

**Professor**  **DECLARATION**

We hereby declare that the Innovative Product Development **ROAD TRAFFIC VECHICLE**

**DETECTION AND TRACKING** submitted to Malla Reddy Engineering College for Women affiliated to

Jawaharlal Nehru Technological University, Hyderabad (JNTUH) for the award of the Degree of Bachelor of

Technology in CSE-DS is a result of original research work done by us. It is further declared that the Innovative Product Development report or any part thereof has not been previously submitted to any University or Institute for the award of Degree.

Place: K. ASFIYA (21RH1A6730)

Date: B. SRAVANI (21RH1A6707)

G. HARINI (22RH5A6703)

S. AKSHAYA DEEPA (21RH1A6759)

# ACKNOWLEDGEMENT

We feel ourselves honored and privileged to place our warm salutation to our college **Malla Reddy Engineering College for Women and Department of Computer Science and Engineering** which gave us the opportunity to have expertise in engineering and profound technical knowledge.

We would like to deeply thank our Honorable Minister of Telangana State **Sri.Ch. Malla Reddy Garu**, founder chairman MRGI, the largest cluster of institutions in the state of Telangana for providing us with all the resources in the college to make our project success.

We wish to convey gratitude to our **Principal Dr. Y. Madhavee Latha**, for providing us with the environment and mean to enrich our skills and motivating us in our endeavor and helping us to realize our full potential.

We would like to thank **Prof. A. Radha Rani**, Director of Computer Science and Engineering & Information Technology for encouraging us to take up a project on this subject and motivating us towards the Project Work.

We express our sincere gratitude to **Dr.V. Pradeep**, Head of the Department of CSE –DS for inspiring us to take up a project on this subject and successfully guiding us towards its completion.

We would like to thank our **Ms. G. Padma**, Assistant Professor and all the Faculty members for their valuable guidance and encouragement towards the completion of our project work.

**With Regards and Gratitude**

**K. ASFIYA (21RH1A6730)**

**B. SRAVANI (21RH1A6707)**

**G. HARINI (22RH5A6703)**

**S. AKSHAYA (21RH1A6759)**

**Road Traffic Vehicle Detection and Tracking using Deep**

**Learning with Custom-Collected and Public Datasets**

# 1.INTRODUCTION

Artificial intelligence (AI) has taken us by storm, helping us to make decisions in everything we do, even in finding our “true love” and the “significant other”. Machine learning and deep learning are revolutionizing smart cities and societies and solving many longstanding problems in sectors such as health, education , and others. Transportation is the backbone of modern societies and economies albeit continuing to cause unbelievable damages of the likes of over a million deaths, 20-50 million injuries to people, and trillions of dollars, each year, around the globe. Traditional approaches of measuring and monitoring road traffic rely on inductive loops that give some basic information about average speed, vehicle occupancy, and traffic flow. The traditional technologies are unable to support real-time monitoring and management of road traffic. Deep learning has the potential to revolutionize many fields and transportation is not an exception. Among the deep learning methods for road transportation, there are methods that involve image and video analysis for automated traffic monitoring with applications in detecting traffic congestion, road safety, and many more. This paper presents a study on the use of off the shelf, highly optimized, deep learning models for vehicle detection and vehicle tracking on roads. While the earlier works have focused on the same, none of the works have applied these models to road traffic in Kingdom of Saudi Arabia (KSA). The vehicles, roads, traffic, traffic conditions, etc. in KSA are different from other countries in several respects such as the mix of traffic involving various vehicles manufacturer and models, the driving culture, road infrastructure, language, and more. We use YOLOv4 for object detection and DeepSORT for tracking the detected vehicles. We have used three different variations of the deep learning models and compared their performance; a pre-trained model with the COCO dataset, and two custom-trained models with different datasets. We have used three different datasets; the COCO dataset [8], the Berkeley DeepDrive dataset [9], and our customdeveloped dataset obtained by a Dash Cam installed onboard vehicle driven on city streets and highways in the Kingdom of Saudi Arabia (KSA). We have collected data in five different traffic conditions, city traffic in day and night, highway traffic in day and night, and traffic in rain. For experiments, we have used Google Colab platform to harness GPU power, CUDA and OpenCV. The results have been evaluated using precision and other metrics. The rest of the paper is organized as follows. Section 2 reviews the relevant works. Section 3 details our methodology, model design, and datasets. Section 4 presents results and analysis. Section 5 concludes the discussion.

# Objective of the Project

Deep learning is revolutionizing smart cities and societies, solving many longstanding problems. Transportation is continuing to cause unbelievable damages including 1.25 million deaths and trillions of dollars annually. This paper presents a study on the use of YOLOv4 for vehicle detection and DeepSORT for tracking the detected vehicles on roads. None of the earlier works have applied these models to road traffic in the Kingdom of Saudi Arabia (KSA). We have used three different variations of the deep learning models and compared their performance; a pretrained model with the COCO dataset, and two custom-trained models with the Berkeley DeepDrive dataset, and our custom-developed dataset obtained by a Dash Cam installed onboard vehicle driven on KSA roads in five different traffic conditions; city traffic in day and night, highway traffic in day and night, and traffic in rain. We have used Google Colab platform to harness GPU power, CUDA and OpenCV. The results have been evaluated using precision and other metrics.

# 2. LITERATURE SURVEY

## “Distributed Artificial Intelligence-as-a-Service (DAIaaS) for Smarter IoE and 6G Environments,”

Artificial intelligence (AI) has taken us by storm, helping us to make decisions in everything we do, even in finding our "true love" and the "significant other". While 5G promises us high-speed mobile internet, 6G pledges to support ubiquitous AI services through next-generation softwarization, heterogeneity, and configurability of networks. The work on 6G is in its infancy and requires the community to conceptualize and develop its design, implementation, deployment, and use cases. Towards this end, this paper proposes a framework for Distributed AI as a Service (DAIaaS) provisioning for Internet of Everything (IoE) and 6G environments. The AI service is "distributed" because the actual training and inference computations are divided into smaller, concurrent, computations suited to the level and capacity of resources available with cloud, fog, and edge layers. Multiple DAIaaS provisioning configurations for distributed training and inference are proposed to investigate the design choices and performance bottlenecks of DAIaaS. Specifically, we have developed three case studies (e.g., smart airport) with eight scenarios (e.g., federated learning) comprising nine applications and AI delivery models (smart surveillance, etc.) and 50 distinct sensor and software modules (e.g., object tracker). The evaluation of the case studies and the DAIaaS framework is reported in terms of end-to-end delay, network usage, energy consumption, and financial savings with recommendations to achieve higher performance. DAIaaS will facilitate standardization of distributed AI provisioning, allow developers to focus on the domain-specific details without worrying about distributed training and inference, and help systemize the mass-production of technologies for smarter environments.

**“Can Building ‘Artificially Intelligent Cities’ Safeguard Humanity from Natural Disasters,**

## Pandemics, and Other Catastrophes? An Urban Scholar’s Perspective,”

In recent years, artificial intelligence (AI) has started to manifest itself at an unprecedented pace. With highly sophisticated capabilities, AI has the potential to dramatically change our cities and societies. Despite its growing importance, the urban and social implications of AI are still an understudied area. In order to contribute to the ongoing efforts to address this research gap, this paper introduces the notion of an artificially intelligent city as the potential successor of the popular smart city brand-where the smartness of a city has come to be strongly associated with the use of viable technological solutions, including AI. The study explores whether building artificially intelligent cities can safeguard humanity from natural disasters, pandemics, and other catastrophes. All of the statements in this viewpoint are based on a thorough review of the current status of AI literature, research, developments, trends, and applications. This paper generates insights and identifies prospective research questions by charting the evolution of AI and the potential impacts of the systematic adoption of AI in cities and societies. The generated insights inform urban policymakers, managers, and planners on how to ensure the correct uptake of AI in our cities, and the identified critical questions offer scholars directions for prospective research and development.

## “UbeHealth: A personalized ubiquitous cloud and edgeenabled networked health

Smart city advancements are driving massive transformations of healthcare, the largest global industry. The drivers include increasing demands for ubiquitous, preventive, and personalized healthcare, to be provided to the public at reduced risks and costs. Mobile cloud computing could potentially meet the future healthcare demands by enabling anytime, anywhere capture and analyses of patients' data. However, network latency, bandwidth, and reliability are among the many challenges hindering the realization of next-generation healthcare. This paper proposes a ubiquitous healthcare framework, UbeHealth, that leverages edge computing, deep learning, big data, high-performance computing (HPC), and the Internet of Things (IoT) to address the aforementioned challenges. The framework enables an enhanced network quality of service using its three main components and four layers. Deep learning, big data, and HPC are used to predict network traffic, which in turn are used by the Cloudlet and network layers to optimize data rates, data caching, and routing decisions. Application protocols of the traffic flows are classified, enabling the network layer to meet applications' communication requirements better and to detect malicious traffic and anomalous data. Clustering is used to identify the different kinds of data originating from the same application protocols. A proof of concept UbeHealth system has been developed based on the framework. A detailed literature review is used to capture the design requirements for the proposed system. The system is described in detail including the algorithmic implementation of the three components and four layers. Three widely used data sets are used to evaluate the UbeHealth system.

## “UTiLearn: A Personalised Ubiquitous Teaching and Learning System for Smart Societies,”

The education industry around the globe is undergoing major transformations. Organisations such as Coursera are advancing new business models for education. A number of major industries have dropped degrees from the job requirements. While the economics of higher education institutions are under threat in a continuing gloomy global economy, digital and lifelong learners are increasingly demanding new teaching and learning paradigms from educational institutions. There is an urgent need to transform teaching and learning landscape in order to drive global economic growth. The use of distance eTeaching and eLearning (DTL) is on the rise among digital natives alongside our evolution towards smart societies. However, the DTL systems today lack the necessary sophistication due to several challenges including data analysis and management, learner-system interactivity, system cognition, resource planning, agility, and scalability. This paper proposes a personalised Ubiquitous eTeaching & eLearning (UTiLearn) Framework that leverages IoT, big data, supercomputing, and deep learning to provide enhanced development, management and delivery of teaching and learning in smart society settings. A proof of concept

UTiLearn system has been developed based on the framework. A detailed design, implementation and evaluation of the UTiLearn system, including its five components, is provided using eleven widely used datasets.

## “Extraction of Naturalistic Driving Patterns with Geographic Information Systems,”

A better understanding of Driving Patterns and their relationship with geographical driving areas could bring great benefits for smart cities, including the identification of good driving practices for saving fuel and reducing carbon emissions and accidents. The process of extracting driving patterns can be challenging due to issues such as the collection of valid data, clustering of population groups, and definition of similar behaviors. Naturalistic Driving methods provide a solution by allowing the collection of exhaustive datasets in quantitative and qualitative terms. However, exploiting and analyzing these datasets is complex and resource-intensive. Moreover, most of the previous studies, have constrained the great potential of naturalistic driving datasets to very specific situations, events, and/or road sections. In this paper, we propose a novel methodology for extracting driving patterns from naturalistic driving data, even from small population samples. We use Geographic Information Systems (GIS), so we can evaluate drivers' behavior and reactions to certain events or road sections, and compare across situations using different spatial scales. To that end, we analyze some kinematic parameters such as speeds, acceleration, braking, and other forces that define a driving attitude. Our method favors an adequate mapping of complete datasets enabling us to achieve a comprehensive perspective of driving performance.

## “A multiple inductive loop vehicle detection system for heterogeneous and lane-less traffic,”

This paper presents a novel inductive loop sensor that can detect vehicles under a heterogeneous and less-lane disciplined traffic and thus can be used to support a traffic control management system in optimizing the best use of existing roads. The loop sensor proposed in this paper detects large (e.g., bus) as well as small (e.g., bicycle) vehicles occupying any available space in the roadway, which is the main requirement for sensing heterogeneous and lane-less traffic. To accomplish the sensing of large as well as small vehicles, a multiple loop system with a new inductive loop sensor structure is proposed. The proposed sensor structure not only senses and segregates the vehicle type as bicycle, motor cycle, scooter, car, and bus but also enables accurate counting of the number of vehicles even in a mixed traffic flow condition. A prototype of the multiple loop sensing system has been developed and tested. Field tests indicate that the prototype successfully detected all types of vehicles and counted, correctly, the number of each type of vehicles. Thus, the suitability of the proposed sensor system for any type of traffic has been established.

## “Tracking Heavy Vehicles Based on Weigh-In-Motion and Inductive Loop Signature Technologies,”

Weigh-in-motion (WIM) has been employed as a major technology to collect heavy vehicles' data on the freeways. Because WIM is one of the most costly and sophisticated data collection systems, how to effectively utilize the valuable WIM data and monitor the performance of WIM stations are particularly important. In this paper, we proposed an innovative and yet practical approach for heavy vehicle tracking that combines the use of both WIM data and the inductive loop signature data. The proposed multilevel vehicle reidentification approach was able to generate promising tracking performance with both inductive loop signatures and WIM data applied.

## “Intelligent traffic light controller using inductive loops for vehicle detection,”

Through this paper we present the use of inductive loops as an instrument to measure traffic density. A microcontroller can be programmed to receive information about traffic density on different lanes, as measured by the inductive loops. Algorithms that not only ease congestion but also ensure the people in less congested lanes dont have to wait too long are discussed. Depending upon the traffic density a suitable algorithm can be executed to clear the congestion. A new design of inductive loop to suit our algorithm in case of multiple lane traffic has also been discussed here. Apart from causing delay, many times traffic congestion has resulted in loss of precious lives since help isnt able to reach the needy on time. In our proposed model we make use of radio transmitterreceiver to detect the presence of any ambulance/ fire brigade/ police vehicle and provide immediate right of way by traffic signal pre-emption. Lastly, there are many people who have a tendency of stopping way beyond the zebra crossing at a red signal. The use of infrared sensors to detect such vehicles and sound a buzzer to alert the traffic police has been presented. Overall, it is a complete model, one solution to many of traffic congestion related problems.

## “An implementation of automatic inductive-loop vehicle sensor using low-cost microcontroller,”

This paper presents a vehicle sensor circuit. This circuit is based on inductive-loop as a resonant circuit. The signal from this circuit is capture on a low-cost microcontroller with a low-speed build on ADC. This automatic calibration algorithm is based on k-mean clustering. The experimental results showed 100% accuracy detecting the subcompact car.

## “Spatiotemporal analysis of bluetooth data: Application to a large urban network,”

The emergence of new technologies allows better monitoring of traffic conditions and understanding of urban network dynamics. Bluetooth technology is becoming widespread, as it represents a cost-effective means for capturing road traffic in both arterials and motorways. Although the extraction of travel time from Bluetooth data is fairly straightforward, data reliability and processing is still challenging with the issues of penetration rate, mode discrimination, and detection quality. This paper presents a methodological contribution to the use of Bluetooth data for the spatiotemporal analysis of a large urban network (Brisbane, Australia). It introduces the concept of the Bluetooth origin–destination (B-OD) matrix, which is built from a network of 79 Bluetooth detectors located within the Brisbane urban area. The BOD matrix describes the dynamics of a subpopulation of vehicles, between pairs of detectors. The results show that the characteristics of urban networks can be effectively represented through B-OD matrices. A comparison with loop detector data enables an assessment of the results' significance. Then, the spatiotemporal structure of the network is analyzed with two different clustering analyses, namely, latent Dirichlet allocation (LDA) and $K$-means. While LDA is used to detect a temporal pattern, the $K$-means algorithm highlights Bluetooth fundamental diagram (BFD) classes. The results show that Bluetooth data has the potential to be a reliable data source for traffic monitoring. By highlighting hidden structures of a large area, the algorithm outputs allow us to provide the road operators with a fine spatiotemporal analysis of their network, in terms of traffic conditions.

## 3. SYSTEM ANALYSIS

**Existing system:**

Police can use image analysis to track down your vehicle, especially if it doesn’t have GPS. A few of the ways image analysis can help include: They can match your license plate or car image at toll booths and intersection cameras Use google street view and compare it to the image of your vehicle

**Disadvantages:**

* Takes more time

**Proposed system**:

This algorithm is popular because of its speed and accuracy. It has been used in various applications to detect traffic signals, people, parking meters, and animals. This article introduces readers to the YOLO algorithm for object detection and explains how it works.

**Advantages:**

* Faster Speed: YOLO algorithms works comparatively faster as compared to another algorithm.
* It is a highly generalized network: Probably YOLO is a highly generalized network because of its algorithm and the way it is trained.
* It processes each frame at the rate of 45 fps which is a larger network to 150 fps which is a smaller network, which is better than real-time.

## MODULES

To implement this project we have designed following modules

1. Generate & Load YOLOv4-DeepSort Model: using this module we will generate and load YOLOV$-DeepSort model
2. Upload Video & Detect Car & Truck: using this module we will upload test video and then apply YOLOV4 to detect vehicle and this detected vehicle frame will be further analyse by

DeepSort to track real vehicles

## 3.4. Software Requirement Specification

### 3.4.1. Overall Description

A Software Requirements Specification (SRS) – a [requirements specification f](http://en.wikipedia.org/wiki/Requirements_specification)or a [software system](http://en.wikipedia.org/wiki/Software_system) is a complete description of the behaviour of a system to be developed. It includes a set of [use cases t](http://en.wikipedia.org/wiki/Use_case)hat 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 a](http://en.wikipedia.org/wiki/Non-functional_requirements)re requirements which impose constraints on the design or implementation (such as [performance engineering r](http://en.wikipedia.org/wiki/Performance_engineering)equirements, [quality s](http://en.wikipedia.org/wiki/Quality_%28business%29)tandards, or design constraints).

System requirements specification: A structured collection of information that embodies the requirements of a system. A [business analyst,](http://en.wikipedia.org/wiki/Business_analyst) sometimes titled [system analyst,](http://en.wikipedia.org/wiki/System_analyst) is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Within the [systems development lifecycle d](http://en.wikipedia.org/wiki/Systems_development_life_cycle)omain, 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 d](http://en.wikipedia.org/wiki/Business_requirements)escribe 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 |  |  | - | intel i3(min) |
| • Speed |  |  | - | 1.1 Ghz |
| • RAM |  |  | - | 4 GB(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 **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

1. In this model the data and functionality are arrived from inside the system.
2. 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.

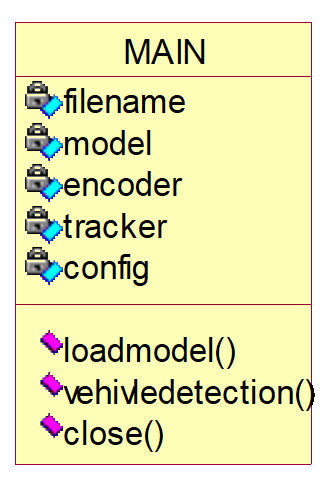
#### • Environmental Model View

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.



load model



vehicle detection



user



exit



**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 |  | load  model |  | vehicle  detection |  | exit |



load the model



successfully load model



detect the vehicle



successfully detect vehicle



exit

the model



successfully exit from model

**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.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  | | --- | | load  moed | |  | | |  | | --- | | detect  vehicle | |  | |

1:

2:

exit



User



successfully upload model



upload model



3:



detect



ve



hicle



4:



successfully detect vehicle



5:



successfully exit



6:



exit



**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.



user



load model



vehicle



detection



exit

**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.



user



load



model



vehicle



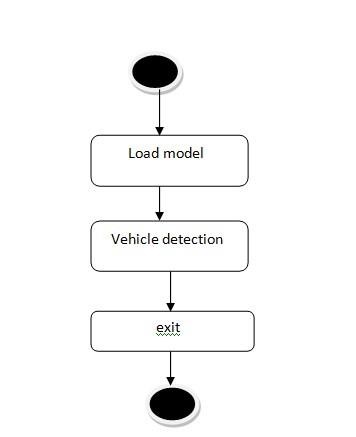
detectio



exit

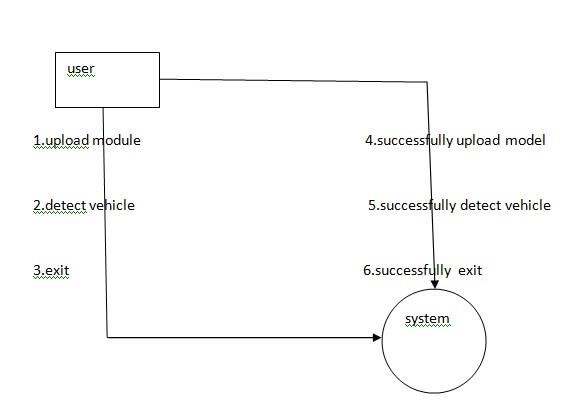
**Activity Diagram:**

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.



1. **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.

## 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.

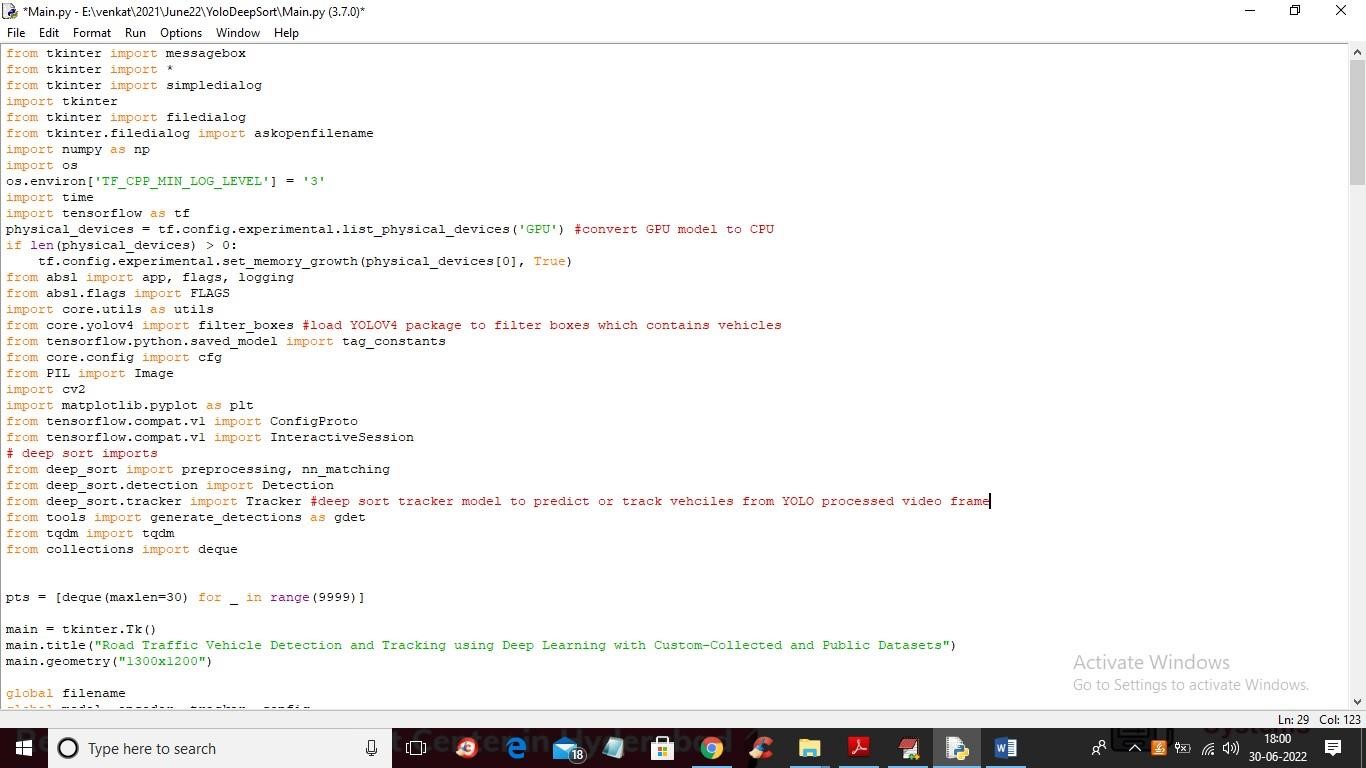
### Acceptance Testing

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**  **Priorit y** |
| **Step** | **Expected** | **Actual** |
| 01 | Load model | Test whether model is loded or not | If the  modelmay  not loaded | We cannot do further  operations | Model loaded we will do  further operations | High | High |
| 02 | Vehicle detection | Test whether  the vehicle  is  detecte  d or not | If the vehicle may not detected | we cannot  do loded Model | upLoad model we will do further | High | High |
| 03 | exit | Exit wether the vehicle is detected | Exit the model | After exit do another operations | Detect the another vehicle by same process | High | High |

**7. SCREENSHOTS**

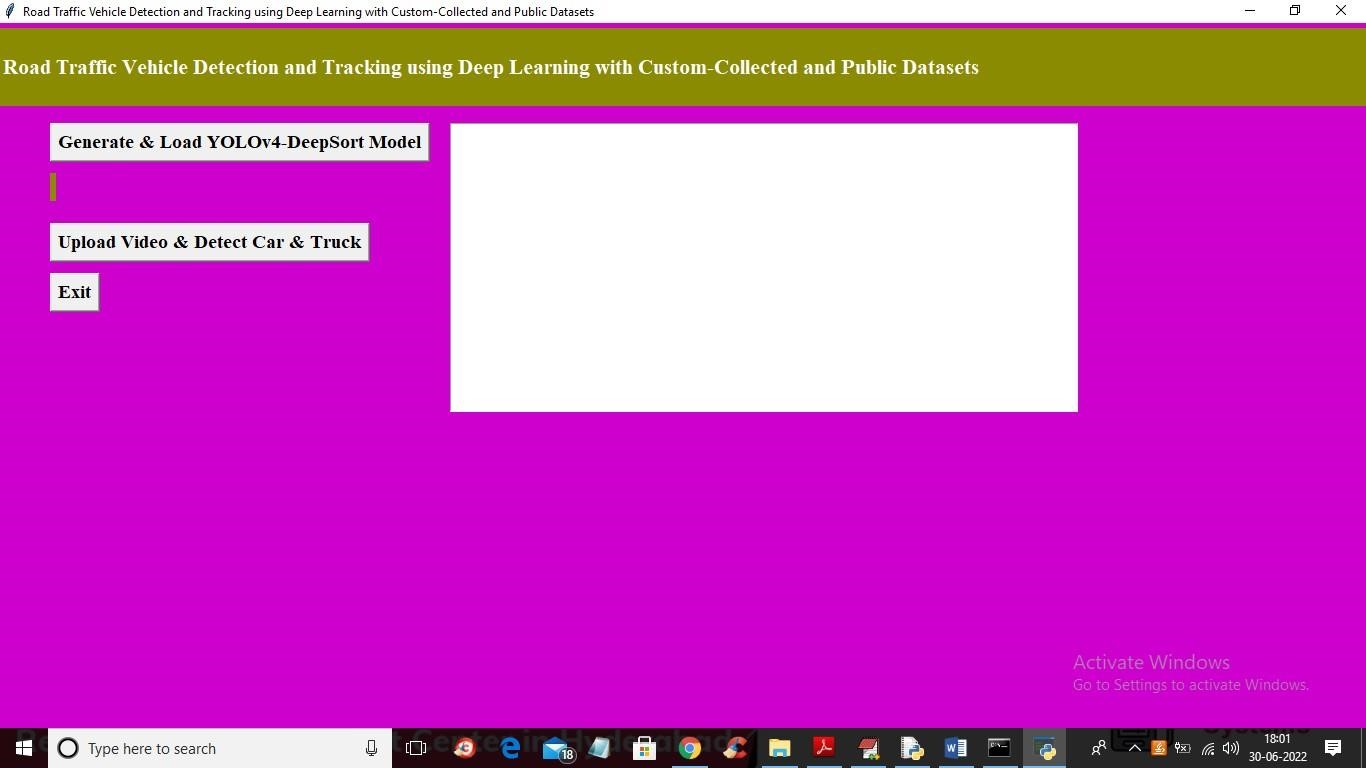
In below screen we are showing code to load YOLO and Deep sort models



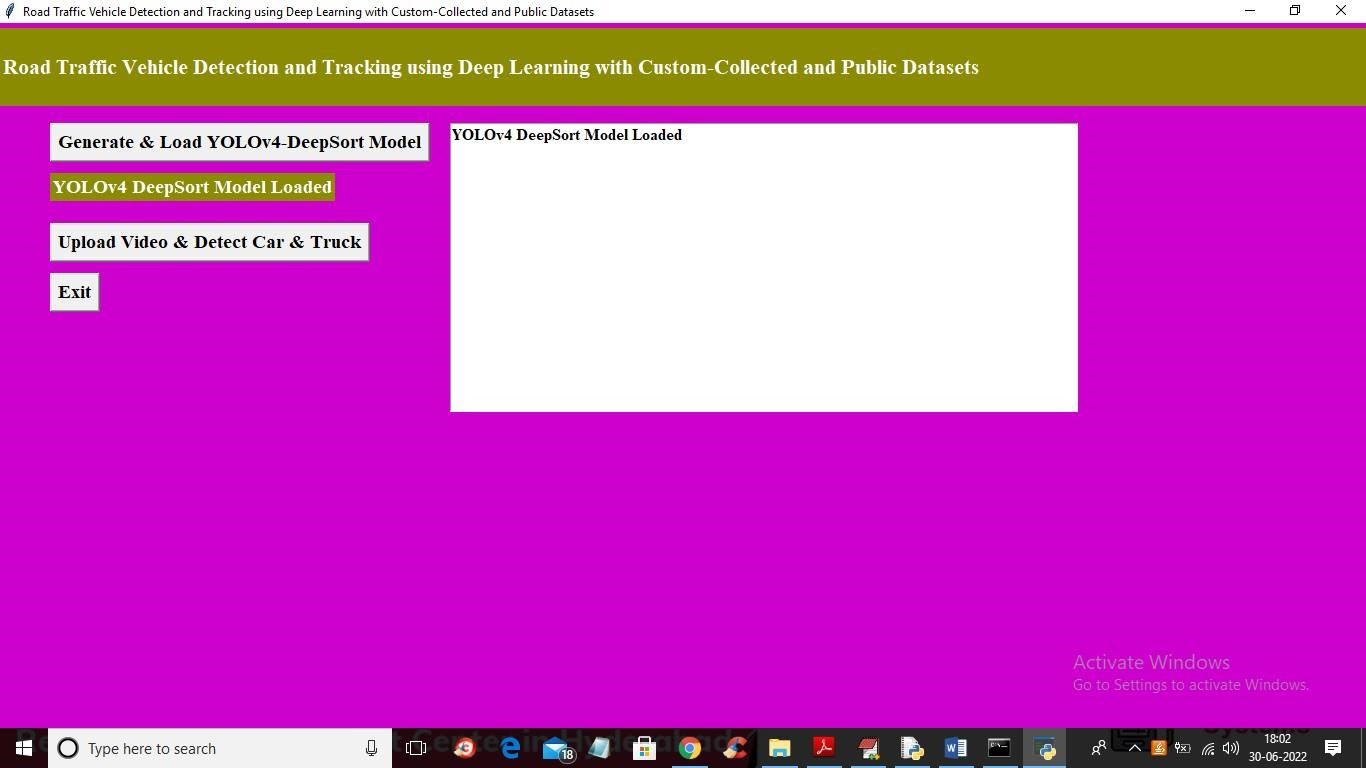
In above screen read red colour comments to know about YOLO and DeepSort packages and classes.

SCREEN SHOTS

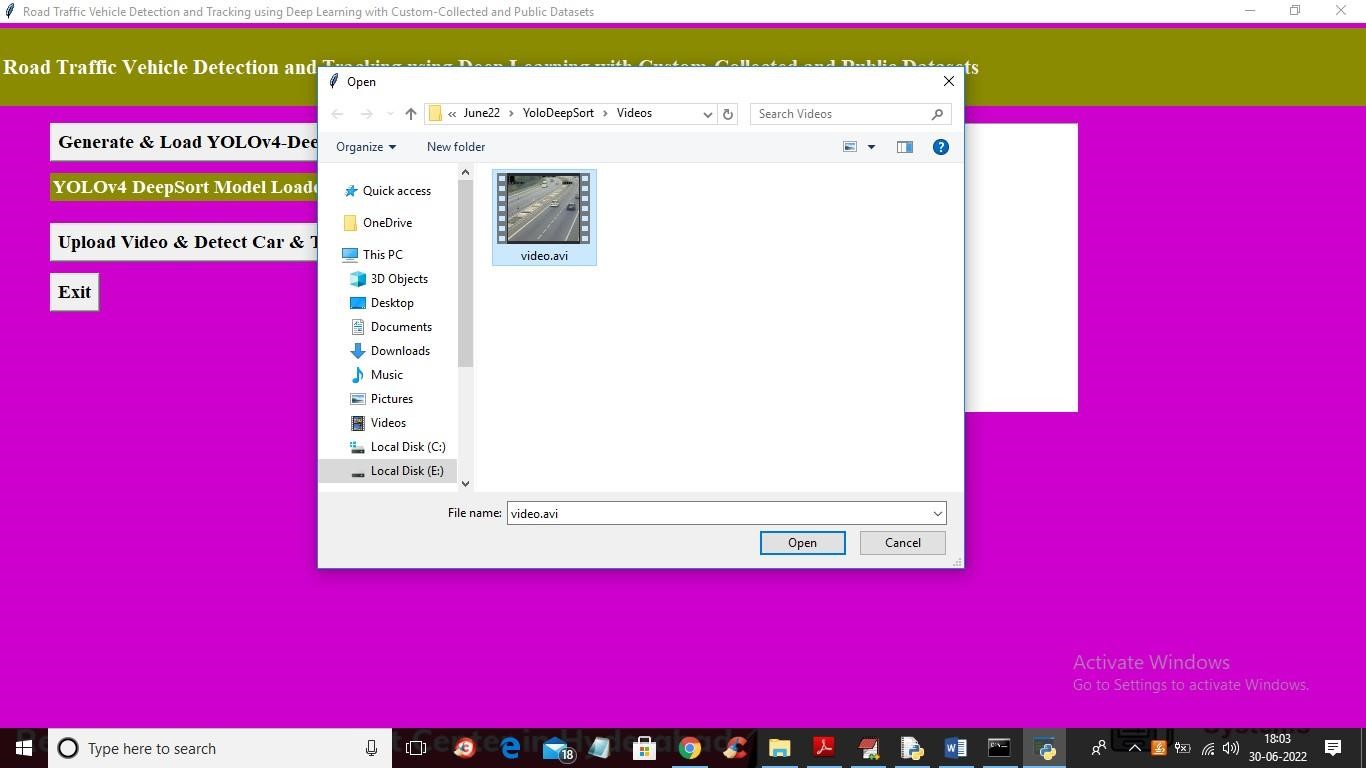
To run project double click on ‘run.bat’ file to get below screen



In above screen click on ‘Generate & Load YOLOv4-DeepSort Model’ button to load model and get below output



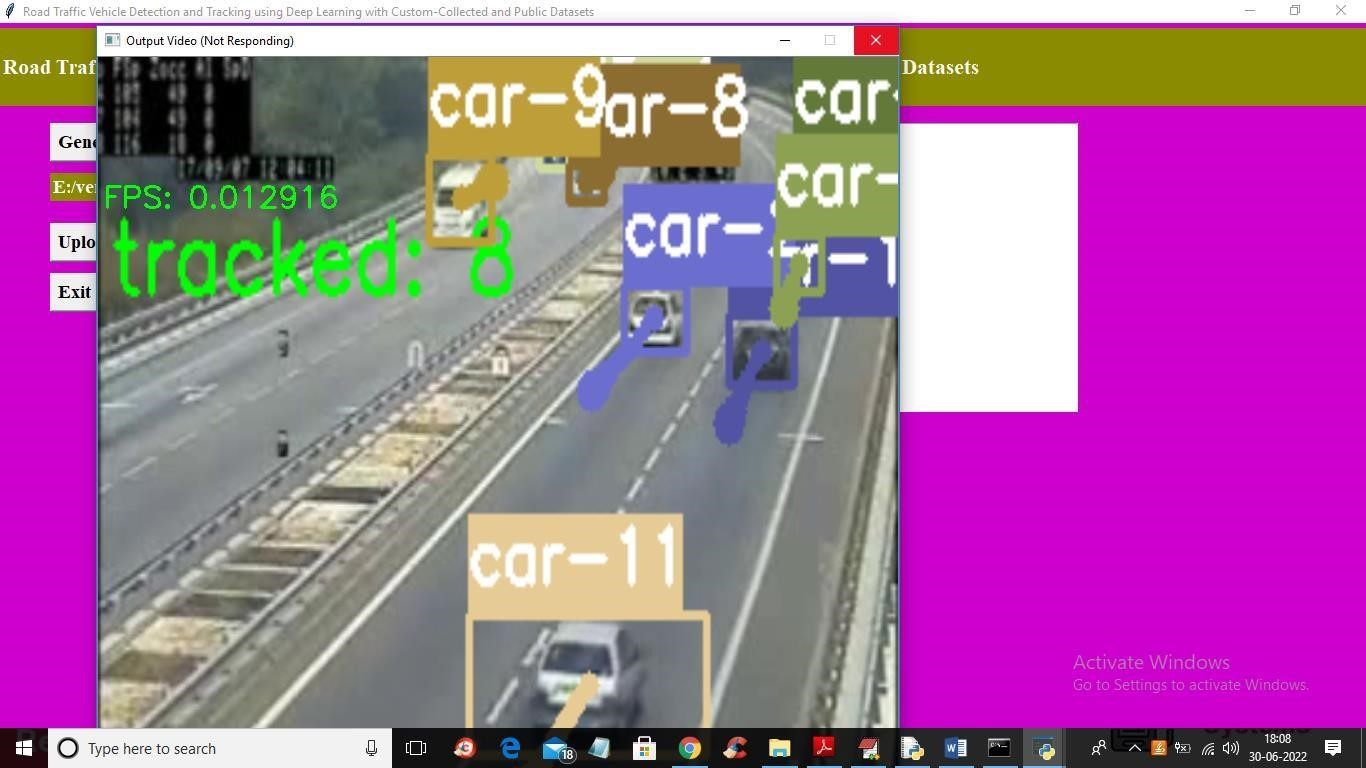
In above screen model is loaded and now click on ‘Upload Video & Detect Car & Truck’ button to upload video and start detecting and tracking vehicles



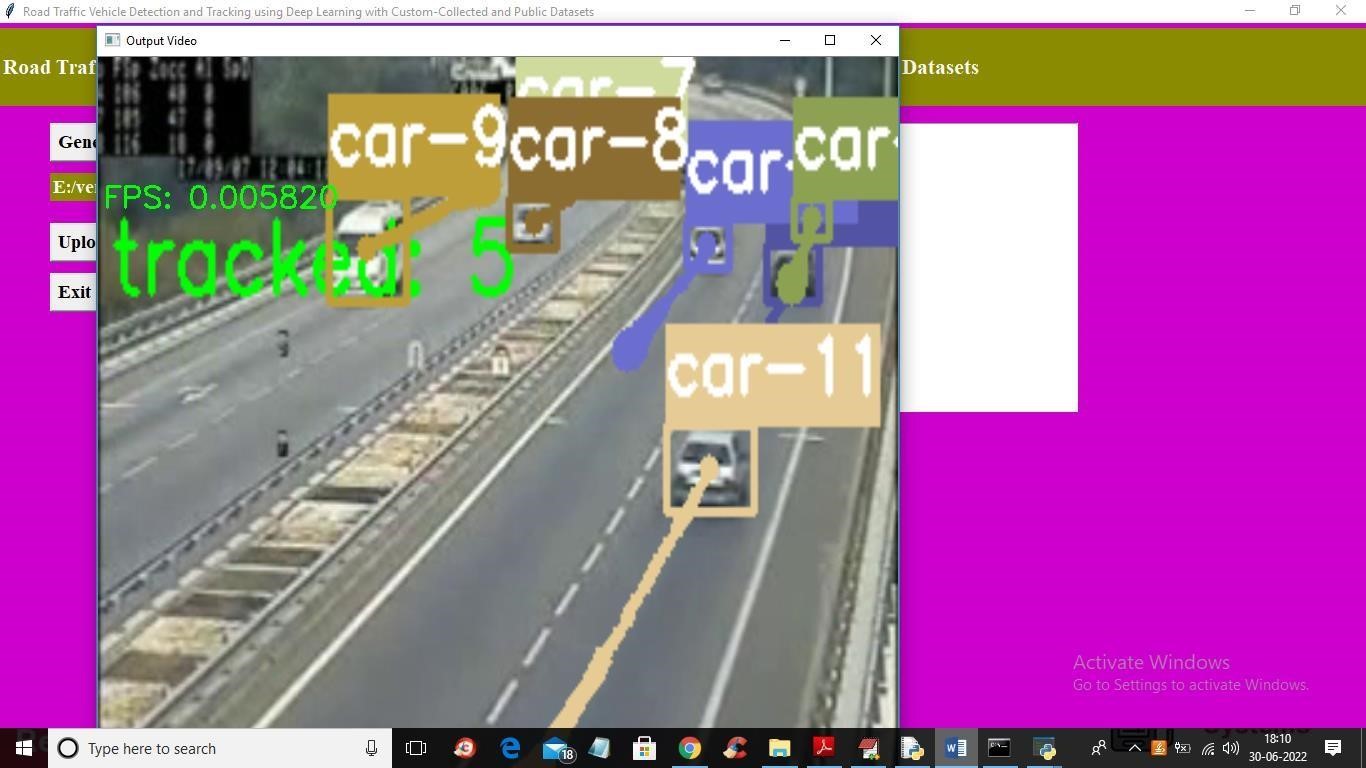
In above screen selecting and uploading ‘traffic video’ file and then click on ‘Open’ button to get below output. To get output u need to wait for few seconds



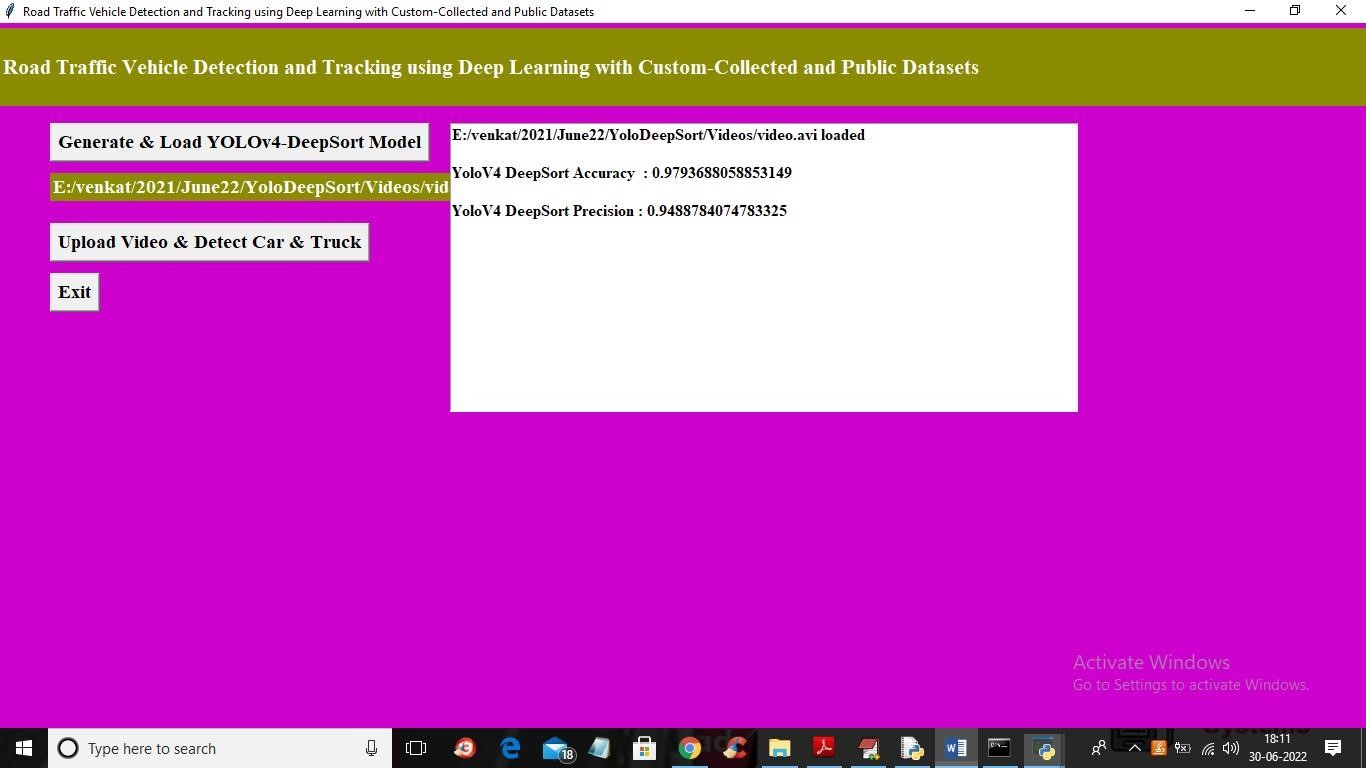
In above screen video will play slowly and then YOLOV4 and DeepSort will start detecting and tracking cars and trucks and in green colour we can see number of real tracked cars



In above screen FPS refer to ‘Frame Per Second’ and application will track vehicles till the end of the video



Similarly you can upload any video and track vehicles



In above screen we can accuracy and precision of propose YOLOV4-DeepSort model

**8. CONCLUSION**

Deep learning is revolutionizing all spheres of our life, smart cities and societies, Industry 4.0, and much more. Transportation is continuing to cause unbelievable damages including 1.25 million deaths and trillions of dollars annually. This paper has presented a study on the use of YOLOv4 for vehicle detection and DeepSORT for tracking the detected vehicles on roads. None of the earlier works have applied these models to road traffic in KSA. We used three different variations of the deep learning models and compared their performance; a pre-trained model with the COCO dataset, and two custom-trained models with the Berkeley DeepDrive dataset, and our customdeveloped dataset obtained by a Dash Cam installed onboard vehicle driven on KSA roads in five different traffic conditions. The five traffic scenarios included city traffic in day and night, highway traffic in day and night, and traffic in the rain. We used the Google Colab platform to harness GPU power, CUDA and OpenCV. The results have been evaluated using precision and tracking success rate and show a mix of performance for the pre-trained and custom-trained models. The pre-trained model was unable to deliver consistently good performance across all five scenarios both in terms of precision and tracking success rate. The results of the custom trained models are comparable to the Pre-Trained model. A large number of misclassification cases by all three models suggest the need of further experimentations. A large number of misclassification cases also suggest that our models are not over-fitted. This shows that there is a need to add more data for training and testing for all three models particularly the custom-trained models. An important finding of this work is that pre-trained models cannot work in KSA environments without retraining due to the differences in the language, driving culture, driving environments, and vehicle models. Future work will look into building larger datasets for vehicle detection, tracking, and other problems in road transportation, and developing highly accurate deep learning models optimized for the environment.

**9.REFERENCES**

1. N. Janbi, I. Katib, A. Albeshri, and R. Mehmood, “Distributed Artificial Intelligence-as-a- Service (DAIaaS) for Smarter IoE and 6G Environments,” Sensors, vol. 20, no. 20, p. 5796, Oct. 2020.
2. T. Yigitcanlar, L. Butler, E. Windle, K. C. Desouza, R. Mehmood, and J. M. Corchado, “Can Building ‘Artificially Intelligent Cities’ Safeguard Humanity from Natural Disasters, Pandemics, and Other Catastrophes? An Urban Scholar’s Perspective,” Sensors, vol. 20, no.

10, p. 2988, May 2020.

1. R. Mehmood, S. See, I. Katib, and I. Chlamtac, Eds., Smart Infrastructure and Applications: Foundations for Smarter Cities and Societies. EAI/Springer Innovations in Communication and Computing, Springer International Publishing, Springer Nature Switzerland AG, pp. 692, 2020.