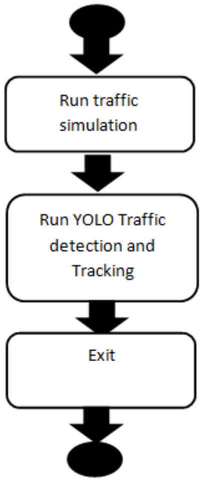
A **deployment diagram** in the Unified Modeling Language models the *physical* deployment of artifacts on nodes. To describe a web site, for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server), what software 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:

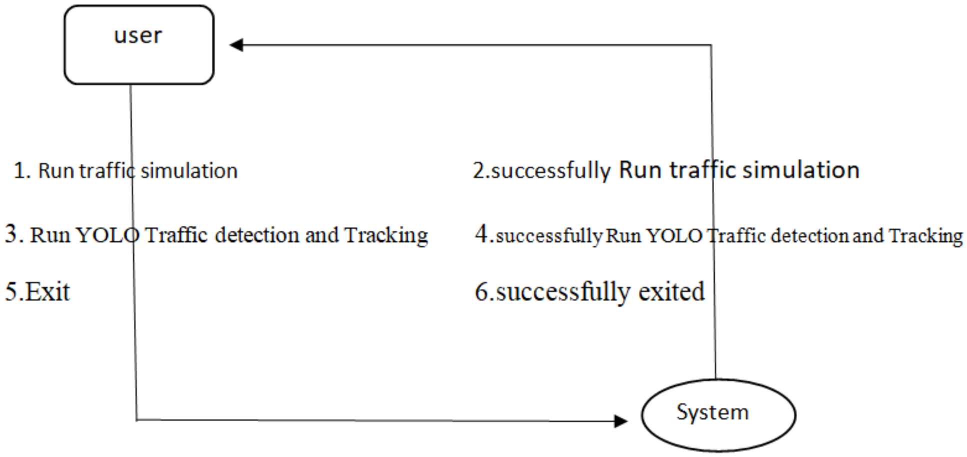
Activity diagram is another important diagram in UML to describe dynamic aspects of the system. It is basically a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another.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


user
1. Run traffic simulation
2.successfully Run traffic simulation
3. Run YOLO Traffic detection and Tracking
4.successfully Run YOLO Traffic detection and Tracking
5.Exit
6.successfully exited
System

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 software's 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:

```
from tkinter import messagebox
from tkinter import *
from tkinter import simpledialog
import tkinter
from tkinter import filedialog
from tkinter.filedialog import askopenfilename
from traffic_simulation import *
from yolo_traffic import *

main = tkinter.Tk()
main.title("Smart Control of Traffic Light Using Artificial Intelligence")
main.geometry("1300x1200")

global filename

def yoloTrafficDetection():
    global filename
    filename = filedialog.askopenfilename(initialdir="Videos")
    pathlabel.config(text=filename)
    text.delete('1.0', END)
    text.insert(END,filename+" loaded\n");
    runYolo(filename)

def runSimulation():
    sim = Simulation()
    sim.runSimulation()

def exit():
    main.destroy()


font = ('times', 16, 'bold')
title = Label(main, text='Smart Control of Traffic Light Using Artificial Intelligence')
title.config(bg='light cyan', fg='pale violet red')
title.config(font=font)
title.config(height=3, width=120)
title.place(x=0,y=5)

font1 = ('times', 14, 'bold')
simulationButton = Button(main, text="Run Traffic Simulation", command=runSimulation)
simulationButton.place(x=50,y=100)
simulationButton.config(font=font1)

pathlabel = Label(main)
pathlabel.config(bg='light cyan', fg='pale violet red')
pathlabel.config(font=font1)
pathlabel.place(x=460,y=100)

yoloButton = Button(main, text="Run Yolo Traffic Detection & Counting",
command=yoloTrafficDetection)
yoloButton.place(x=50,y=150)
yoloButton.config(font=font1)

exitButton = Button(main, text="Exit", command=exit)
exitButton.place(x=460,y=150)
exitButton.config(font=font1)


font1 = ('times', 12, 'bold')
text=Text(main,height=20,width=150)
scroll=Scrollbar(text)
text.configure(yscrollcommand=scroll.set)
text.place(x=10,y=250)
text.config(font=font1)
```

```
main.config(bg='snow3')
main.mainloop()
```

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 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 use 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.

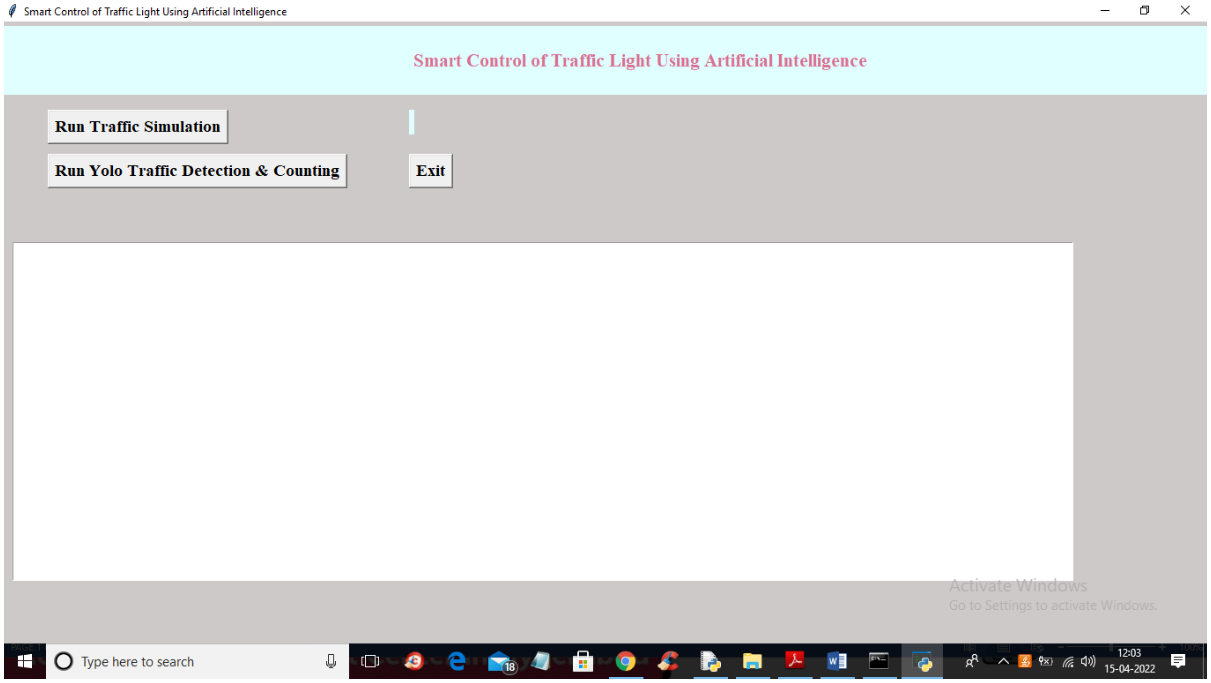
Acceptance Testing

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.

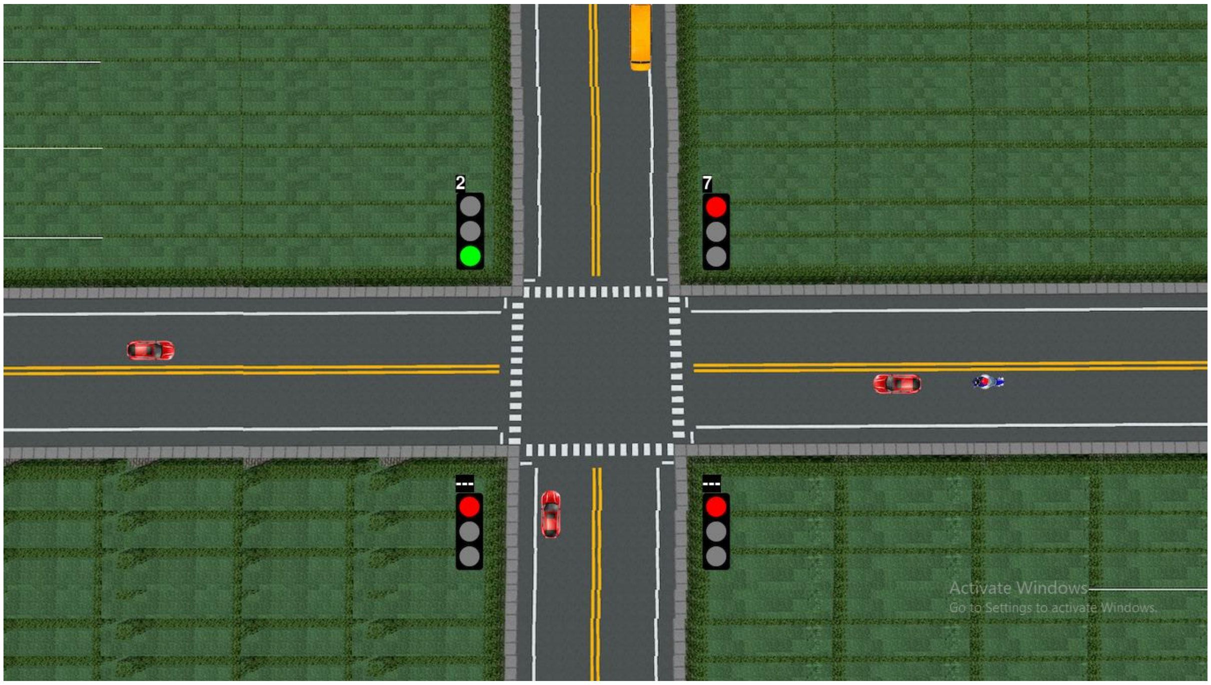
Test Case Id	**Test Case Name**	**Test Case Desc.**	**Test Steps**			**Test Case Status**	**Test Priority**
			Step	**Expected**	**Actual**		
01	Run traffic simulation	Test whether the Run traffic simulation or not	If the traffic simulation may not Run	We cannot do further operations	Personality Dataset uploaded we will do further operations	High	High
02	Run yolo traffic detection and tracking	Test whether Run yolo traffic detection and tracking or not	If the Run yolo traffic detection and tracking may not uploaded	We cannot do further operations	we will do further operations	High	High
03	EXIT	Test whether EXIT or not	If the may Not EXIT	We cannot do further operations	we will do further operations	High	High

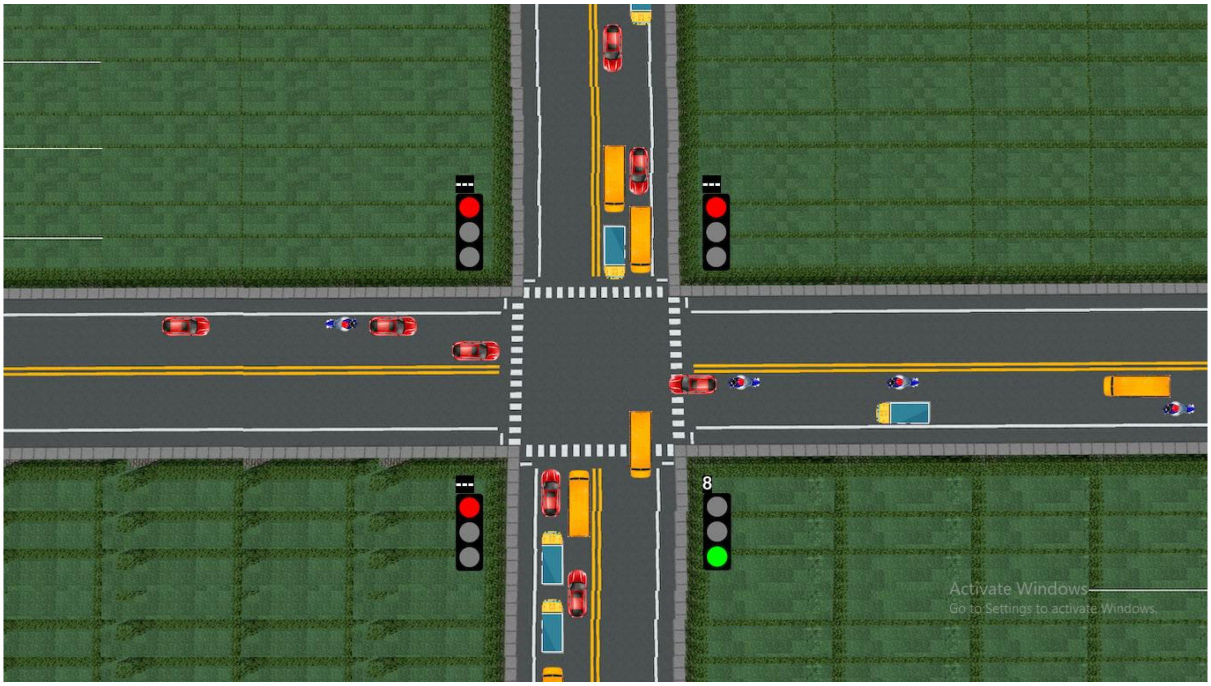
SCREEN SHOTS:

To run project double click on 'run.bat' file to get below output

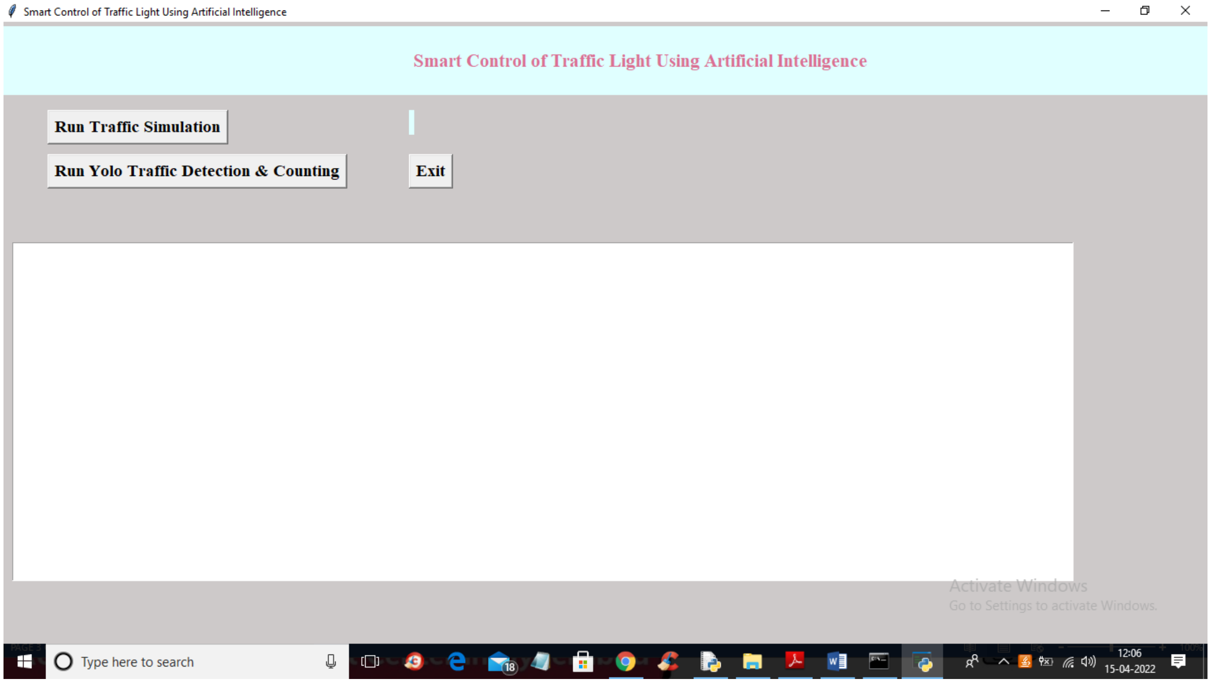

Smart Control of Traffic Light Using Artificial Intelligence
Smart Control of Traffic Light Using Artificial Intelligence
Run Traffic Simulation
Run Yolo Traffic Detection & Counting
Exit
Type here to search
12:03
15-04-2022

In above screen click on 'Run Traffic Simulation' button to start PYGAME simulation and get below output

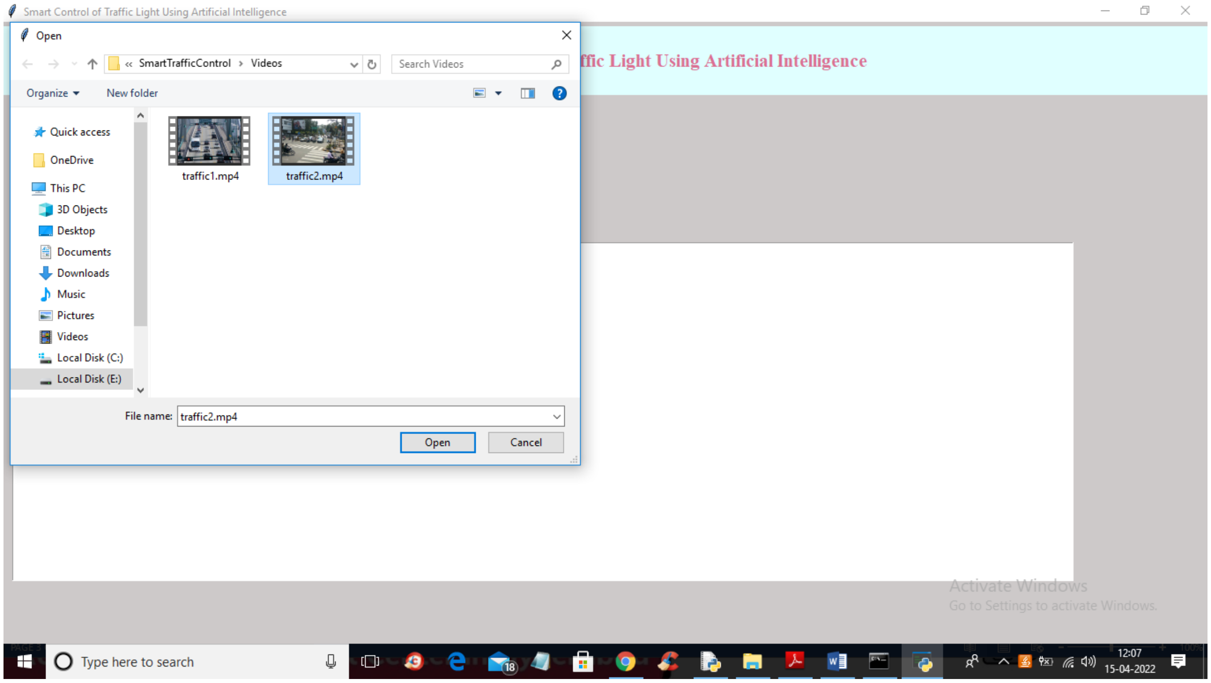




In above screen you can see PYGAME simulation output and at each lane traffic density is calculated and then adjust green and red line. This simulation run in INFINITE loop so you press ‘windows’ key from keyboard and then close application and then restart and run second YOLO module

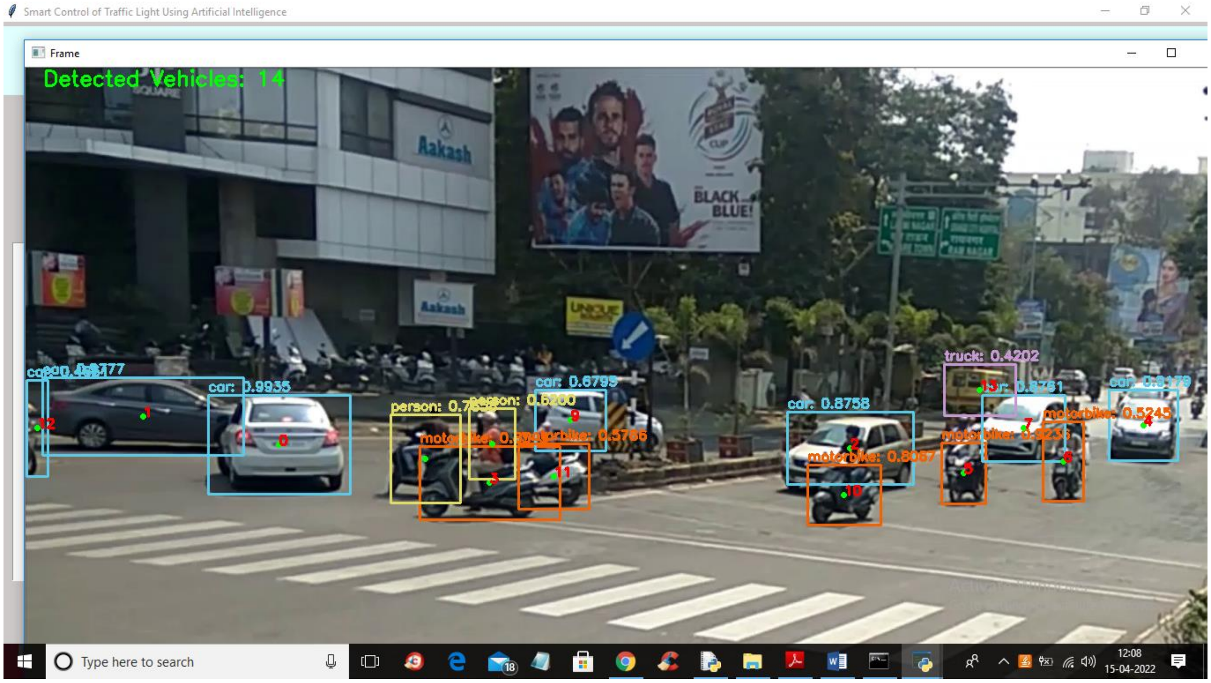

Smart Control of Traffic Light Using Artificial Intelligence
Smart Control of Traffic Light Using Artificial Intelligence
Run Traffic Simulation
Run Yolo Traffic Detection & Counting
Exit
Activate Windows
Go to Settings to activate Windows.
Type here to search
12:06
15-04-2022

Now in above screen click on ‘Run Yolo Traffic Detection & Counting’ button to upload traffic video and then estimate traffic density



File name: traffic2.mp4

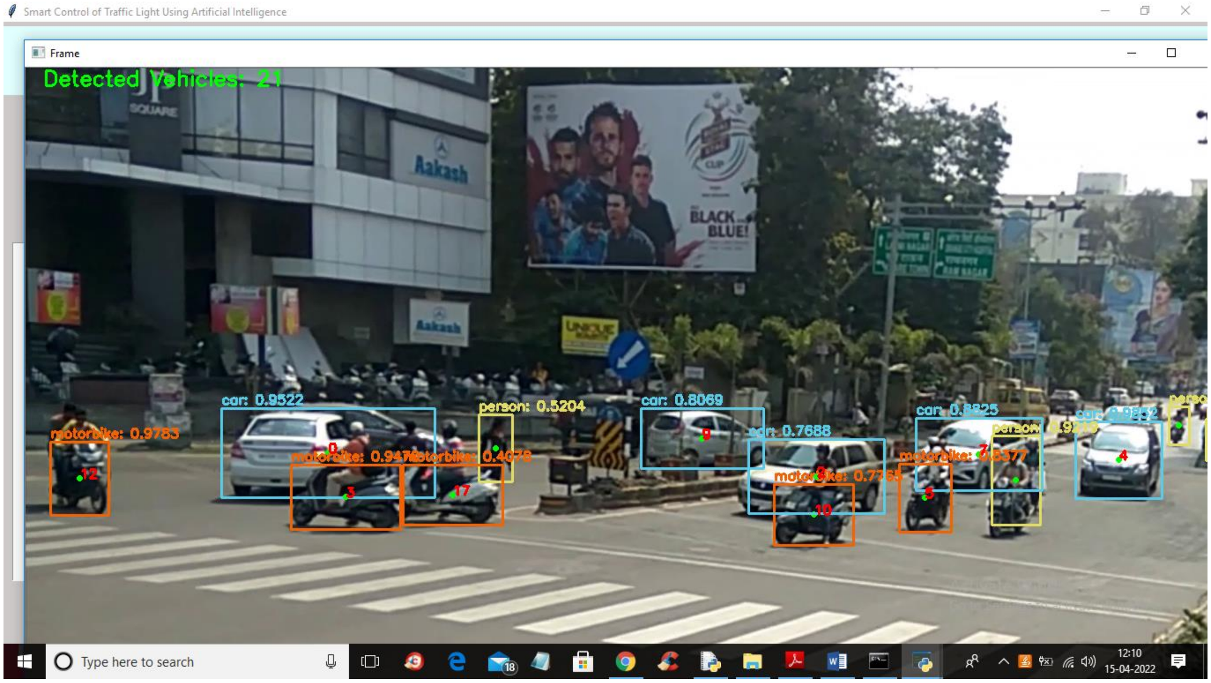
In above screen selecting and uploading ‘traffic2.mp4’ video and then click on ‘Open’ button to get below output



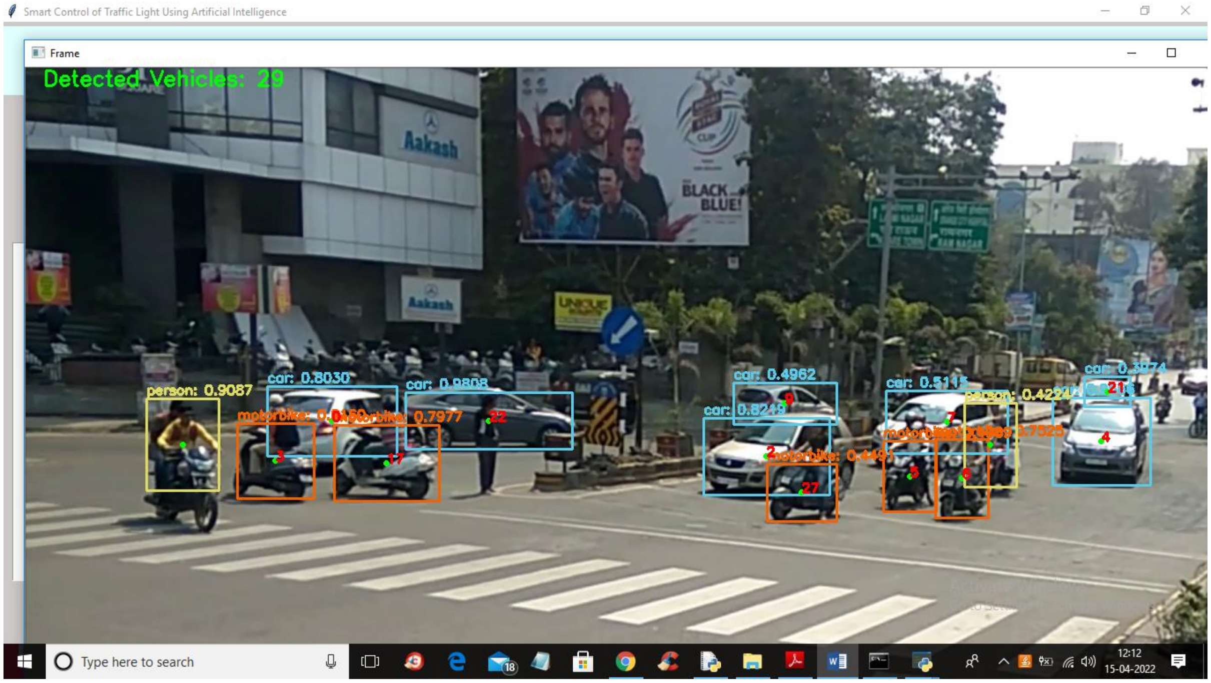
Detected Vehicles: 14

In above screen detecting traffic and then estimating its count and based on that traffic time will be adjusted. YOLO runs very slowly in normal laptop so let it finish all frame processing then u will get output.mp4 file which you can play as normal video with traffic density.

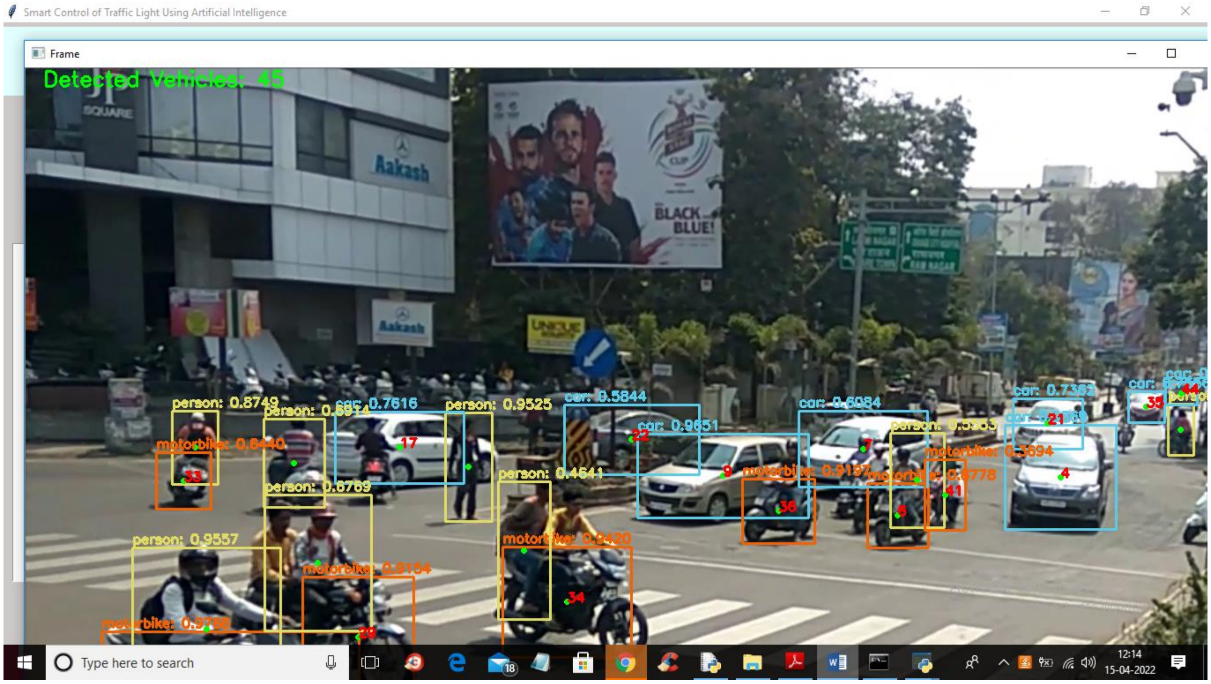
KH



Smart Control of Traffic Light Using Artificial Intelligence



Smart Control of Traffic Light Using Artificial Intelligence



8. CONCLUSION

Our method is proposing a solution to lessen down congestion on roads and will also look after to reduce accidents. We proved here the solution of daily traffic and fatal accidents. Thus above proposed theory will make our roads safer place to travel.

In future detectors can be placed in emergency vehicles so our traffic signal can easily detect. Control center can work automated removing all manual help.

9. REFERENCES

[1] TomTom.com, 'Tom Tom World Traffic Index', 2019. [Online]. Available: https://www.tomtom.com/en_gb/traffic-index/ranking/

[2] Khushi, "Smart Control of Traffic Light System using Image Processing," 2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC), Mysore, 2017, pp. 99-103, doi: 10.1109/CTCEEC.2017.8454966.

[3] A. Vogel, I. Oremović, R. Šimić and E. Ivanjko, "Improving Traffic Light Control by Means of Fuzzy Logic," 2018 International Symposium ELMAR, Zadar, 2018, pp. 51-56, doi: 10.23919/ELMAR.2018.8534692.

[4] A. A. Zaid, Y. Suhweil and M. A. Yaman, "Smart controlling for traffic light time," 2017 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), Aqaba, 2017, pp. 1-5, doi: 10.1109/AEECT.2017.8257768.

[5] Renjith Soman "Traffic Light Control and Violation Detection Using Image Processing”.” IOSR Journal of Engineering (IOSRJEN), vol. 08, no. 4, 2018, pp. 23-27

[6] A. Kanungo, A. Sharma and C. Singla, "Smart traffic lights switching and traffic density calculation using video processing," 2014 Recent Advances in Engineering and Computational Sciences (RAECS), Chandigarh, 2014, pp. 1-6, doi: 10.1109/RAECS.2014.6799542.
[7] Siddharth Srivastava, Subhadeep Chakraborty, Raj Kamal, Rahil, Minocha, “Adaptive traffic light timer controller” , IIT KANPUR, NERD MAGAZINE
[8] Ms. Saili Shinde, Prof. Sheetal Jagtap, Vishwakarma Institute Of Technology, Intelligent traffic management system:a Review, IJIRST 2016
[9] Open Data Science, ‘Overview of the YOLO Object Detection Algorithm’, 2018. [Online]. Available: https://medium.com/@ODSC/ overview-of-the-yolo-object-detection-algorithm-7b52a745d3e0
[10] J. Hui, ‘Real-time Object Detection with YOLO, YOLOv2 and now YOLOv3’, 2018. [Online]. Available: https://medium.com/ @jonathan _hui/ real-time-object-detection-with-yolo-yolov2-28b1b93e2088
[11] J. Redmon, ‘Darknet: Open Source Neural Networks in C’, 2016. [Online]. Available: https://pjreddie.com/darknet/
[12] Tzutalin, ‘LabelImg Annotation Tool’, 2015. [Online]. Available: https://github.com/tzutalin/labelImg [13] Li, Z., Wang, B., and Zhang, J. “Comparative analysis of drivers' start- up time of the first two vehicles at signalized intersections”, 2016 J. Adv. Transp., 50: 228–239. doi: 10.1002/atr.1318
[14] Arkatkar, Shriniwas & Mitra, Sudeshna & Mathew, Tom. “India” in Global Practices on Road Traffic Signal Control, ch.12, pp.217-242