SMART TRAFFIC MANAGEMENT SYSTEM USING DEEP LEARNING

Project Report

This thesis submitted

in partial fulfillment of the requirement for the award of degree of

MASTER OF COMPUTER APPLICATIONS (MCA)

Submitted by
Ms. K. NANDINI
(Regd.No.22L31F0017)

under the esteemed guidance of

Mrs. SRI JYOTHI

DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS

VIGNAN'S INSTITUTE OF INFORMATION TECHNOLOGY

(Autonomous)

Affiliated to JNTUGV, Vizianagaram & Approved by AICTE, New Delhi

Re-Accredited by NAAC (CGPA of 3.41/ 4.00)

ISO 9001:2008, ISO 14001:2004, OHSAS 18001:2007 Certified Institution

VISAKHAPATNAM – 530039

VIGNAN's INSTITUTE OF INFORMATION TECHNOLOGY
(AUTONOMOUS)
(Approved by AICTE-New Delhi & Affiliated to JNTUGV,Vizianagaram)
Beside VSEZ, Duvvada, Vadlapudi Post, Gajuwaka, Visakhapatnam - 530 049.

MASTER OF COMPUTER APPLICATIONS

CERTIFICATE

This is to certify that the Mini Project report entitled "**Smart Traffic Management System Using Deep Learning**" is a bonafide record of project work carried out under my supervision by K. Nandini bearing Regd. No 22L31F0017, in partial fulfillment of the degree of **Master of Computer Applications** of Vignan Institute of Information Technology(A) affiliated to Jawaharlal Nehru Technology University Vizianagaram, during the academic year 2023-2024.

Signature of Project Guide	**Head of the Department**
Mrs. SRI JYOTHI	Dr. G. Neelima
Department of MCA	Associate Professor & HOD
	Department of MCA

External Examiner

VIGNAN's INSTITUTE OF INFORMATION TECHNOLOGY
(AUTONOMOUS)
(Approved by AICTE-New Delhi & Affiliated to JNTUGV,Vizianagaram)
Beside VSEZ, Duvvada, Vadlapudi Post, Gajuwaka, Visakhapatnam - 530 049.

DECLARATION

I hereby declare that this project report entitled "**SMART TRAFFIC MANAGEMENT SYSTEM**" has undertaken by us for the fulfillment of Degree in Master ofComputer Applications. I declare that this project report has not been submitted anywhere in the part of fulfillment for any degree of any other University.

PLACE: Visakhapatnam

DATE:

K. NANDINI
(22L31F0017)

VIGNAN's INSTITUTE OF INFORMATION TECHNOLOGY (AUTONOMOUS)

(Approved by AICTE-New Delhi & Affiliated to JNTUGV,Vizianagaram)

Beside VSEZ, Duvvada, Vadlapudi Post, Gajuwaka, Visakhapatnam - 530 049.

ACKNOWLEDGEMENT

An endeavor over a long period can be successfully with the advice andsupport of many well-wishers. I take this opportunity to express ourgratitude and appreciation to all of them.

I express my sincere gratitude to our internal guide, **Mrs. SRI JYOTHI** for his encouragement and cooperation in completion of our project. We are very fortunate in getting the generoushelp and constant encouragement from him.

I would be very grateful to our project coordinator, **Mrs. A. SIRISHA** for thecontinuous monitoring of our project work. We truly appreciate for her timeand effort spent.

I would like to thank our Head of the Department **Dr. G. NEELIMA** and all other teaching and non-teaching staff of the department fortheir cooperation and guidance during our project.

I sincerely thank to **Dr. SUDHAKAR JYOTHULA**, Principal of VIGNAN'S INSTITUTE OF INFORMATION TECHNOLOGY (A) for her inspirationto undergo this project.

I wanted to convey my sincere gratitude to **Dr. V. MADHUSUDHAN RAO**, Rector of VIGNAN’S INSTITUTE OF INFORMATION TECHNOLOGY (A) for allocating the required resources and for the knowledge sharing during my project work.

I extended our grateful thanks to our honorable Chairman **Dr. L. RATHAIHA** for giving us an opportunity to study in his esteemedinstitution.

K. Nandini

(22L31F0017)

VIGNAN's INSTITUTE OF INFORMATION TECHNOLOGY
(AUTONOMOUS)
(Approved by AICTE-New Delhi & Affiliated to JNTUGV,Vizianagaram)
Beside VSEZ, Duvvada, Vadlapudi Post, Gajuwaka, Visakhapatnam - 530 049.

MASTER OF COMPUTER APPLICATIONS

Vision of the Department:

We aim to generate groomed, technical competent and skilled intellectual professionals with ethical/human values to serve as a valuable resource for modern industry and current society.

Mission of the Department:

- Providing strong theoretical and practical knowledge in computer science discipline with an emphasis on software development.
- To provide need-based quality training in the field of information technology.
- Impart quality education to meet global standards and achieve excellence in teaching-learning and research.
- To provide students with the tools to become productive, participating global citizens and life-long learners.

VIGNAN's INSTITUTE OF INFORMATION TECHNOLOGY
(AUTONOMOUS)
(Approved by AICTE-New Delhi & Affiliated to JNTUGV,Vizianagaram)
Beside VSEZ, Duvvada, Vadlapudi Post, Gajuwaka, Visakhapatnam - 530 049.

MASTER OF COMPUTER APPLICATIONS

Vision of the Institute (VIIT):

We envision being recognized leader in technical education. We shall aim at national excellence by creating competent and socially conscious technical manpower for the current and future Industrial requirements and development of the nation.

Mission of the Institute (VIIT):

- Introducing innovative practices of Teaching and Learning.
- Undertaking research and development in thrust areas.
- Continuously collaborating with industry.
- Promoting a strong set of ethical values.
- Serving the surrounding region and nation at large.

VIGNAN's INSTITUTE OF INFORMATION TECHNOLOGY (AUTONOMOUS)
(Approved by AICTE-New Delhi & Affiliated to JNTUGV,Vizianagaram)
Beside VSEZ, Duvvada, Vadlapudi Post, Gajuwaka, Visakhapatnam - 530 049.

MASTER OF COMPUTER APPLICATIONS

PROGRAMOUTCOMES	
PO1	**Engineering Knowledge:** Apply the knowledge of mathematics science engineering fundamentals and mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems and engineering problems.
PO2	**Problem analysis:** Identify, formulate, review research Literature, and analyse complex engineering problems reaching substantiated conclusions using the first principles of mathematics, natural sciences, and engineering sciences
PO3	**Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety, and the cultural societal, and environmental considerations
PO4	**Conduct investigations of complex problems:** Use research-based knowledge and research methods including sign of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5	**Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with a understanding of the limitations.

PO6	**The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice
PO7	**Environment and sustainability:** Understand the impact of professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development and need for sustainable development.
PO8	**Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9	**Individual and teamwork:** Function effectively as an individual and as a member or leader in diverse teams and individual, and as a member or leader in diverse teams, and in multi-disciplinary settings.
PO10	**Communication:** Communicate effectively on complex engineering activities with engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11	**Project management and finance**: Demonstrate knowledge and understanding of the engineering and knowledge and understanding of the engineering and management principles and apply these to one's work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12	**Life-long learning:** Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

VIGNAN's INSTITUTE OF INFORMATION TECHNOLOGY
(AUTONOMOUS)
(Approved by AICTE-New Delhi & Affiliated to JNTUGV,Vizianagaram)
Beside VSEZ, Duvvada, Vadlapudi Post, Gajuwaka, Visakhapatnam - 530 049.

MASTER OF COMPUTER APPLICATIONS

Program Educational Objectives (PEOS):

The post graduations of Master of Computer Application will be able:

PEO1: To successful professional in industry, government sector, academia, research,

Entrepreneurial pursuit and consulting firms.

PEO2: To contribute to society as broadly educated, expressive, ethical and responsible citizens with proven expertise.

PEO3: To thrive to pursue life-long learning to fulfil their goals.

Program Specific Outcomes (PSOs):

MCA program has been designed to prepare graduates for attaining the following program

Specific outcomes:

PSO1: They can identity, critically analyse, formulate and develop computer applications.

PSO2: Function competently as an individual and as a leader in multidisciplinary projects.

ABSTRACT

The most important challenge to sustainable mobility is persistent congestions of differing strength and duration in the dense transport networks. The standard Adaptive Traffic Signal Control cannot properly address this kind of congestion. Deep learning-based mechanisms have proved their significance to anticipate in adjective outcomes to improve the decision making on the predictions of traffic length. The deep learning models have long been used in many application domains which needed the identification and prioritization of adverse factors for a simplifying human life. Several methods are being popularly used to handle real time problems occurring from traffic congestion. This study demonstrates the capability of DL models to overcome the traffic congestion by simply allowing the vehicles through a signal depending on the length of vehicles. Our proposed method integrates a numeral of approach, intended to advance the cooperativeness of the explore operation. In this work, we develop the application to regulate the traffic by releasing better signal at desired time intervals.

KEYWORDS: Traffic, YOLO, Deep Learning.

INDEX

LIST OF FIGURES

CHAPTER-1

INTRODUCTION

INTRODUCTION

1.1 MOTIVATION

The motivation behind employing deep learning for traffic prediction in intelligent transportation systems lies in the urgent need to combat persistent congestion in dense transport networks. Traditional methods like Adaptive Traffic Signal Control fall short in addressing varied congestion levels effectively. Deep learning models offer promise in accurately anticipating traffic outcomes, aiding decision-making for congestion predictions. By leveraging these models, this study aims to revolutionize traffic management, prioritizing efficient vehicle flow based on traffic length, enhancing cooperation among systems, and optimizing signal timing for smoother traffic regulation in real-time scenarios**.**

1.2 PROBLEM STATEMENT

The problem statement revolves around traffic prediction for Intelligent Transportation Systems (ITS) using deep learning. This involves the development of advanced neural network models capable of analyzing vast amounts of traffic data to forecast congestion, traffic flow, and travel times accurately. The aim is to enhance traffic management, reduce congestion, improve transportation efficiency, and promote the overall effectiveness of intelligent transportation systems in modern urban environments through data-driven deep learning approaches.

1.3 OBJECTIVE OF THE PROJECT

To detect the vehicles in the uploaded image and counts the number of vehicles, by using YOLO model.

1.4 SCOPE

The scope of this project encompasses the development and implementation of deep learning models and algorithms for traffic prediction within Intelligent Transportation Systems (ITS). This includes data collection, preprocessing, feature engineering, model architecture design, training, and validation. The project aims to forecast various traffic parameters, such as congestion, travel times, and traffic flow, using diverse data sources like sensors, GPS, and traffic cameras. Ethical considerations related to data privacy and responsible use of deep learning in traffic management will also be explored. The ultimate goal is to provide urban planners, transportation authorities, and commuters with reliable traffic predictions for enhanced mobility and efficiency.

1.5 PROJECT INTRODUCTION:

Over the last decade by solving many very complex and sophisticated real-world problems. The application areas included almost all the real-world domains such as healthcare, autonomous vehicle (AV), business applications, and image processing. DL algorithms' learning is typically based on trial-and-error method quite opposite of conventional algorithms, which follows the programming instructions based on decision statements like if-else. One of the most significant areas of DL is simplifying human problems, in many application areas including medical domain, governments every sector is showing their interest to introduce AI to their systems. Various models have wide applicability in working with the conditions of real time. There are lots of studies performed for regulating traffic using deep learning techniques such as image segmentation, object detection etc.., In particular, the study is focused on live traffic regulating near a traffic signal and study is also focused on the decreasing the waiting time depending on vehicle counts and early response. These systems can be very helpful in decision making to handle the present scenario to guide early interventions to manage these traffic regulations very effectively. This study aims to provide an better system which can be able to release the traffic depending on the count of vehicles. This project targets to develop web application in order to handle the "Traffic Congestion". Python-Flask is used as front end which is used to craft the user interface. MySQL is used as back end and used to craft the database and save the particulars. Anybody with a little computer knowledge can approach and deal with the software with ease; hence it can be termed user friendly.

CHAPTER-2

LITERATURE SURVEY

2. LITERATURE SURVEY

[1] Rutger Claes, Tom Holvoet, and Danny Weyns. A decentralized approach for anticipatory vehicle routing using delegate multiagent systems. IEEE Transactions on Intelligent Transportation Systems, 12(2):364–373, 2011.

Advanced vehicle guidance systems use real-time traffic information to route traffic and to avoid congestion. Unfortunately, these systems can only react upon the presence of traffic jams and not to prevent the creation of unnecessary congestion. Anticipatory vehicle routing is promising in that respect, because this approach allows directing vehicle routing by accounting for traffic forecast information. This paper presents a decentralized approach for anticipatory vehicle routing that is particularly useful in large-scale dynamic environments. The approach is based on delegate multiagent systems, i.e., an environment-centric coordination mechanism that is, in part, inspired by ant behavior. Antlike agents explore the environment on behalf of vehicles and detect a congestion forecast, allowing vehicles to reroute. The approach is explained in depth and is evaluated by comparison with three alternative routing strategies. The experiments are done in simulation of a real-world traffic environment. The experiments indicate a considerable performance gain compared with the most advanced strategy under test, i.e., a traffic-message-channel-based routing strategy.

[2] Mehul Mahrishi and Sudha Morwal. Index point detection and semantic indexing of videos - a comparative review. Advances in Intelligent Systems and Computing, Springer, 2020.

Mobile Ad Hoc Network (MANET) has the ability to self-configure and establish a mobile wireless mesh that can be used in extreme conditions, such as in areas affected by disasters. One of the routings in MANET is AODV routing. AODV is one of the reactive routing needed to send data. However, in the implementation of disaster conditions, AODV has weaknesses that are vulnerable to extreme environmental conditions. In this study, communication will be modeled that leads to disruption due to disaster. MANET AODV-DTN is used to improve network performance. With this system, the Probability Delivery Ratio (PDR) parameter value can be increased as evidenced by the variable modification of the number of nodes to be 0.431%, reducing the average delay by 63.525%, and producing the energy consumption increased by 0.170%. Simulation with the variable modification of speed obtained by PDR 0.482%, reducing the average delay by 78.710% and energy consumption increased by 0.167%. Modification of buffer size variables obtained 0.729% PDR results, reducing the average delay of

71.603% and energy consumption increased by 0.161%. From these data, MANET AODV-DTN is better than MANET AODV.

[3] C. Zhang, P. Patras, and H. Haddadi. Deep learning in mobile and wireless networking: A survey. IEEE Communications Surveys Tutorials, 21(3):2224–2287, third quarter 2019.

The rapid uptake of mobile devices and the rising popularity of mobile applications and services pose unprecedented demands on mobile and wireless networking infrastructure. Upcoming 5G systems are evolving to support exploding mobile traffic volumes, real-time extraction of fine-grained analytics, and agile management of network resources, so as to maximize user experience. Fulfilling these tasks is challenging, as mobile environments are increasingly complex, heterogeneous, and evolving. One potential solution is to resort to advanced machine learning techniques, in order to help manage the rise in data volumes and algorithm-driven applications. The recent success of deep learning underpins new and powerful tools that tackle problems in this space. In this paper we bridge the gap between deep learning and mobile and wireless networking research, by presenting a comprehensive survey of the crossovers between the two areas. We first briefly introduce essential background and state-of-theart in deep learning techniques with potential applications to networking. We then discuss several techniques and platforms that facilitate the efficient deployment of deep learning onto mobile systems. Subsequently, we provide an encyclopedic review of mobile and wireless networking research based on deep learning, which we categorize by different domains. Drawing from our experience, we discuss how to tailor deep learning to mobile environments. We complete this survey by pinpointing current challenges and open future directions for research.

CHAPTER-3

SYSTEM ANALYSIS

3.SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

To fully understand just how powerful this traffic congestion problem is, we need to look at it from the very beginning which is traditional way of controlling traffic. Traditional way basically uses, a person should make traffic observations to estimate or provide the clearance to the lane which consists of high count of vehicles. Traditional method later updated by using remotes-controlled system to give the right signal to the lanes. And these practices fail because the persons may not be available at every time at the traffic center.

3.1.1 DISADVANTAGES:

1. Low Efficiency: The system operates with suboptimal resource utilization, often resulting in sluggish performance and inefficiencies in delivering desired outcomes.

2. Time Consuming: Requires a substantial amount of time to complete tasks or processes, which may lead to delays or inefficiencies.

3. High Complexities: Involves intricate and interconnected elements, demanding specialized knowledge or skills to manage or understand effectively.

4. Resources Consuming: Utilizes significant resources, such as computational power, manpower, or financial investment, potentially straining available assets.

3.2 PROPOSED SYSTEM

We propose this application that can be considered a useful system since it helps to reduce the limitations obtained traditional methods. By providing support through the a DL based analysis, it can be able to generate best results for attributes without any overlap. The system is developed in a Flask based Python environment. The models involved in this application are YOLO.

3.2.1ADVANTAGES:

1. High Efficiency: The system operates with optimal resource utilization, achieving desired outcomes promptly and effectively while minimizing waste, enhancing productivity.

2. Time Saving: Tasks are completed quickly, reducing time requirements and streamlining processes for improved efficiency.

3. Low Complexities: Involves straightforward and manageable elements, making tasks easy to understand and execute with minimal intricacies or complications.

CHAPTER-4

SYSTEM SPECIFICATIONS

4. SYSTEM SPECIFICATIONS

4.1 Functional and non-functional requirements

Requirement’s analysis is very critical process that enables the success of a system or software project to be assessed. Requirements are generally split into two types: Functional and non-functional requirements.

Functional Requirements: These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

Examples of functional requirements:

1) Authentication of user whenever he/she logs into the system
2) System shutdown in case of a cyber-attack
3) A verification email is sent to user whenever he/she register for the first time on some software system.

Non-functional requirements: These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioral requirements. They basically deal with issues like:

- Portability
- Security
- Maintainability
- Reliability
- Scalability
- Performance
- Reusability
- Flexibility

Examples of non-functional requirements:

1) Emails should be sent with a latency of no greater than 12 hours from such an activity.

2) The processing of each request should be done within 10 seconds
3) The site should load in 3 seconds whenever of simultaneous users are > 10000

4.2 HARDWARE CONFIGURATION:

• Processor	- I3/Intel Processor
• RAM	- 4GB (min)
• Hard Disk	- 128 GB

4.3 SOFTWARE CONFIGURATION:

• Operating System	:	Windows 7+
• Server-side Script	:	Python 3.6+
• IDE	:	PyCharm
• Libraries Used	:	Pandas, Numpy, Sklearn, Yolo
• Framework	:	Flask

CHAPTER-5

SYSTEM ENIVORNMENT

5. SYSTEM ENIVORNMENT

5.1 PYTHON PROGRAMMING LAUGUAGE

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Python Features

Python's features include −

- **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
- **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** − Python provides interfaces to all major commercial databases.
- **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** − Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

- It supports functional and structured programming methods as well as OOP.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection..

5.2 FLASK FRAMEWORK

Flask is a web application framework written in Python. Armin Ronacher, who leads an international group of Python enthusiasts named Pocco, develops it. Flask is based on Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects.

By default, the Flask route responds to the **GET** requests. However, this preference can be altered by providing methods argument to **route ()** decorator.

In order to demonstrate the use of **POST** method in URL routing, first let us create an HTML form and use the **POST** method to send form data to a URL.

Flask has been used here to create a user-friendly web interface for our IDS. Security personal can be used this interface to:

1. View real-time data on network traffic or system activity.
2. Monitor the results of the ML model, including classifications of normal and anomalous events.
3. Visualize trends and patterns in network activity using charts and graphs.

Benefits of using Flask:

- **Flexibility:** Flask allows you to tailor the interface and functionalities to your specific needs.
- **Ease of Use:** For developers familiar with Python, Flask offers a relatively simple way to build a web interface for your IDS.
- **Open Source:** Being open-source, Flask allows for customization and community support.

CHAPTER-6

SYSTEM DESIGN

6. SYSTEM DESIGN

6.1 FLOW OF THE PROJECT:

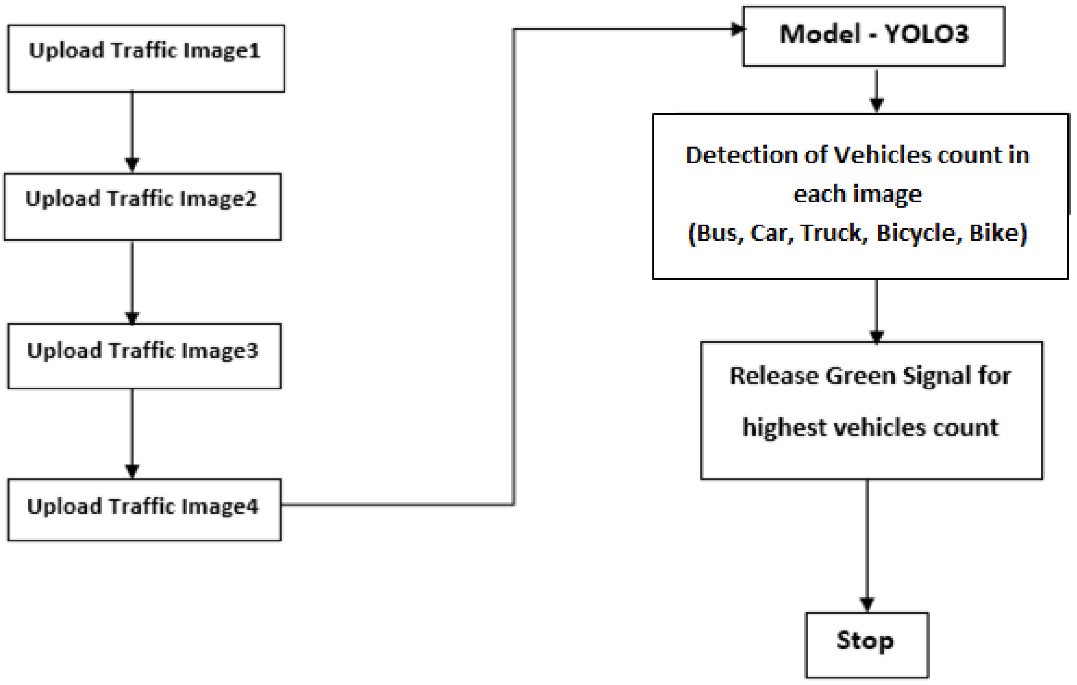
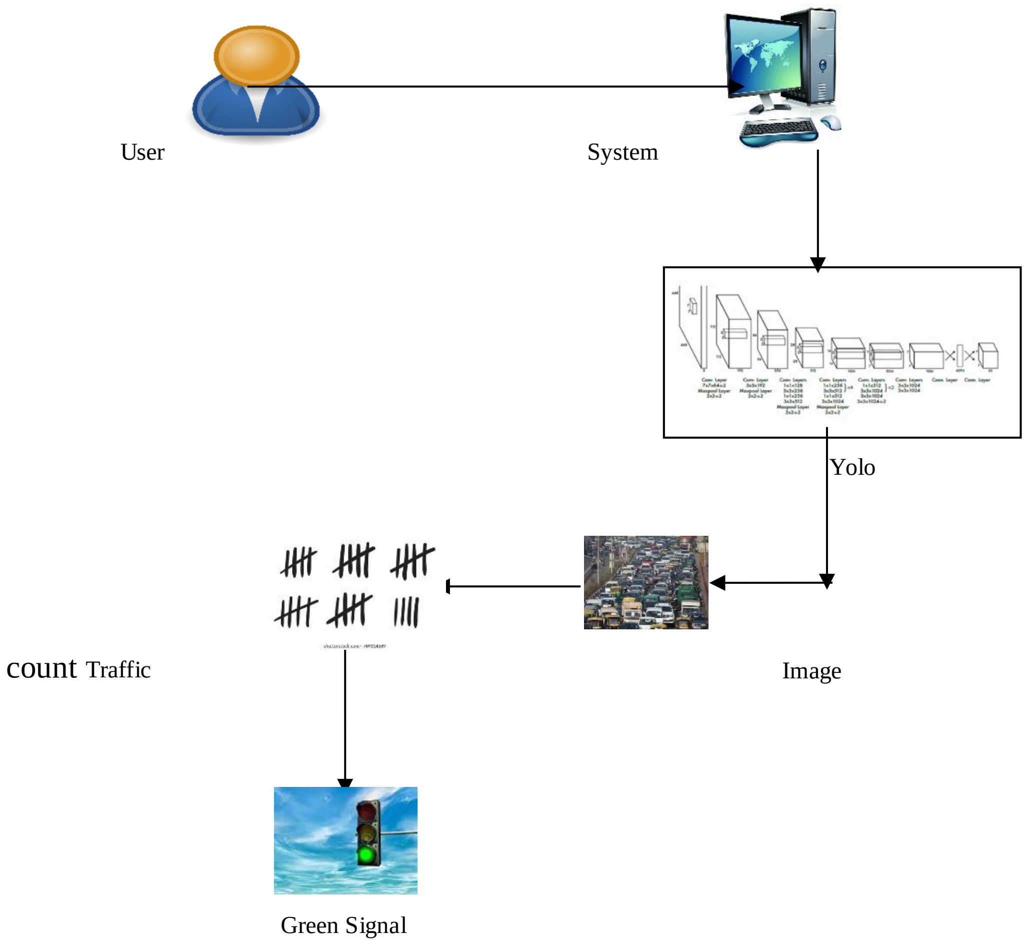

Upload Traffic Image1
Upload Traffic Image2
Upload Traffic Image3
Upload Traffic Image4
Model - YOLO3
Detection of Vehicles count in each image (Bus, Car, Truck, Bicycle, Bike)
Release Green Signal for highest vehicles count
Stop

Fig: Flow chart diagram

6.1.2 SYSTEM ARCHITECTURE



Introduction of Input Design:

In an information system, input is the raw data that is processed to produce output. During the input design, the developers must consider the input devices such as PC, MICR, OMR, etc.

Therefore, the quality of system input determines the quality of system output. Well-designed input forms and screens have following properties −

- It should serve specific purpose effectively such as storing, recording, and retrieving the information.

- It ensures proper completion with accuracy.
- It should be easy to fill and straightforward.
- It should focus on user's attention, consistency, and simplicity.
- All these objectives are obtained using the knowledge of basic design principles regarding −
    - What are the inputs needed for the system?
    - How end users respond to different elements of forms and screens?

Objectives for Input Design:

The objectives of input design are −

- To design data entry and input procedures
- To reduce input volume
- To design source documents for data capture or devise other data capture methods
- To design input data records, data entry screens, user interface screens, etc.
- To use validation checks and develop effective input controls.

Output Design: The design of output is the most important task of any system. During output design, developers identify the type of outputs needed, and consider the necessary output controls and prototype report layouts.

Objectives of Output Design:

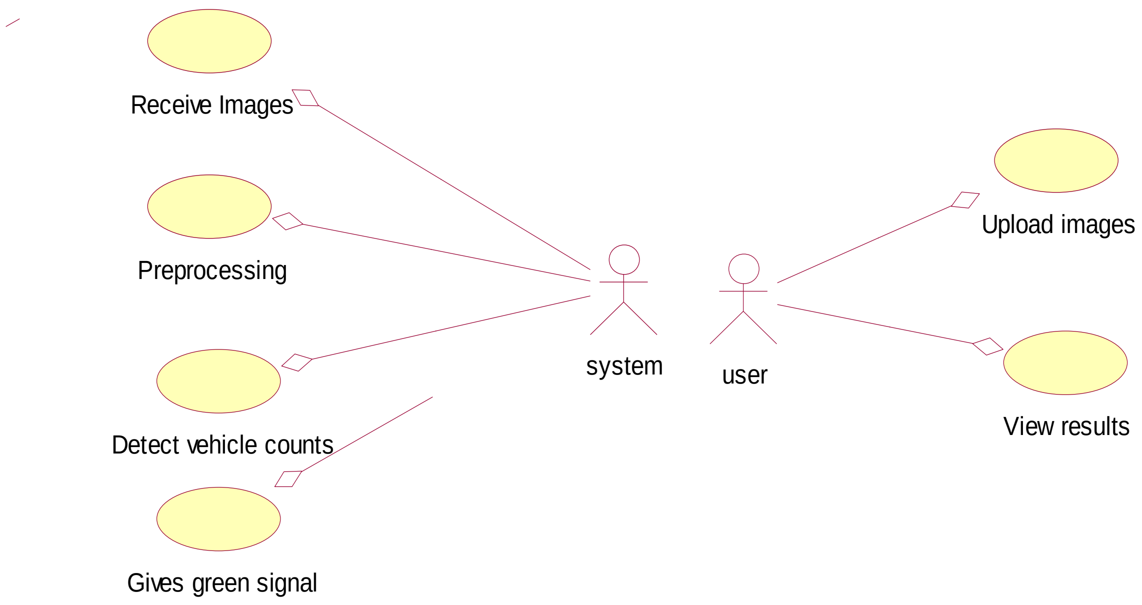
The objectives of input design are:

- To develop output design that serves the intended purpose and eliminates the production of unwanted output.
- To develop the output design that meets the end user's requirements.
- To deliver the appropriate quantity of output.
- To form the output in appropriate format and direct it to the right person.
- To make the output available on time for making good decisions.

6.2 UML DAIGRAM

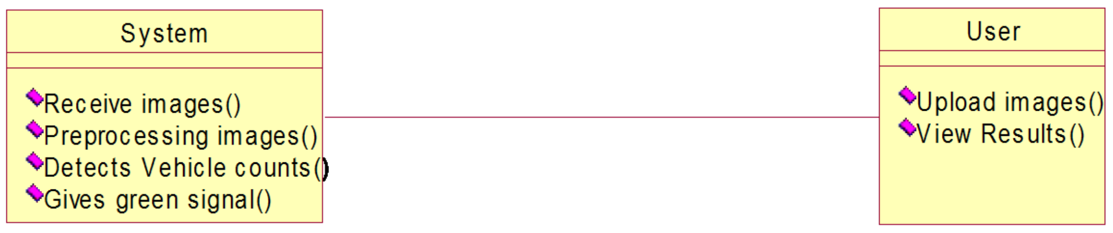
6.2.1 USE CASE DIAGRAM

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



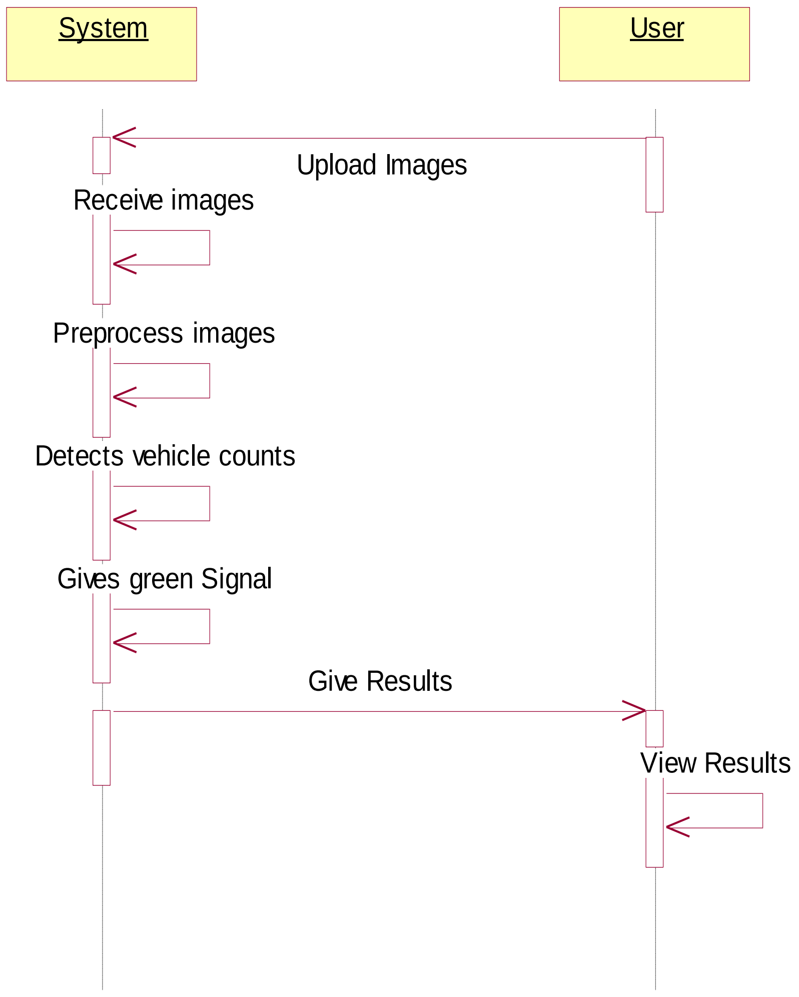
6.2.2 CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



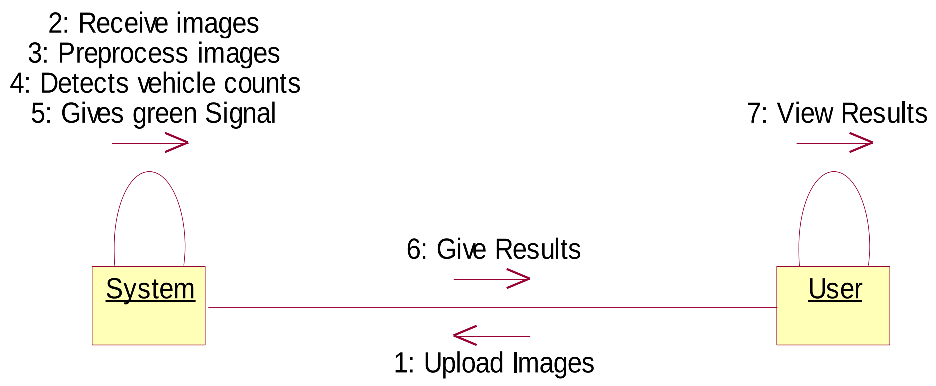
6.2.3 SEQUENCE DIAGRAM:

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



6.2.4 COLLABORATION DIAGRAM:

In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization whereas the collaboration diagram shows the object organization.



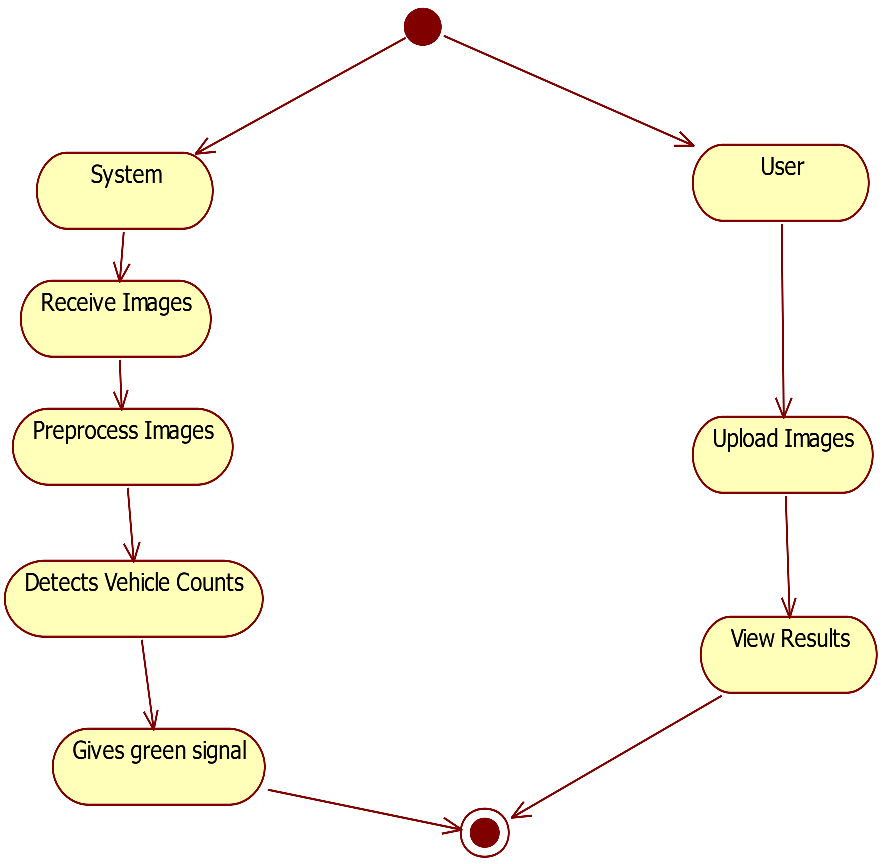
6.2.5 DEPLOYMENT DIAGRAM

Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware used to deploy the application.

System
User

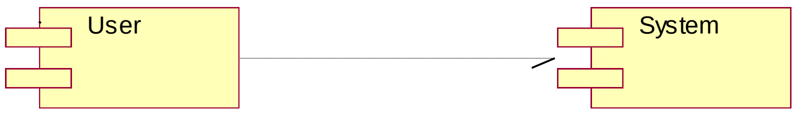
6.2.6 ACTIVITY DIAGRAM:

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



6.2.7 COMPONENT DIAGRAM

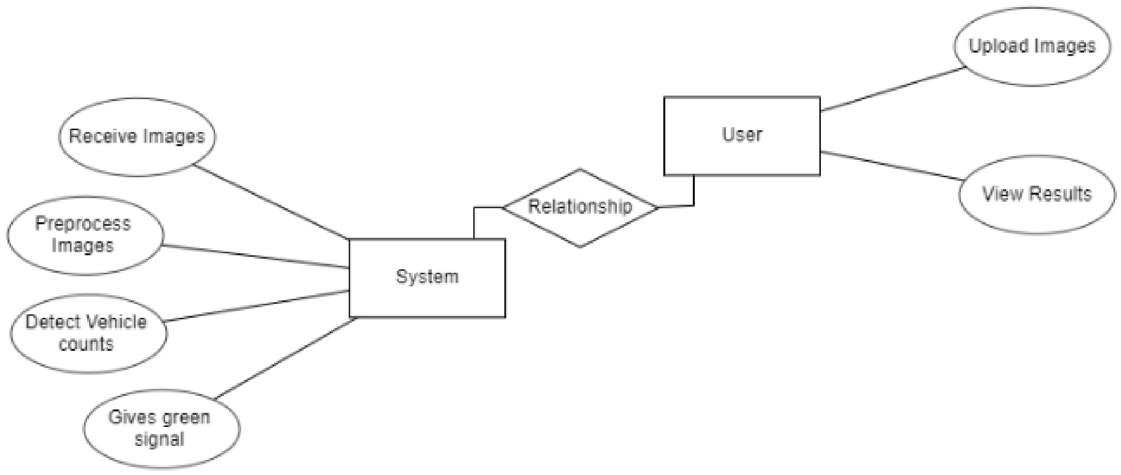
A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical **c**omponents in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required function is covered by planned development.



6.2.8 ER DIAGRAM:

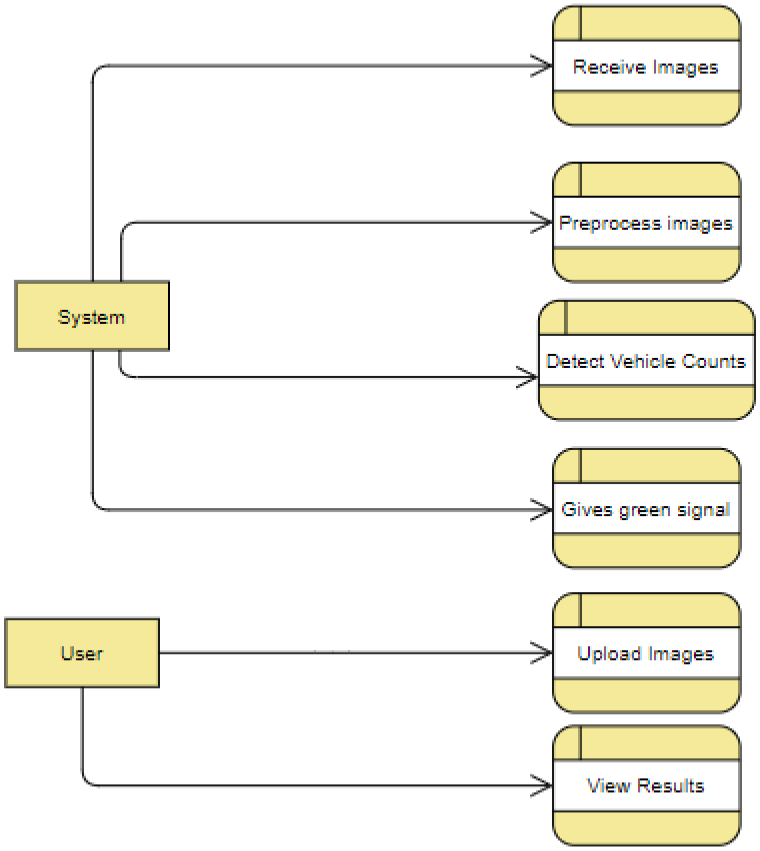
An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram). An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.

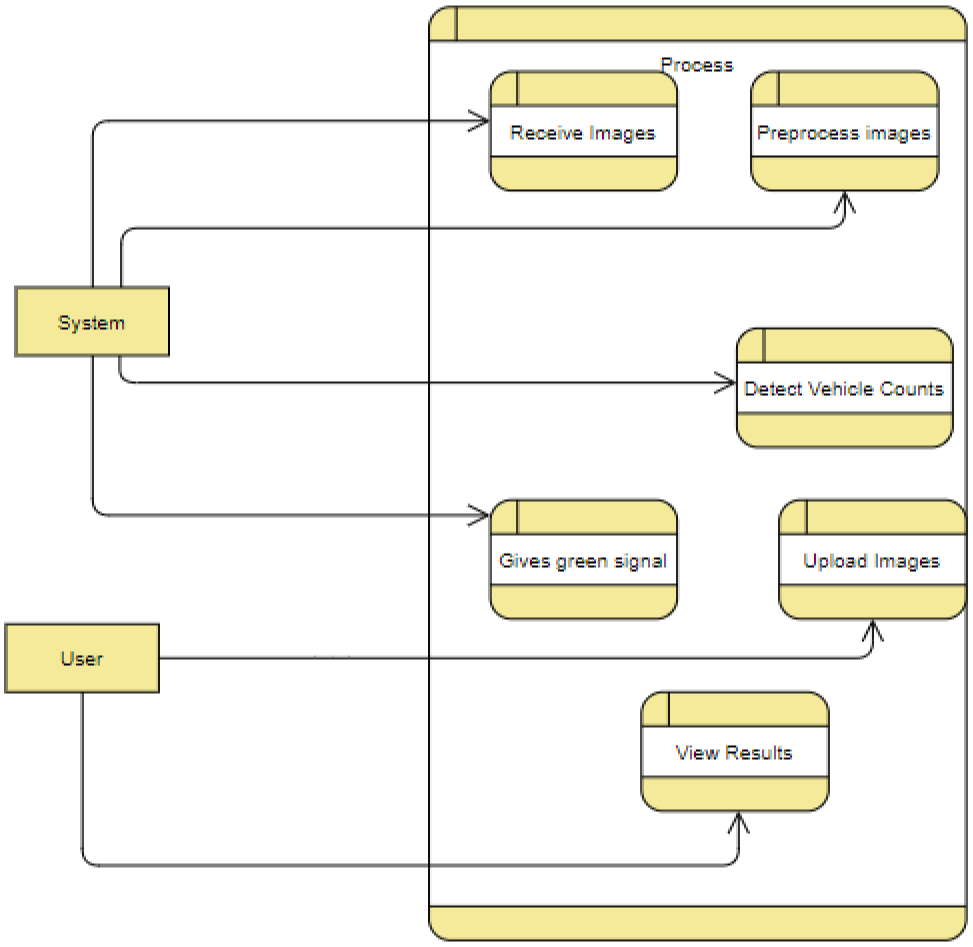
An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database. Let's have a look at a simple ER diagram to understand this concept.



6.2.9 DFD DIAGRAM:

A Data Flow Diagram (DFD) is a traditional way to visualize the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both. It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.


System
Receive Images
Preprocess images
Detect Vehicle Counts
Gives green signal
User
Upload Images
View Results


Process
Receive Images
Preprocess images
System
Detect Vehicle Counts
Gives green signal
Upload Images
User
View Results

CHAPTER-7

IMPLEMENTATION

IMPLEMENTATION

7.1 METHODOLOGY

In this first part of the traffic management system, the inputs will be accepted in separate four individual videos of each lane in focus from four-way traffic junction. These videos might come in any resolution or color format. The next phase is to apply object detection to the received frames from the previous stage. YOLOv4(YOU ONLY LOOK ONCE) is one of the many real-time object detection algorithms and it is a subtype of CNN (Convolutional Neural Network) which can identify a set of different classes of objects in a given image or a video. Once the object detection phase is complete, there will be a set of results from detecting each lane containing the number of vehicles detection in the frame (bounding boxes). YOLO plays an crucial role. It is single-stage processing architecture and used for real-time object detection of images or videos and provides an dynamic output.

7.2 MODULES

7.2.1 Receive Images:

The system will receive images from user. This module is responsible for receiving images from the user, who may provide them through various means such as uploading from a device or capturing them using a camera. The images serve as input data for the traffic management system. Once received, the images are passed on to the pre-processing module for further analysis.

7.2.2 Pre-processing:

The system will perform pre-processing on the uploaded images. The pre-processing module performs essential operations on the uploaded images to prepare them for subsequent analysis. This typically includes resizing the images to a standard size, converting them to grayscale, normalizing pixel values, and removing any noise or artifacts. These pre-processing steps ensure that the images are of consistent quality and format, which facilitates accurate detection and counting of vehicles in the next module.

7.2.3 Detects Vehicles count:

The system will detect the vehicle counts. Using deep learning algorithms, this module analyzes the pre-processed images to detect and count the number of vehicles present in each image. Techniques such as object detection and tracking are employed to identify vehicles and estimate their numbers accurately. The output of this module is the count of vehicles detected in each image.

7.2.4 Gives green signal:

The system then clears the traffic by giving green signal. Based on the vehicle count obtained from the previous module, this module determines which lane has the highest vehicle count and subsequently clears the traffic in that lane by giving a green signal. This decision is crucial for optimizing traffic flow and reducing congestion. By dynamically adjusting traffic signals based on real-time vehicle counts, the system helps manage traffic more efficiently

7.2.5 User Interface:

The user interface provides an interface for the users to interact with the system. This includes functionalities such as uploading images, viewing results, and possibly providing feedback or input parameters. Users can upload multiple images (image-1, image-2, etc.) for analysis, and the system processes them sequentially to make informed decisions about traffic signal control.

7.2.6 Upload Images:

This submodule within the user interface module allows users to upload images one by one for analysis. Users can select images from their devices and upload them to the system for processing. Each uploaded image undergoes pre-processing and vehicle count detection before the system determines the appropriate traffic signal control action.

- **a.** Upload image-1
- **b.** Upload image-2
- **c.** Upload image-3
- **d.** Upload image-4

7.2.7 View Results:

Clear the lane which has highest vehicle count by simply giving green signal. Once the analysis is complete, this submodule displays the results to the user. It highlights the lane with the highest vehicle count and indicates that the traffic signal in that lane will be set to green to clear the congestion. Users can review the results and observe how the system dynamically manages traffic flow based on real-time data.

7.3 ALGORITHMS

7.3.1 CNN:

In deep learning, a convolutional neural network (CNN, or ConvNet) is a class of deep neural networks, most commonly applied to analysing visual imagery. They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on the shared-weight architecture of the convolution kernels that shift over input features and provide translation equivariant responses.

CNNs are regularized versions of multilayer perceptrons. Multilayer perceptrons usually mean fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer. The "full connectivity" of these networks makes them prone to overfitting data. Typical ways of regularization, or preventing overfitting, include: penalizing parameters during training (such as weight decay) or trimming connectivity (skipped connections, dropout, etc.) CNNs take a different approach towards regularization: they take advantage of the hierarchical pattern in data and assemble patterns of increasing complexity using smaller and simpler patterns embossed in their filters. Therefore, on a scale of connectivity and complexity, CNNs are on the lower extreme.

The name "convolutional neural network" indicates that the network employs a mathematical operation called convolution. Convolutional networks are a specialized type of neural networks that use convolution in place of general matrix multiplication in at least one of their layers.

APPLICATIONS:

- Image recognition
- Video analysis
- Natural language processing
- Anomaly Detection
- Drug discovery
- Health risk assessment and biomarkers of aging discovery

7.3.2 YOLO:

Yolo is a part of object detection, Object detection is a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class (such as humans, buildings, or cars) in digital images and videos. Well-researched domains of object detection include face detection and pedestrian detection. Object detection has applications in many areas of computer vision, including image retrieval and video surveillance.

Every object class has its own special features that helps in classifying the class – for example all circles are round. Object class detection uses these special features. For example, when looking for circles, objects that are at a particular distance from a point (i.e., the center) are sought. Similarly, when looking for squares, objects that are perpendicular at corners and have equal side lengths are needed. A similar approach is used for face identification where eyes, nose, and lips can be found and features like skin color and distance between eyes can be found.

APPLICATIONS:

It is widely used in computer vision tasks such as image annotation, activity recognition, face detection, face recognition, video object co-segmentation. It is also used in tracking objects, for example tracking a ball during a football match, tracking movement of a cricket bat, or tracking a person in

.

SOURCE CODE:

```
#opencv-python==4.9.0.80
from re import DEBUG
from flask import Flask,render_template,session,redirect,url_for,request
import os
import cv2
import matplotlib
matplotlib.use("Agg")
import matplotlib.pyplot as plt
import numpy as np
import cvlib as cv
from cvlib.object_detection import draw_bbox
import operator
app=Flask(__name__)
req_classes = ['bicycle', 'car', 'motorcycle', 'bus', 'truck']
@app.route('/')
def index():
    return render_template("index.html")
vehicles_count={}
vehicles_count_IMAGE={}

# Load class labels
```

```
with open('coco.names', 'r') as f:
    classes = [line.strip() for line in f.readlines()]
# Initialization
class_ids = []
confidences = []
boxes = []
vehicle_classes = ['car', 'motorbike', 'bus', 'truck']
@app.route("/upload1",methods=["POST","GET"])
def upload1():
    try:
      if request.method=="POST":
        myfile = request.files['file']
         global fn1
           fn1 = myfile.filename
            mypath = os.path.join('static/img/team/', fn1)
            global image1
            image1=mypath
            myfile.save(mypath)
            img = cv2.imread(mypath)
            net = cv2.dnn.readNet('yolov3.weights', 'yolov3.cfg')
            layer_names = net.getLayerNames()
       output_layers = [layer_names[i - 1] for i in net.getUnconnectedOutLayers().flatten()
            # Load image
            # img = cv2.imread('4.jpg')
            if img is None:
                raise FileNotFoundError("The image file was not found or could not be opened. Please check the file path.")
            height, width = img.shape[:2]
            # Preprocess image and run forward pass
            blob = cv2.dnn.blobFromImage(img, 0.00392, (416, 416), (0, 0, 0), True, crop=False)
            net.setInput(blob)
            outs = net.forward(output_layers)
            # Load class labels
            with open('coco.names', 'r') as f:
                classes = [line.strip() for line in f.readlines()]
            # Initialization
            class_ids = []
            confidences = []
            boxes = []
            vehicle_classes = ['car', 'motorbike', 'bus', 'truck']
            # Process detections
            for out in outs:
                for detection in out:
                    scores = detection[5:]
                    class_id = np.argmax(scores)
```

```
                    confidence = scores[class_id]

                    if confidence > 0.5 and classes[class_id] in vehicle_classes:

                        center_x = int(detection[0] * width)

                        center_y = int(detection[1] * height)

                        w = int(detection[2] * width)

                        h = int(detection[3] * height)

                        x = int(center_x - w / 2)

                        y = int(center_y - h / 2)

                        boxes.append([x, y, w, h])

                        confidences.append(float(confidence))

                        class_ids.append(class_id)

            # Applying Non-Max Suppression

            indices = cv2.dnn.NMSBoxes(boxes, confidences, score_threshold=0.5,
nms_threshold=0.4)


            # Ensure indices is a list (for compatibility with different OpenCV versions)

            if len(indices) and isinstance(indices[0], tuple):

                indices = [i[0] for i in indices]

            elif len(indices) and isinstance(indices, np.ndarray):

                indices = indices.flatten()

            # Draw bounding boxes and labels on the image

           for i in indices:

                box = boxes[i]
                x, y, w, h = box[0], box[1], box[2], box[3]
                cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)
                cv2.putText(img, f'{classes[class_ids[i]]}: {round(confidences[i] * 100, 2)}%',
(x, y - 5), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)
            global count1
            vehicle_count = len(indices)
            print(f'Vehicles detected: {vehicle_count}')
            count1=vehicle_count
            cv2.imshow('Image', img)
            cv2.waitKey(0)
            cv2.destroyAllWindows()
            return redirect(url_for('upload2'))
    except:
        return render_template("uploadimages.html",msg="fail")
    return render_template("uploadimages.html")
# @app.route("/upload1",methods=["POST","GET"])
# def upload1():
#     try:
#         if request.method=="POST":
#             myfile = request.files['file']
#             global fn1
#            fn1 = myfile.filename mypath = os.path.join('static/img/team/', fn1)
#             global image1
#            image1=mypath
#            myfile.save(mypath)
#            im = cv2.imread(mypath)
#            bbox, label, conf = cv.detect_common_objects(im, model='yolov3.weights')
#            output_image = draw_bbox(im, bbox, label, conf)
#            plt.imshow(output_image)
#            plt.savefig(mypath)
#            count = [label.count(c) for c in req_classes]
#            global count1
#            count1 = sum(count)
#            print('The first uploaded image count is:', count1)
#            vehicles_count["1st Image"]=count1
          print(count3)
          cv2.imshow('Image', img)
          cv2.waitKey(0)
          cv2.destroyAllWindows()
        return render_template("success1.html", mh="hi")
    except:
        return render_template("upload3y.html",msg="fail")

    return render_template("upload3y.html")if (__name__)==('__main__'):
    app.run(debug=True)
```

CHAPTER-8

SYSTEM STUDY & TESTING

8.SYSTEM STUDY AND TESTING

8.1Feasibility Study

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

Three key considerations involved in the feasibility analysis are

- Economical feasibility
- Technical feasibility
- Social feasibility

Economical Feasibility

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

Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

Social Feasibility

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

8.2 System Testing

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

8.2.1 Types of Tests

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

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

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Functional testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input	: identified classes of valid input must be accepted.
Invalid Input	: identified classes of invalid input must be rejected.
Functions	: identified functions must be exercised.
Output	: identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

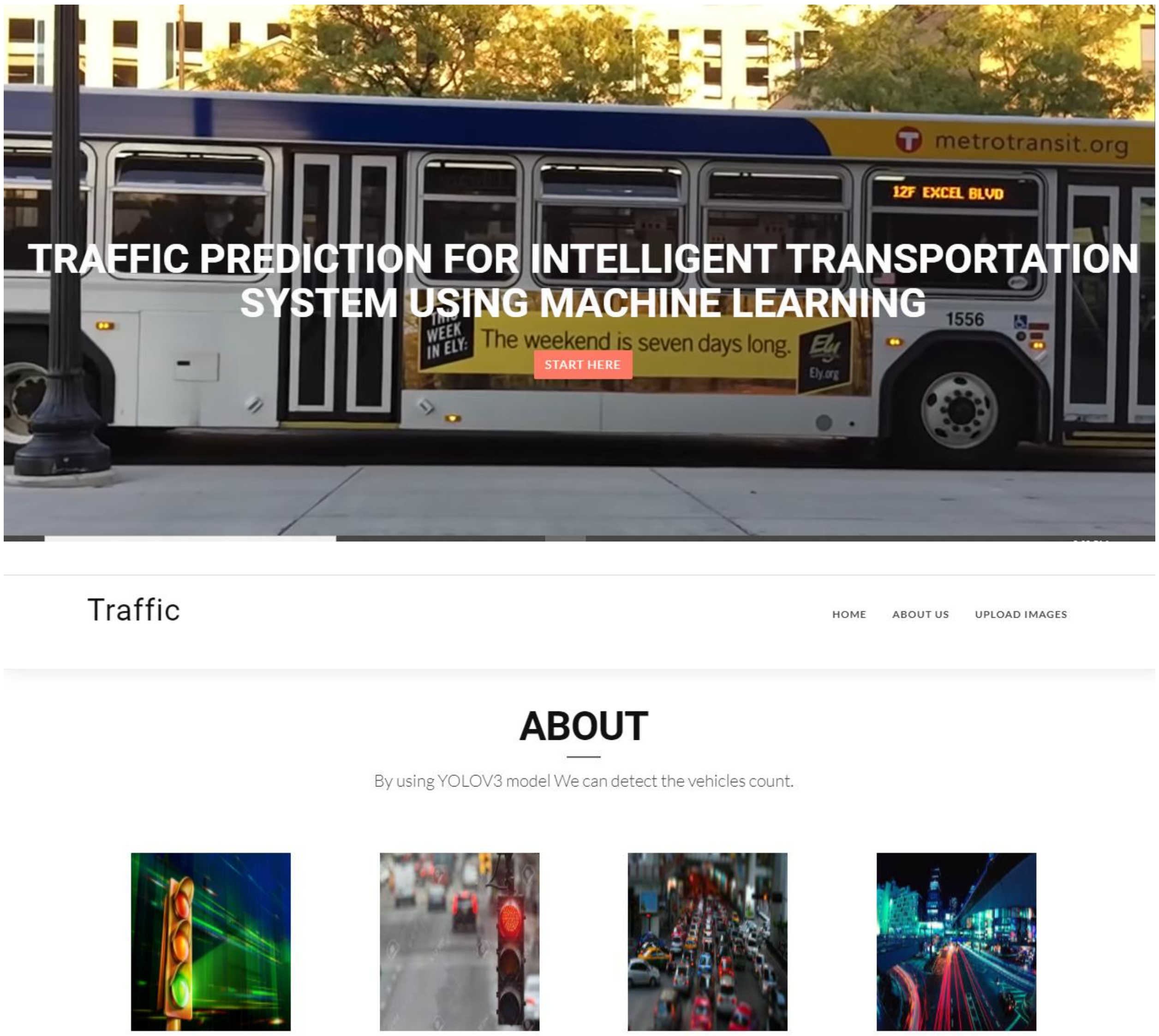
Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

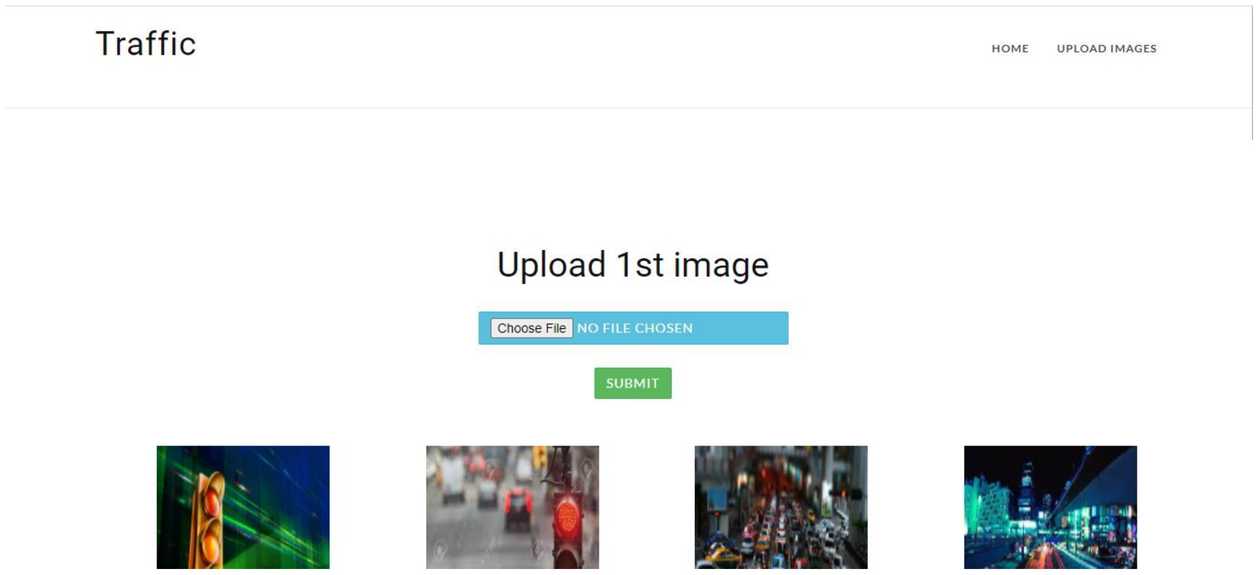
CHAPTER-10

EXPERTMENTAL OUTPUT

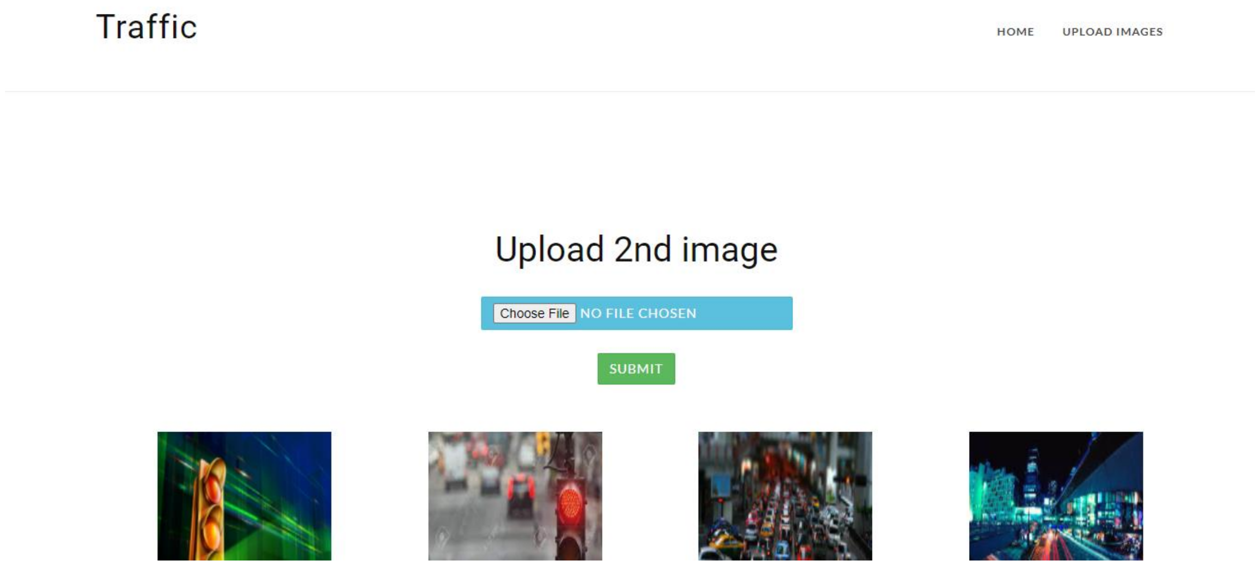
Home Page:


TRAFFIC PREDICTION FOR INTELLIGENT TRANSPORTATION SYSTEM USING MACHINE LEARNING
START HERE
Traffic
HOME
ABOUT US
UPLOAD IMAGES
ABOUT
By using YOLOV3 model We can detect the vehicles count.

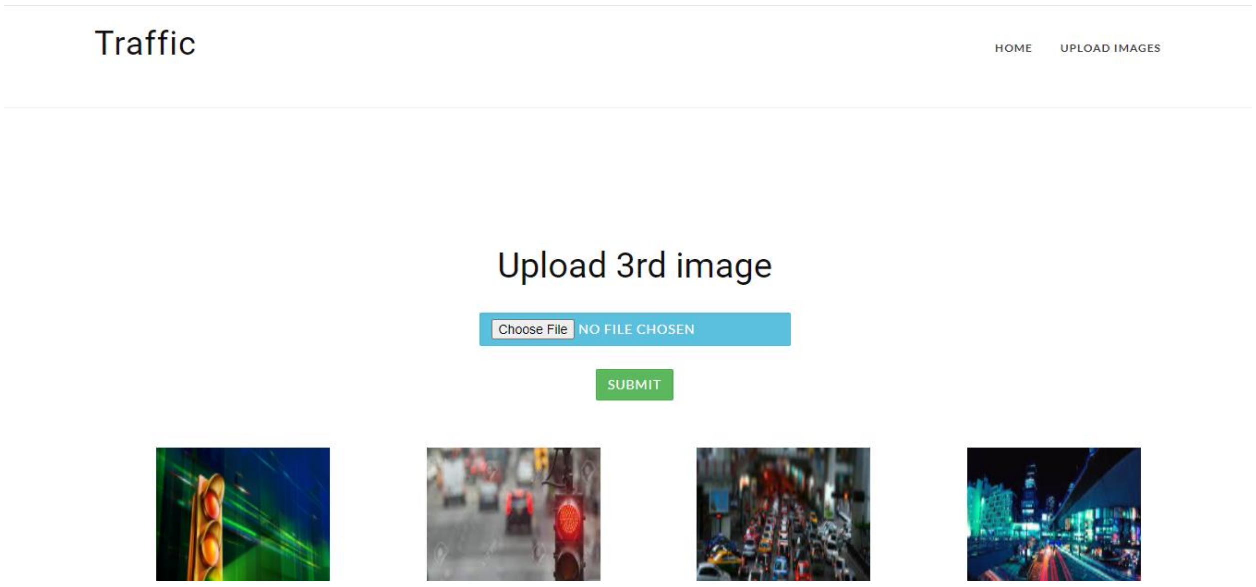
Upload 1st image:



Upload 2nd image:



Upload 3rd image:

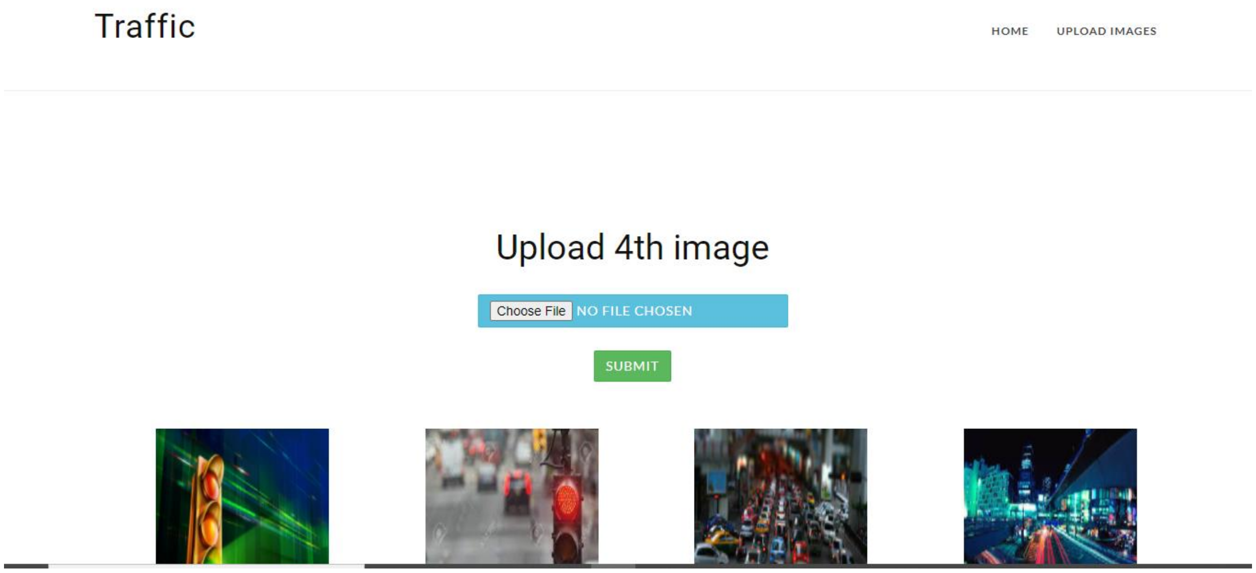


Traffic

HOME UPLOAD IMAGES

Upload 3rd image

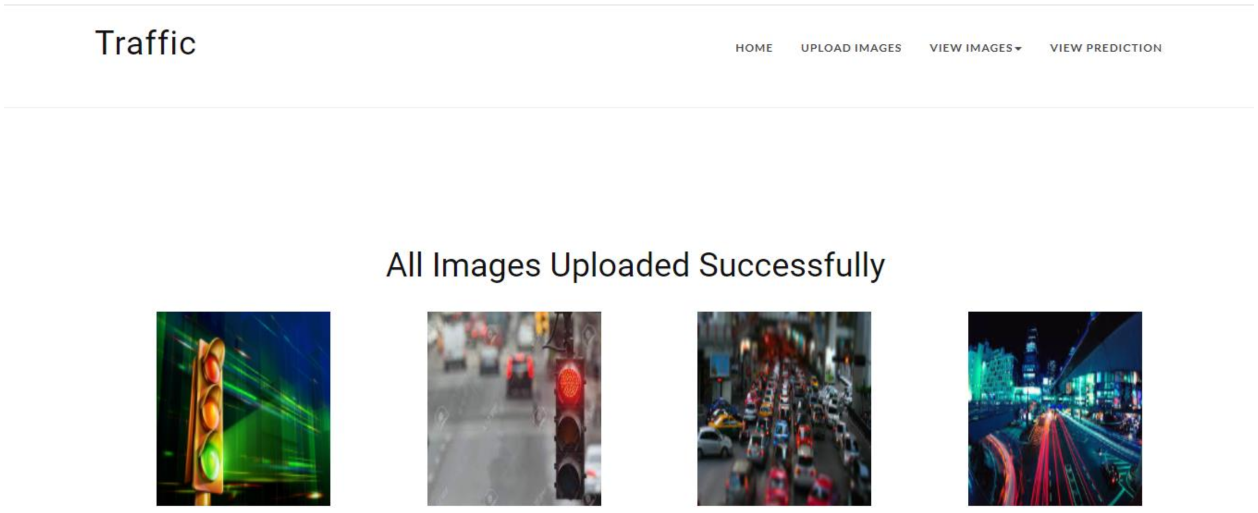
Upload 4th image:



Traffic

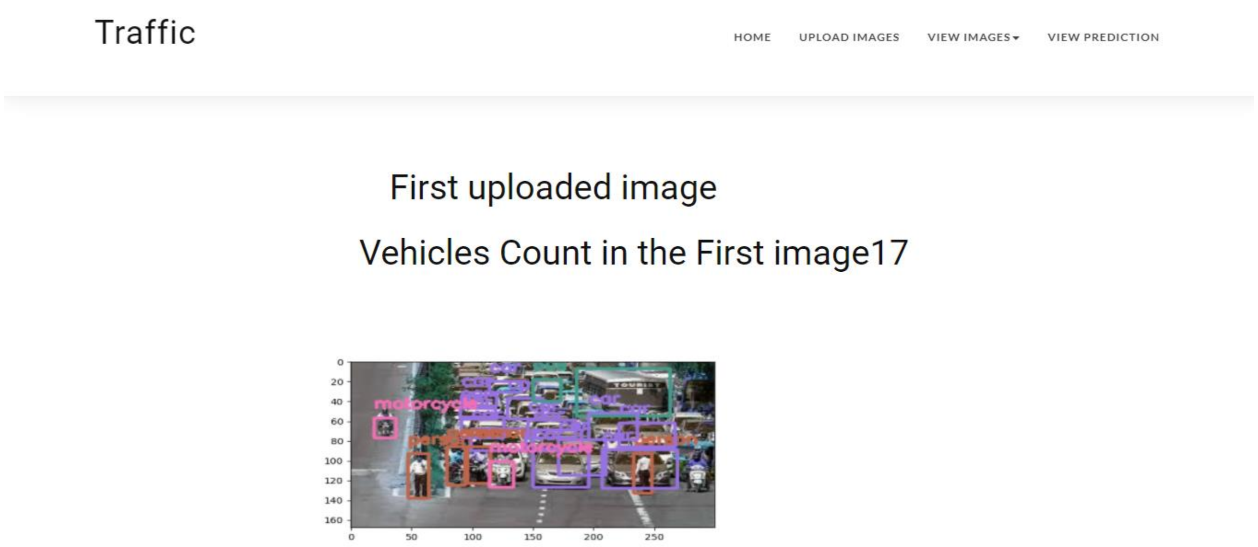
HOME UPLOAD IMAGES

Upload 4th image



All Images Uploaded Successfully

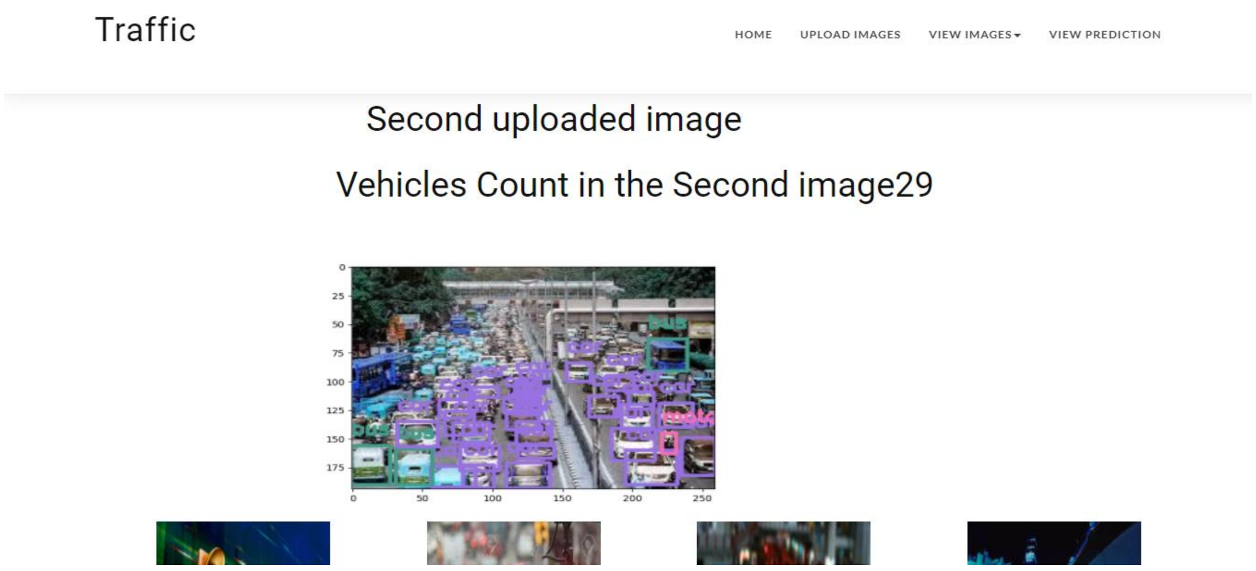
View First uploaded image and count:



First uploaded image

Vehicles Count in the First image17

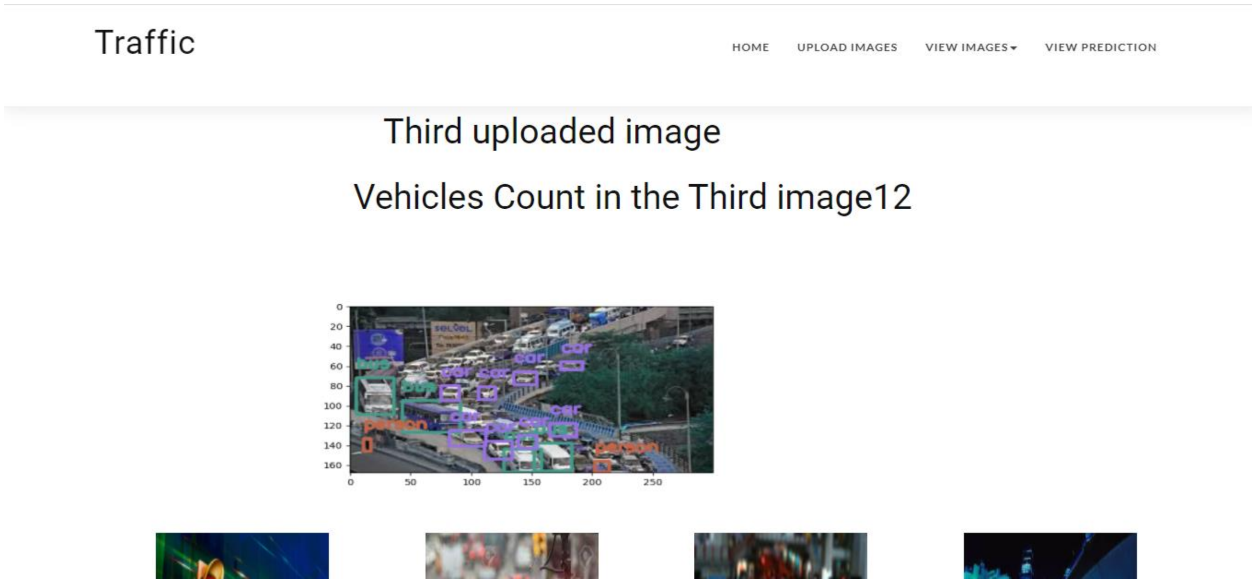
View 2nd uploaded images and count:



Second uploaded image

Vehicles Count in the Second image29

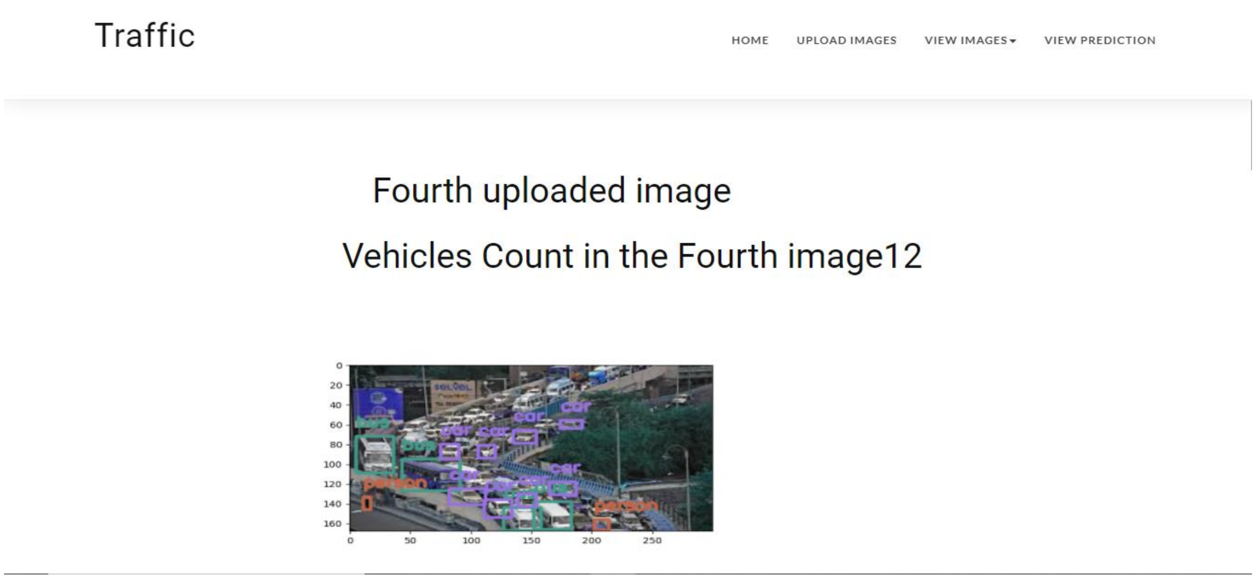
View 3rd uploaded image:



Third uploaded image

Vehicles Count in the Third image12

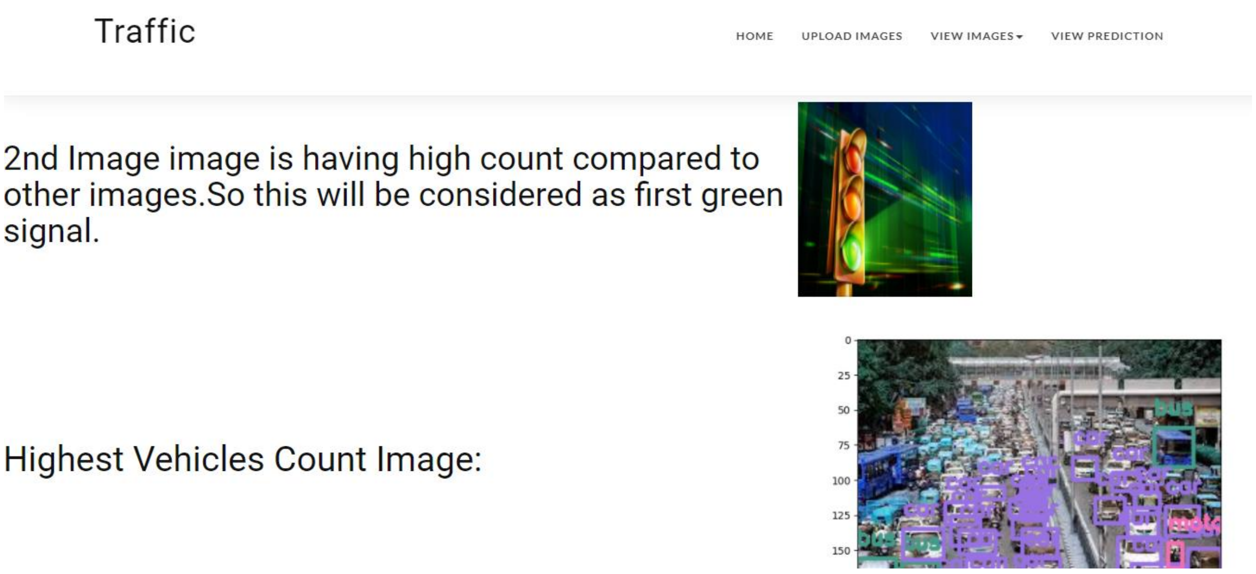
View 4th uploaded image and counts:



Fourth uploaded image

Vehicles Count in the Fourth image12

View Prediction:



2nd Image image is having high count compared to other images.So this will be considered as first green signal.

Highest Vehicles Count Image:

CHAPTER-11

CONCLUSION&FUTURE ENHANCEMENT

11. CONCLUSION& FUTURE ENHANCEMENT

In this application, we have successfully created a system that controls traffic signals manually. This is developed in a user-friendly environment using Flask via Python programming. The system is likely to collect images from the user to clear signals for the lanes which has highest count of vehicles. Future enhancements in traffic prediction for intelligent transportation systems using deep learning may involve integrating multi-modal data sources such as weather, event schedules, and urban development plans to create more comprehensive predictive models. Implementing reinforcement learning techniques can enhance adaptability to dynamic traffic conditions, enabling the system to learn and optimize traffic signal control in real-time. Additionally, exploring the fusion of deep learning with edge computing could facilitate quicker decision-making at the local level, improving responsiveness and overall traffic flow management for smarter and more efficient transportation networks.

CHAPTER-12

BIBILOGRAPHY

12. BIBILOGRAPHY

1. Rutger Claes, Tom Holvoet, and Danny Weyns. A decentralized approach for anticipatory vehicle routing using delegate multiagent systems. IEEE Transactions on Intelligent Transportation Systems, 12(2):364–373, 2011.
2. Mehul Mahrishi and Sudha Morwal. Index point detection and semantic indexing of videos - a comparative review. Advances in Intelligent Systems and Computing, Springer, 2020.
3. C. Zhang, P. Patras, and H. Haddadi. Deep learning in mobile and wireless networking: A survey. IEEE Communications Surveys Tutorials, 21(3):2224–2287, third quarter 2019.
4. Chun-Hsin Wu, Jan-Ming Ho, and D. T. Lee. Travel-time prediction with support vector regression. IEEE Transactions on Intelligent Transportation Systems, 5(4):276–281, Dec 2004.
5. Yang, C., Zhang, Y., & Lu, C. (2020). Traffic Sign Detection and Recognition Using YOLOv4 in the Wild. Electronics, 9(2), 276.
6. Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Advances in Neural Information Processing Systems (NIPS).
7. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. In European Conference on Computer Vision (ECCV).
8. Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J. (2018). Path Aggregation Network for Instance Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
9. Cheng, Z., Qian, C., & Zhu, C. (2018). Multimodal Traffic Flow Prediction: A Deep Learning Approach. IEEE Transactions on Intelligent Transportation Systems, 19(11), 3577-3587.
10. Luo, Y., Tian, Y., Li, X., & Tang, Y. (2019). Vehicle Detection from 3D Lidar Using Fully Convolutional Network. Sensors, 19(23), 5226.
11. Ji, S., & Ye, Q. (2018). Multi-Class Traffic Sign Detection and Recognition for Autonomous Driving. Sensors, 18(4), 1338.
12. Wu, D., Zhang, F., Yang, S., & Du, Y. (2018). Deep Learning-Based Traffic Sign Detection and Recognition. Sensors, 18(8), 2665.
13. Wang, Y., Yu, X., Huang, S., & Sun, J. (2020). Large-Scale Traffic Sign Detection and Recognition in the Wild. Sensors, 20(11), 3121.
14. Chen, W., Li, H., & Su, H. (2018). Traffic Sign Detection and Recognition Using Modified YOLOv2. In Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO).