Traffic Monitoring System using Python-OpenCV and YOLOv8

MURSHIDABAD COLLEGE OF ENGINEERING & TECHNOLOGY

Under guidance of **Prof. Anindya Bakshi**

(HOD, CSE Dept.)

Group Member

Debajyoti Talukder

Sukanya Saha | Surojit Barman | Kazi Nasrin Akhtar

NOVEMBER 2023

Murshidabad College of Engineering and Technology

ESTD : 1998

A Presentation On :

"Traffic Monitoring System Using Python-OpenCV and YOLOV8" under the guidance of Prof. Anindya Bakshi Sir (HOD, CSE Dept.)

Submitted By: Debajyoti Talukder (10600120012), Sukanya Saha (10600120016), Kazi Nasrin Akhter (10600120017), and Surojit Barman (10600121059)

Acknowledgement

I would like to express my profound gratitude and thanks to Prof. Anindya Bakshi, HOD, Department of Computer Science and Engineering (CSE), Murshidabad College of Engineering and Technology, for the intellectual support, inspiring guidance, and his invaluable encouragement, suggestions, and cooperation that helped a lot to complete this project successfully.

Date: 4/12/23

Place: Berhampore, Murshidabad

Abstract

- Traffic monitoring is a critical component of intelligent transportation systems. It helps to improve traffic flow, reduce congestion and enhance safety. Traditional traffic monitoring systems are often expensive and complex to deploy and maintain.
- This project proposes a Traffic Monitoring System using Python-OpenCV and YOLOv8. YOLOv8 is a state-of-the-art object detection algorithm that is fast, accurate and efficient. Python-OpenCV is a popular computer vision library for python.
- The project is implemented using python and is easy to deploy and maintain. It ensures to prove itself to be an efficient tool for traffic engineers and law enforcement authorities to improve traffic safety and efficiency

Introduction

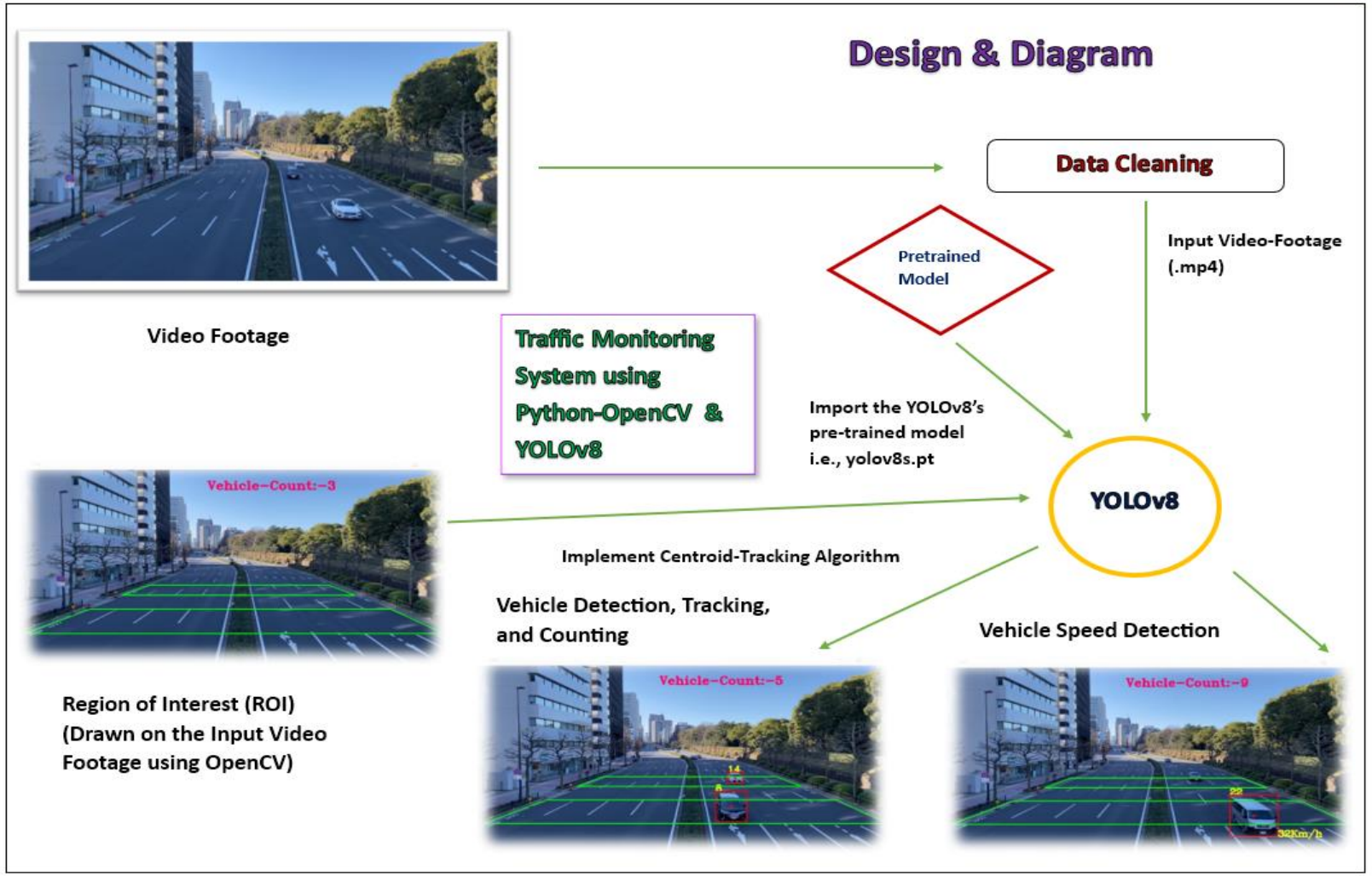
- The existing traffic monitoring systems typically use radar, lidar, or cameras to detect vehicles. These systems are expensive to install and maintain, and they can only detect vehicles in a limited area. The proposed system is a more cost-effective and efficient way to monitor traffic. It uses **Python-OpenCV and YOLOv8** to detect, count and track vehicles in the video footage. The system can also detect vehicle speed and detects if a vehicle is violating the speed limit.

Goal of the Project

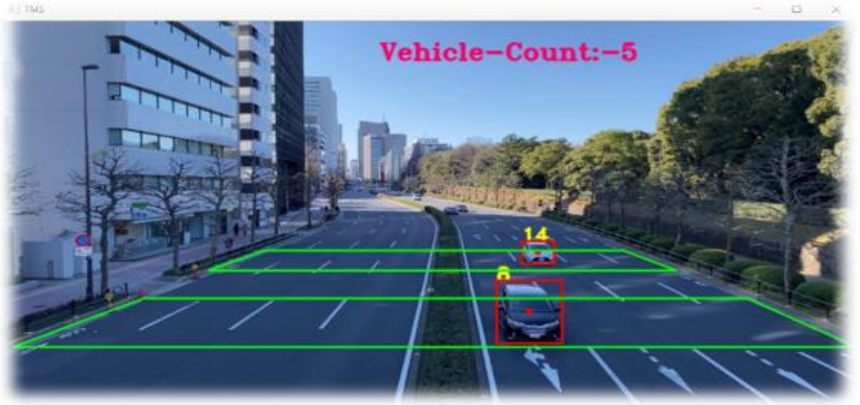
- Main objective and goal of this project is to create a **Traffic Monitoring System using Python-OpenCV and YOLOv8,** which will be able to:

  1. Detect, track, and count vehicles.
  2. Detect vehicle speed.
  3. Detect vehicle speed limit violations.

Design & Diagram



Design & Diagram (Contd.)



Centroid-Tracking Algorithm

1. Bounding Box Co-ordinates obtained fromYOLOv8 Model Predictions

2. Tracker Module (Implements Centroid Tracking Algorithm) Calculates Centroid (cx, cy) of each Vehicles

rect = bounding box co-ordinates of each vehicle obtained from YOLOv8 pretrained model predictions
x, y, w, h = rect
cx = (x + x + w) // 2
cy = (y + y + h) // 2

3. Calculates distance between every pair of possible centroids of moving vehicles

dist = math.hypot(cx - pt[0], cy - pt[1])

// Uses Eucledian Diatance formula

// If this distance is less than a threshold value, the object or vehicle is considered to be the same otherwise the vehicle will be considered as different

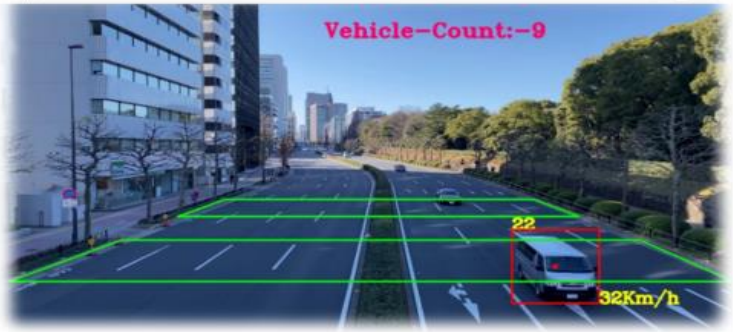
// same objects will be assigned same ID and different objects will be assigned different IDs

4. Assigns ID to every bounding box it detects

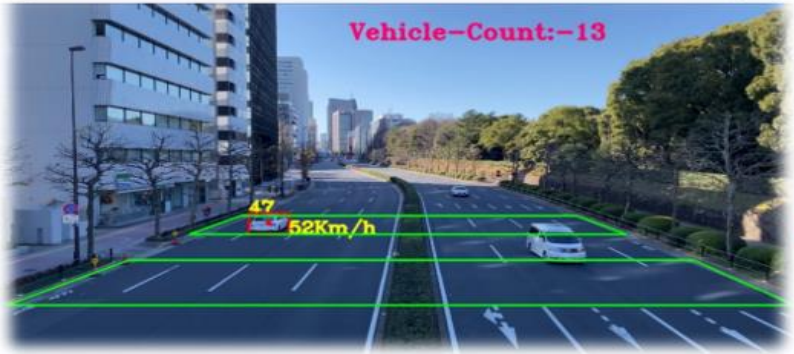
5. Returns bounding box co-ordinates and object ID assigned

Design & Diagram (Contd.)

We have used the YOLOv8's pretrained model (e.g., yolov8s.pt). All YOLOv8 models for object detection are already pre-trained on the COCO dataset, which is a huge collection of images of 80 different types.



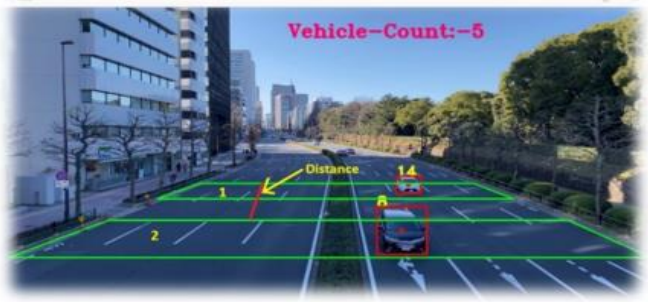
Vehicle Speed Detection

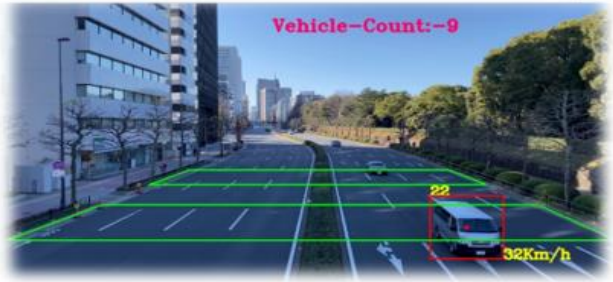


Vehicle Speed Detection
(Vehicles moving in another direction)

The speed of the vehicles that are being tracked can be estimated using the distance between the starting line of first Region of Interest (ROI) and starting line of second Region of Interest (ROI) and the time taken by the vehicle to cover the distance.

Design & Diagram (Contd.)





1. Import Tracker Module (Returns bounding box co-ordinates and assigned IDs of the vehicles).
2. Import required Python Libraries (NumPy, Pandas, OpenCV).
3. Import YOLOv8's Pretrained Model, i.e., yolov8s.pt
4. The centroid tracking algorithm implemented using the Tracker Module returns bounding box co-ordinates and object ID assigned.
5. counts the number of vehicles that passes through the second region of interest (ROI) [This is the main ROI for vehicle counting].

6. Initializes dictionaries to store IDs and corresponding elapsed time taken by the vehicles to cover the distance between the two Regions of Interest (ROI) chosen.

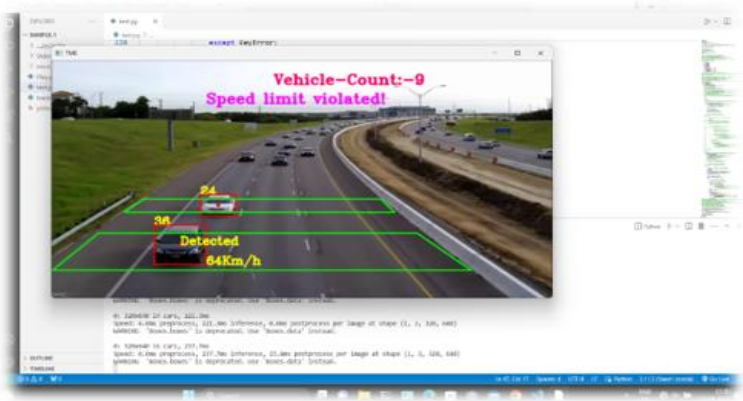
$$Speed = \frac{distance}{Elapsed\ Time}$$

7. Checks if the vehicle is forward moving vehicles (moving from first Region of Interest / ROI to second Region of Interest / ROI). Display the vehicle speed, Id, and bounding box for each detected vehicle. For the backward moving vehicles (moving from second Region of Interest / ROI to first Region of Interest / ROI), it displays the bounding box, ID, and vehicle speed when the vehicle passes the first Region of Interest / ROI.

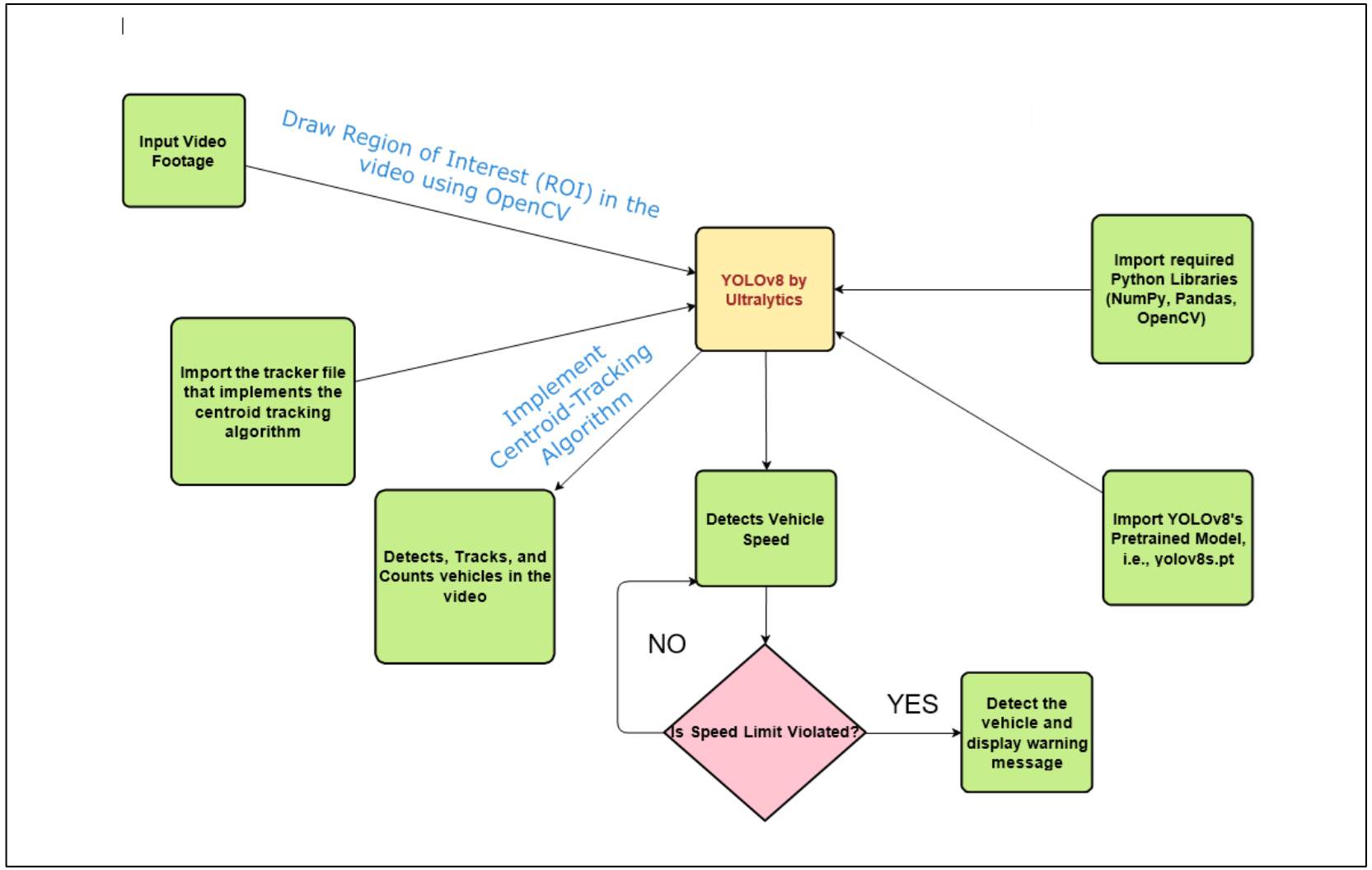
Design & Diagram (Contd.)

Let, speed_limit = 60 #Set Speed Limit

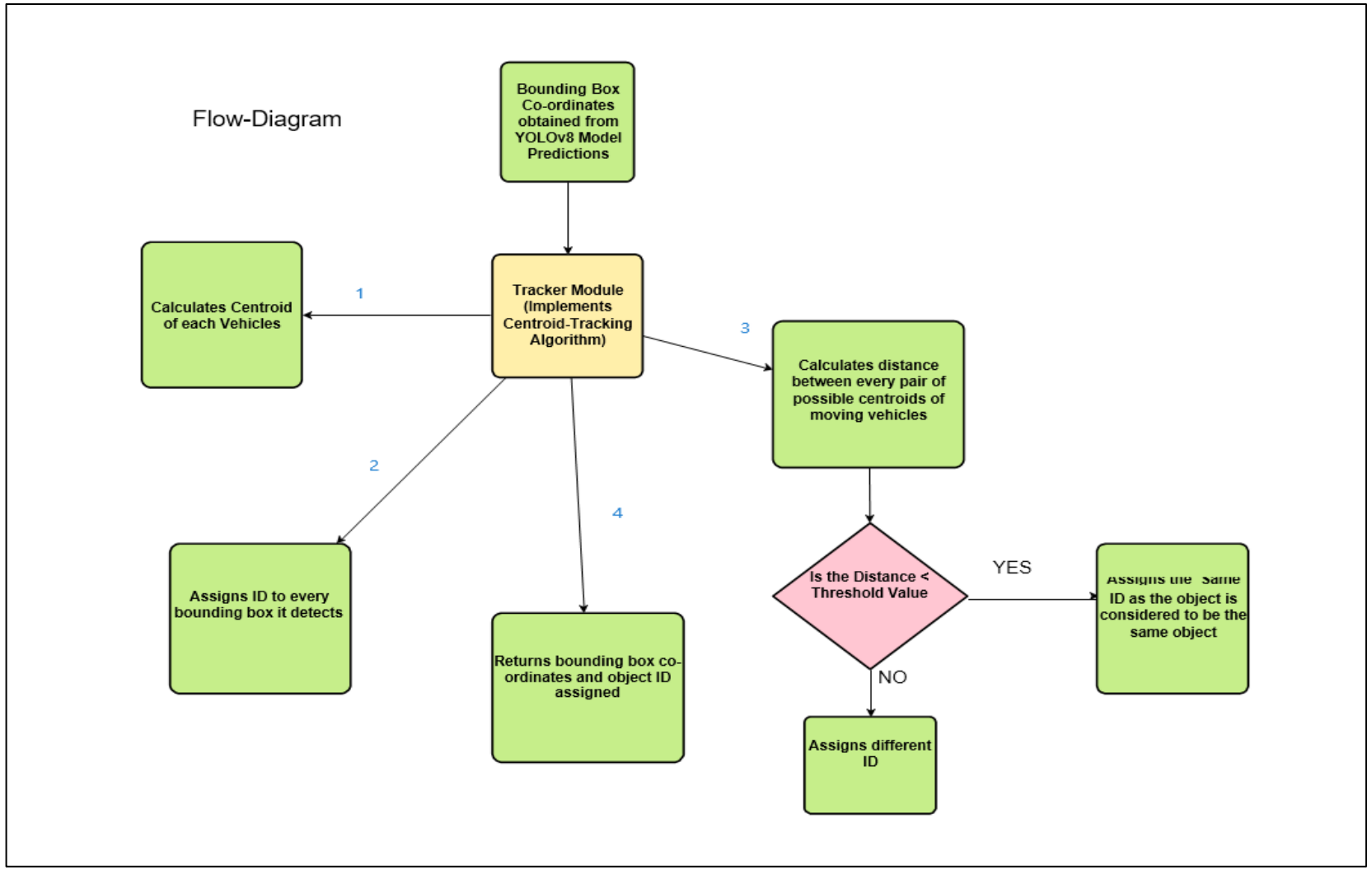
A warning message will be displayed if the system detects any vehicle speed limit violations



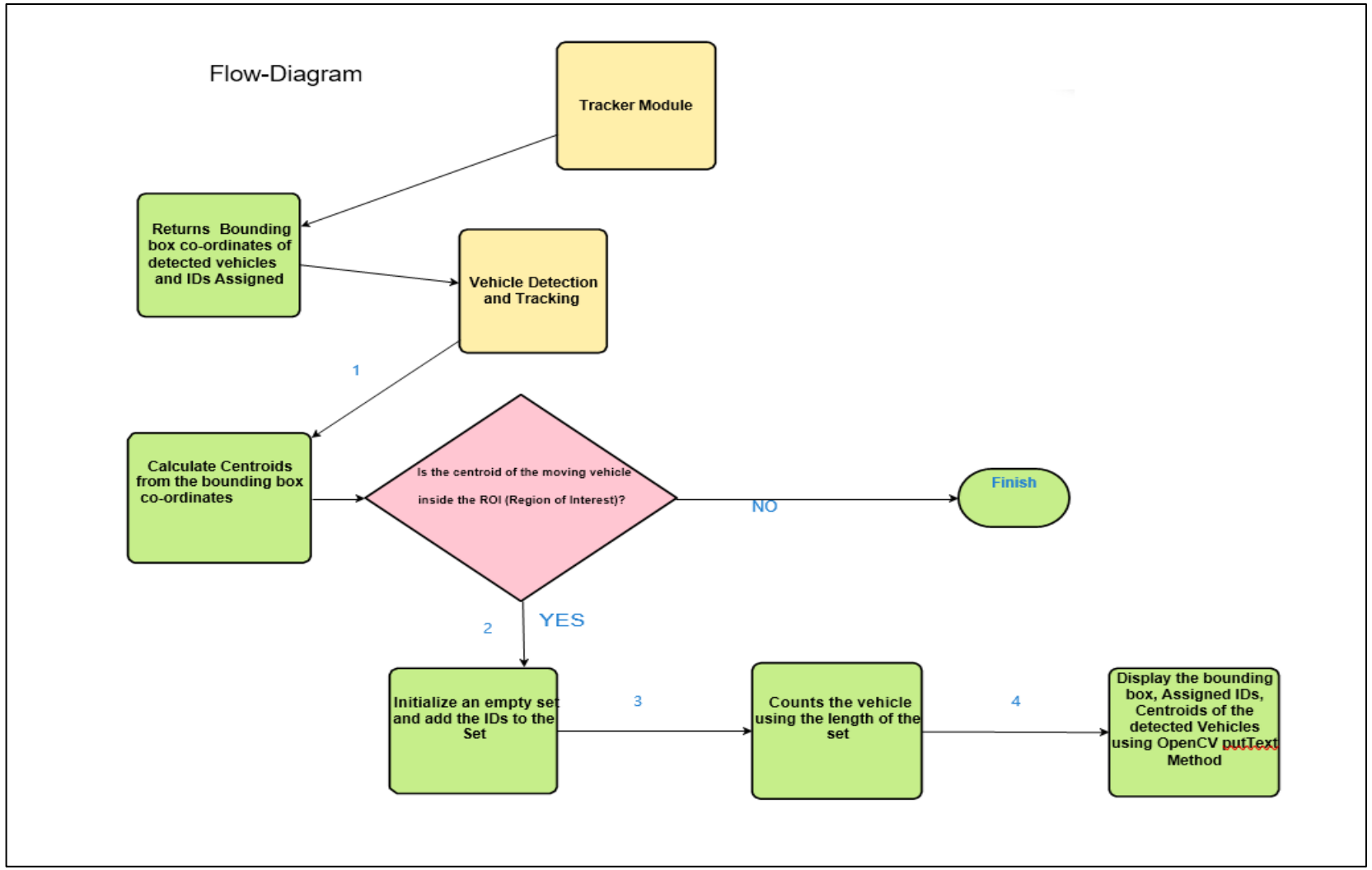
Design & Diagram (Flow Chart)



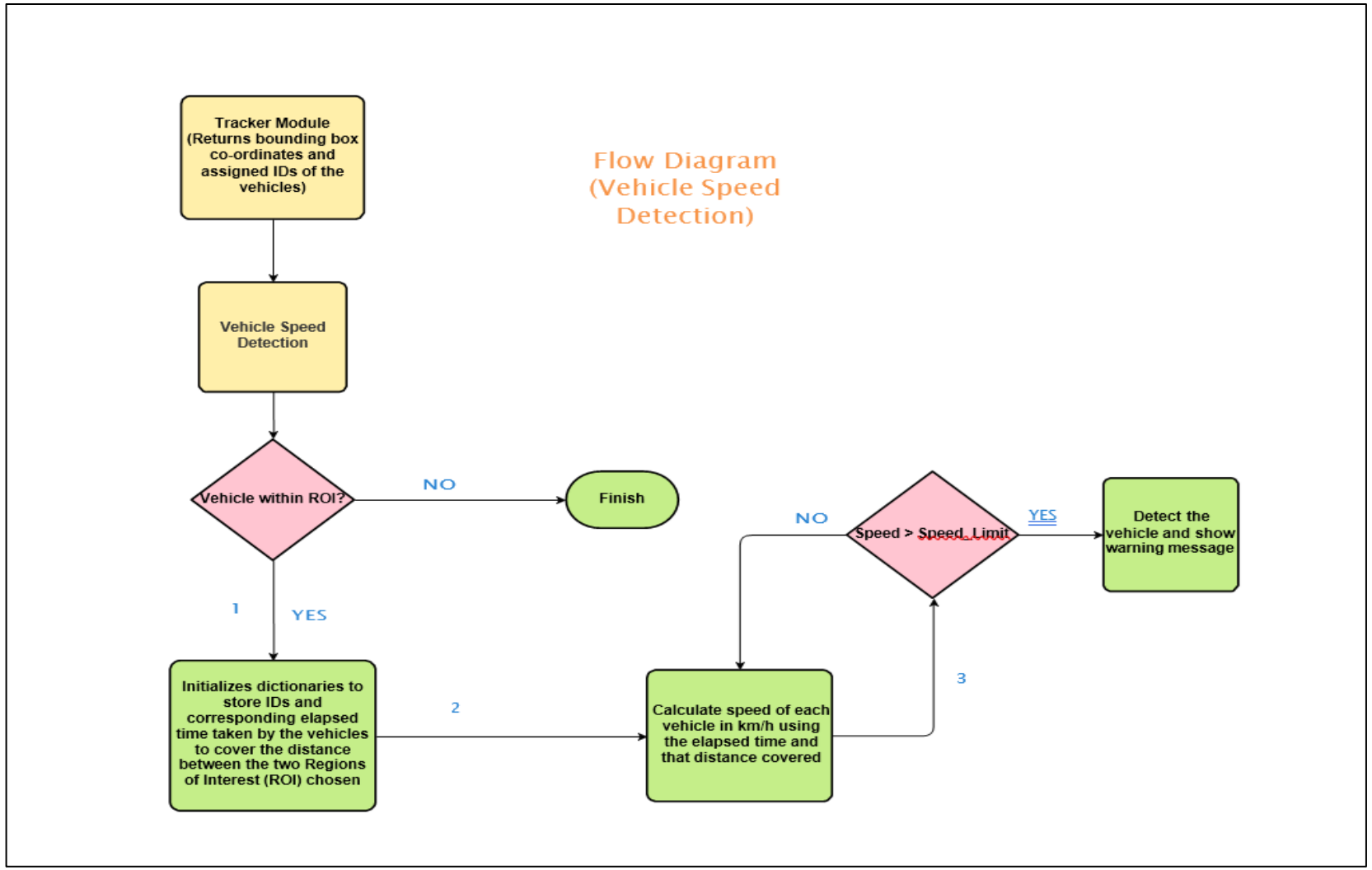
Design & Diagram (Flow Chart) (Contd.)



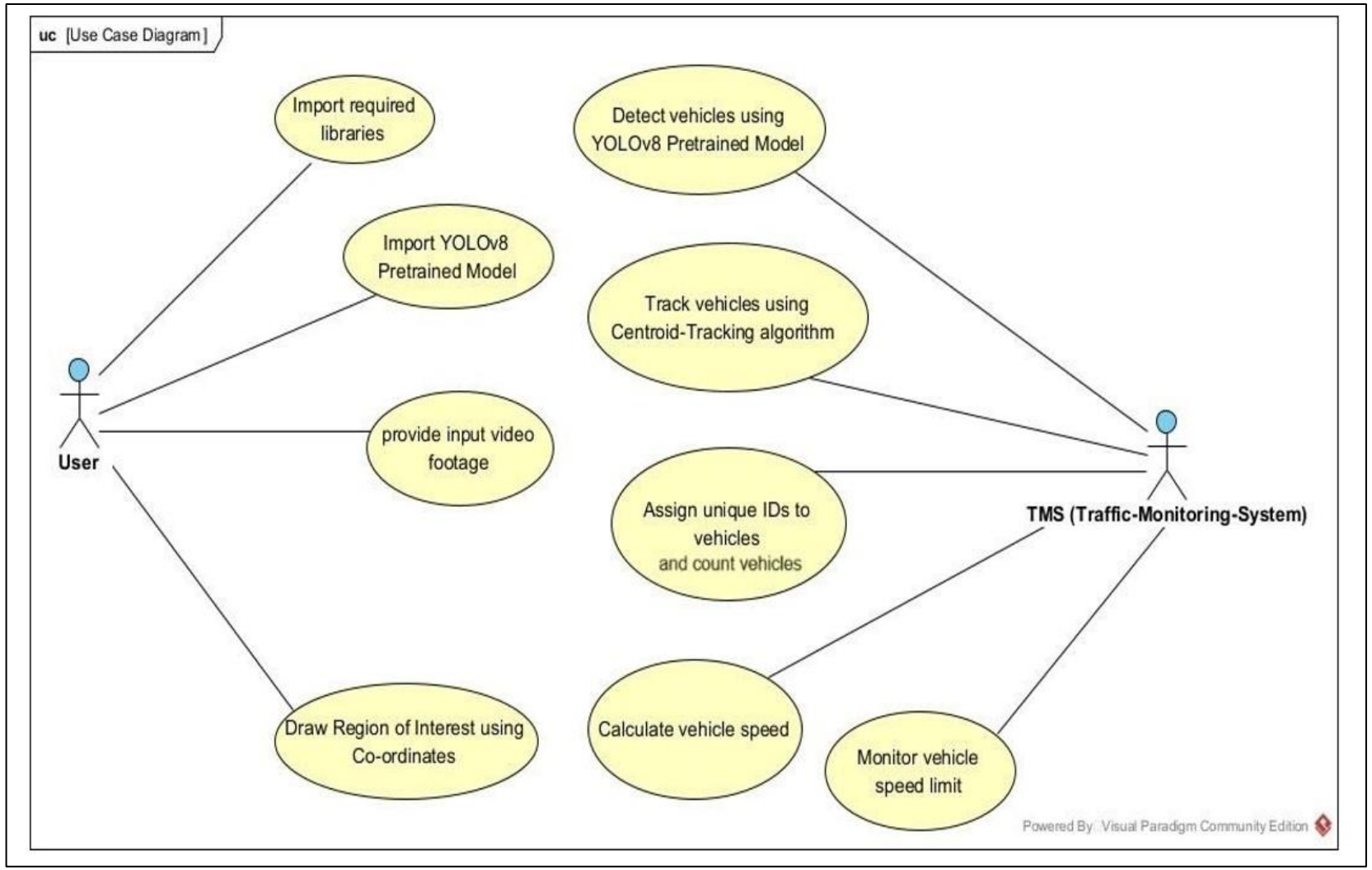
Design & Diagram (Flow Chart) (Contd.)



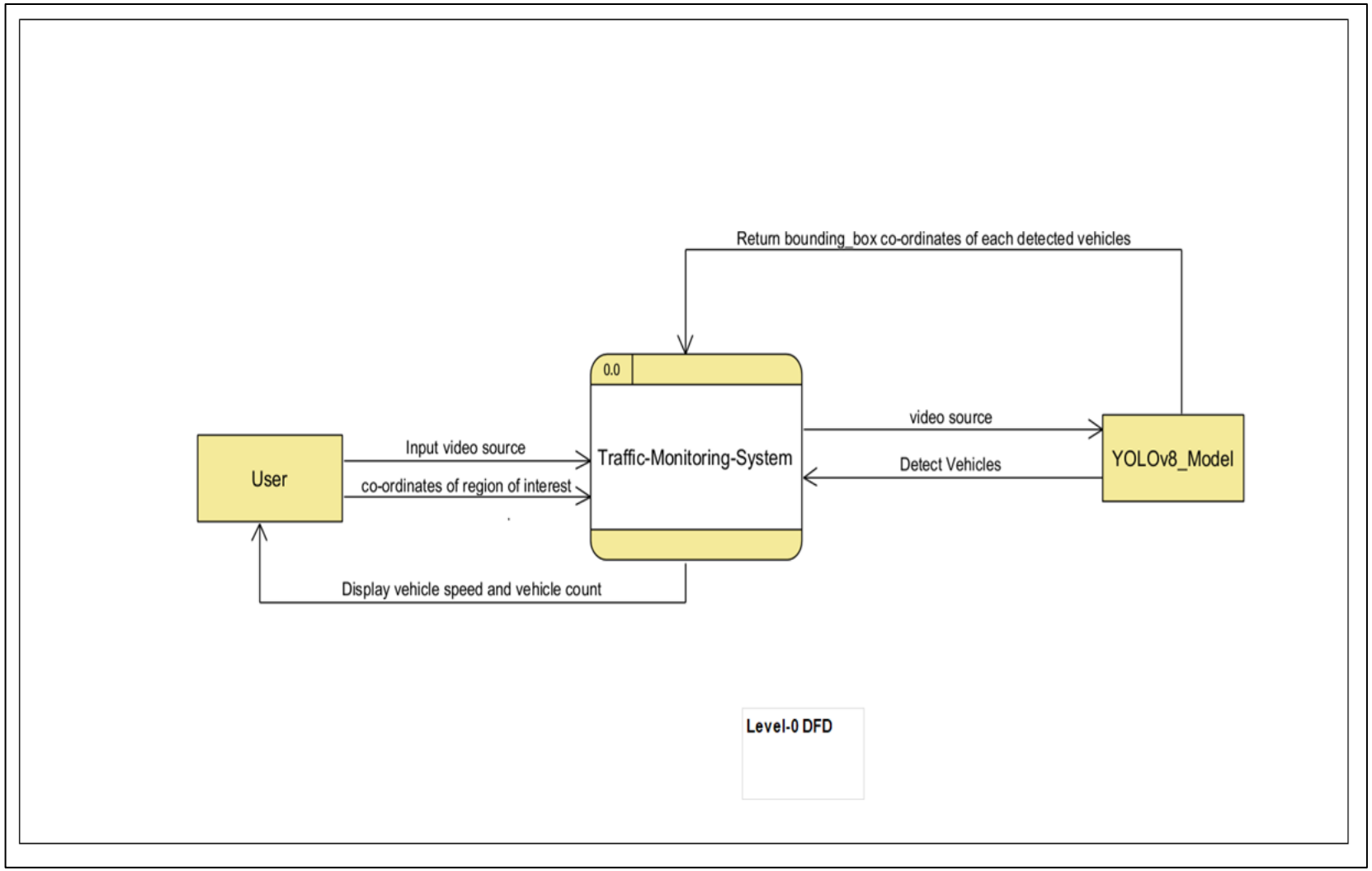
Design & Diagram (Flow Chart) (Contd.)



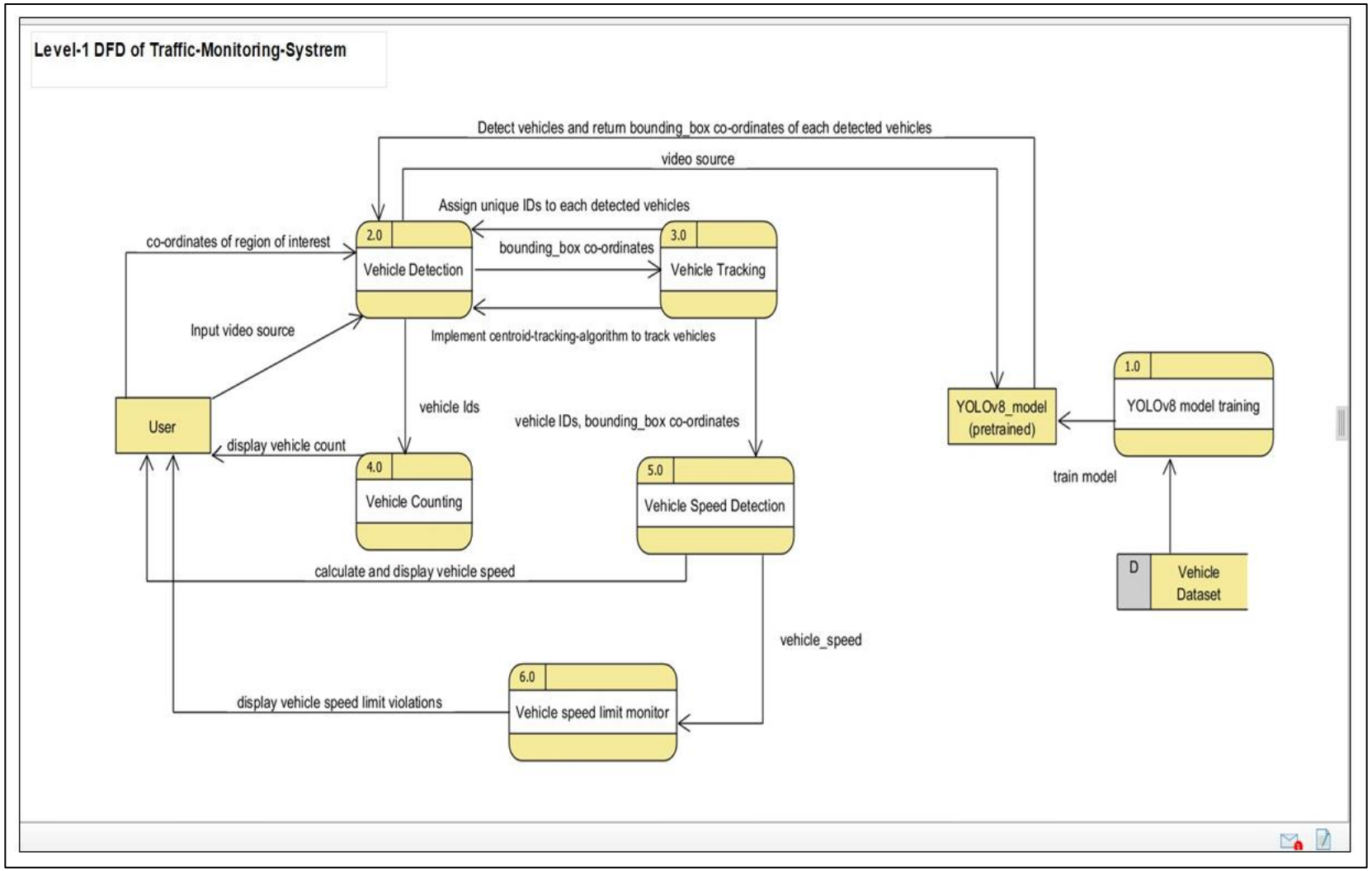
Design & Diagram (Use Case Diagram)



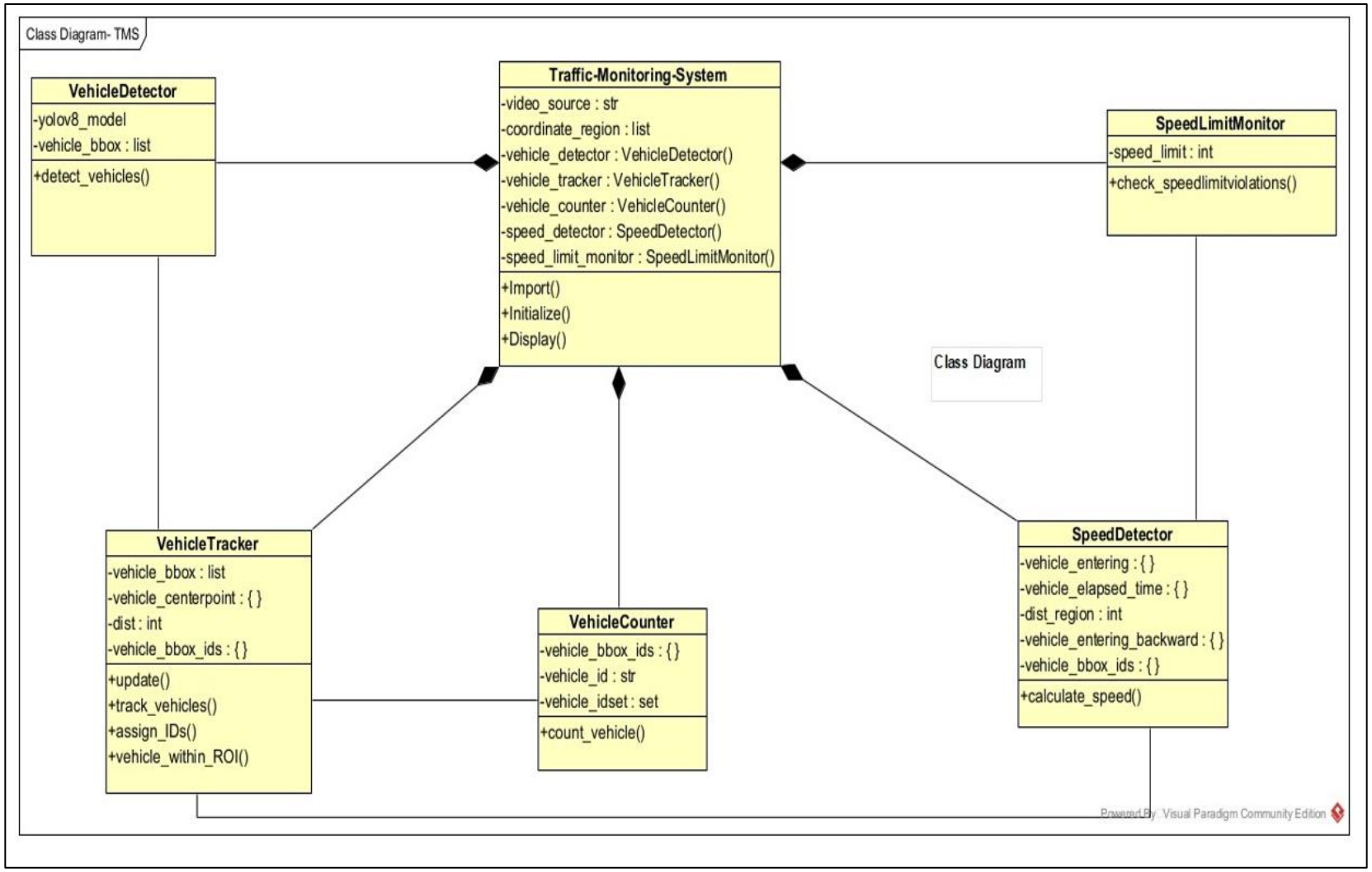
Design & Diagram (DFD)



Design & Diagram (DFD)



Design & Diagram (Class Diagram)



Basic Information

- Machine Learning is a subfield of Artificial Intelligence that enables machines to improve at a given task with experience. It is important to note that all machine learning techniques are classified as Artificial Intelligence ones.
- In this project we have used Deep Learning, a specialized field of Machine Learning that that relies on training of Deep Artificial Neural Networks (ANNs) using a large dataset such as images or texts.
- What differentiates deep learning from machine learning techniques are in their ability to extract feature automatically.

How to split a dataset

TRAINING SET

The subset of data used to train a machine learning model

TEST SET

The subset of data used to evaluate the performance of a trained machine learning model on unseen examples, simulating real-world data

VALIDATION SET

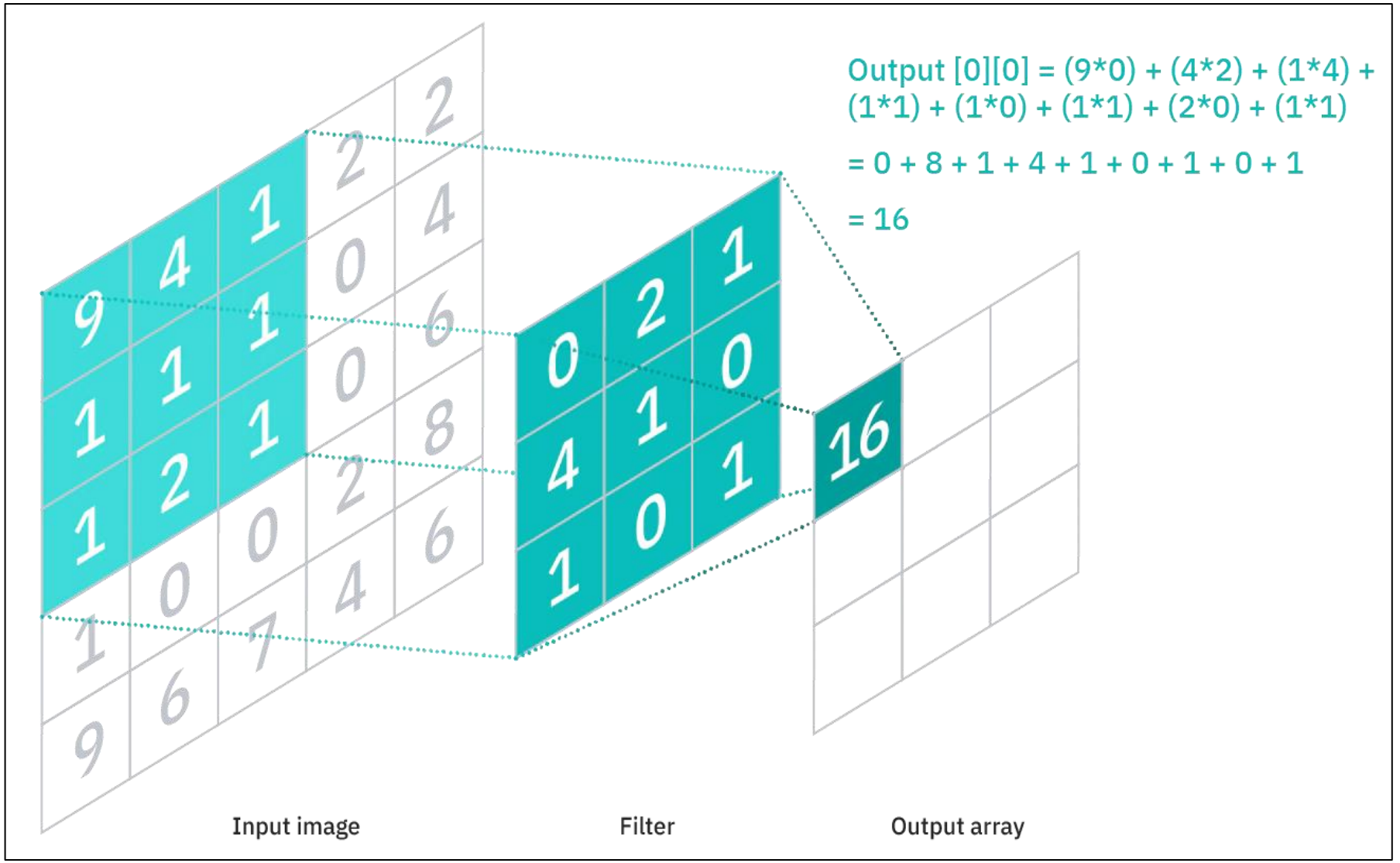
The intermediary subset of data used during the model development process to fine-tune hyperparameters

Convolutional Neural Networks (CNN)

- Convolutional Neural Networks (CNN) is the most successful Deep Learning method used to process multiple arrays, e.g., 1D for signals, 2D for images and 3D for videos. CNN consists of a list of Neural Network layers that transform the input data into an output (class/prediction).
- CNN Architecture:
- Convolutional Neural Network consists of multiple layers like the input layer, Convolutional layer, Pooling layer, and fully connected layers.
- The Convolutional layer applies filters to the input image to extract features, the Pooling layer down-samples the image to reduce computation, and the fully connected layer makes the final prediction. The network learns the optimal filters through backpropagation and gradient descent.

How CNN Works?

- The convolutional layer is the core building block of a CNN, and it is where the majority of computation occurs. It requires a few components, which are input data, a filter, and a feature map. Let's assume that the input will be a color image, which is made up of a matrix of pixels in 3D. This means that the input will have three dimensions—a height, width, and depth—which correspond to RGB in an image. We also have a feature detector, also known as a kernel or a filter, which will move across the receptive fields of the image, checking if the feature is present. This process is known as a convolution.
- The feature detector is a two-dimensional (2-D) array of weights, which represents part of the image. While they can vary in size, the filter size is typically a 3x3 matrix; this also determines the size of the receptive field. The filter is then applied to an area of the image, and a dot product is calculated between the input pixels and the filter. This dot product is then fed into an output array. Afterwards, the filter shifts by a stride, repeating the process until the kernel has swept across the entire image. The final output from the series of dot products from the input and the filter is known as a feature map, activation map, or a convolved feature.


Output [0][0] = (9*0) + (4*2) + (1*4) +
(1*1) + (1*0) + (1*1) + (2*0) + (1*1)
= 0 + 8 + 1 + 4 + 1 + 0 + 1 + 0 + 1
= 16
Input image
Filter
Output array

Introduction to YOLOv8

- Technically speaking, YOLOv8 is a group of convolutional neural network models, created and trained using the PyTorch framework. YOLOv8 is the latest version of YOLO by **Ultralytics**. As a cutting-edge, state-of-the-art (SOTA) model, YOLOv8 builds on the success of previous versions, introducing new features and improvements for enhanced performance, flexibility, and efficiency. YOLOv8 supports a full range of vision AI tasks, including detection, segmentation, pose estimation, tracking, and classification. This versatility allows users to leverage YOLOv8's capabilities across diverse applications and domains.
- **YOLOv8** is the latest family of YOLO based Object Detection models from **Ultralytics** providing state-of-the-art performance. Leveraging the previous YOLO versions, the YOLOv8 model is faster and more accurate while providing a unified framework for training models for performing Object Detection, Instance Segmentation, and Image Classification.

ultralytics

YOLOv8

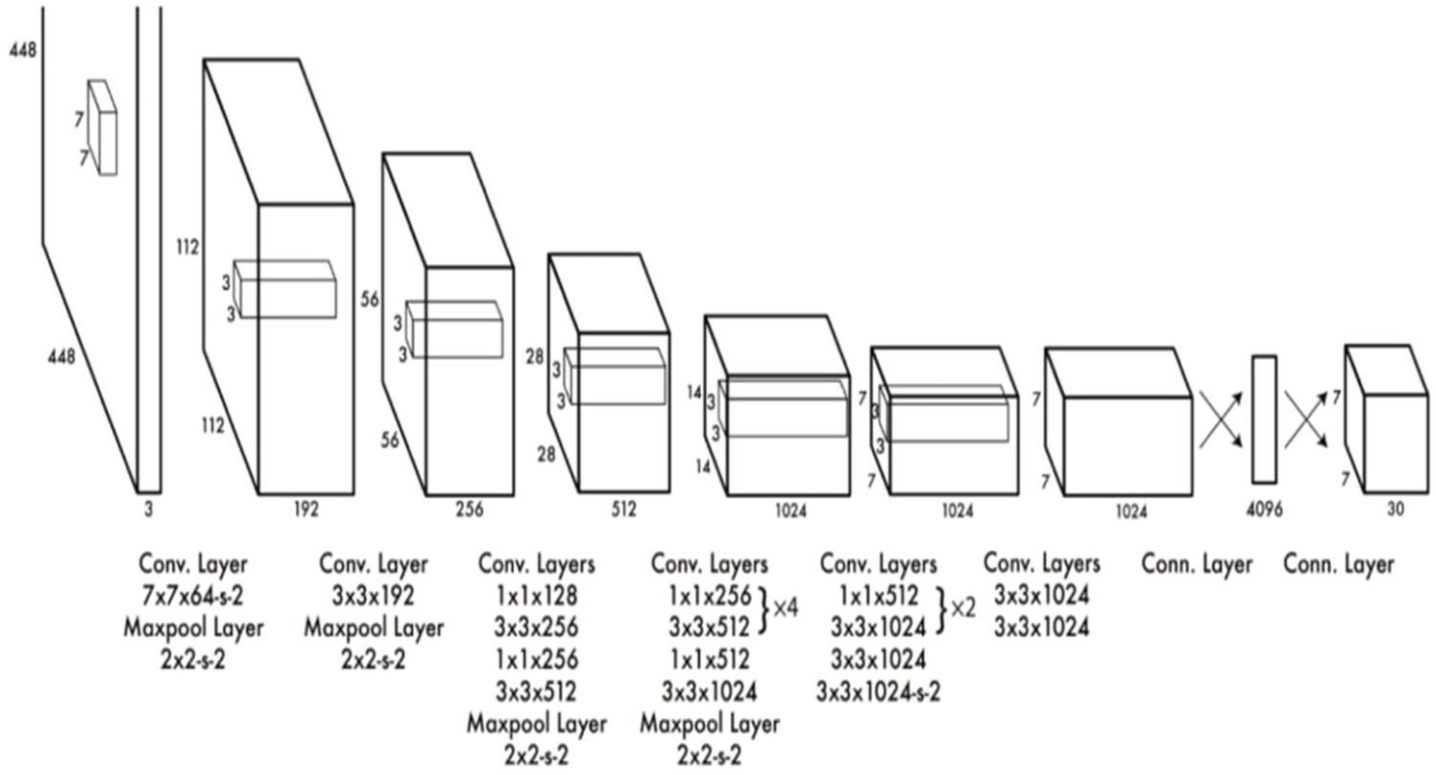
State-of-the-Art YOLO Models

How CNN Classifies Traffic Video Footages ?

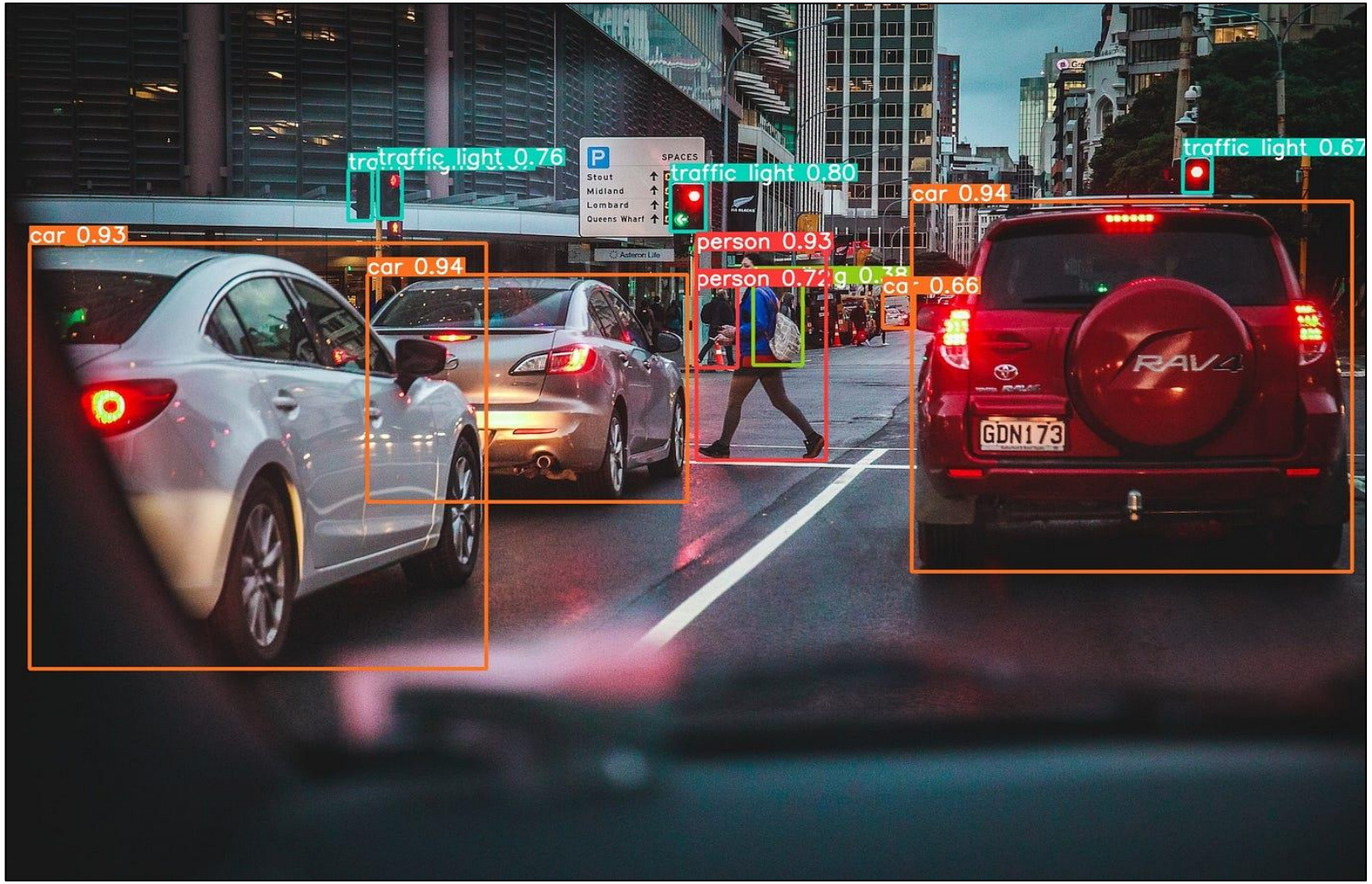
- YOLO is a single-shot detector that uses a fully convolutional neural network (CNN) to process an image. You Only Look Once (YOLO) proposes using an end-to-end neural network that makes predictions of bounding boxes and class probabilities all at once. It differs from the approach taken by previous object detection algorithms, which repurposed classifiers to perform detection.
- Following a fundamentally different approach to object detection, YOLO achieved state-of-the-art results, beating other real-time object detection algorithms by a large margin. While algorithms like Faster RCNN work by detecting possible regions of interest using the Region Proposal Network and then performing recognition on those regions separately, YOLO performs all of its predictions with the help of a single fully connected layer.
- YOLOv8 utilizes a convolutional neural network that can be divided into two main parts: the backbone and the head. The head of YOLOv8 consists of multiple convolutional layers followed by a series of fully connected layers. These layers are responsible for predicting bounding boxes, objectness scores, and class probabilities for the objects detected in an image.

Contd.

- YOLOv8 introduces a **new backbone network**, Darknet-53, which is significantly faster and more accurate than the previous backbone used in YOLOv7. DarkNet-53 is a convolutional neural network that is 53 layers deep and can classify images into 1000 object categories.
- Additionally, YOLOv8 is more efficient than previous versions because it uses a larger feature map and a more efficient convolutional network. This allows the model to detect objects in a more accurate and faster way. With a larger feature map, the model can capture more complex relationships between different features and can better recognize patterns and objects in the data. A larger feature map also helps to reduce the amount of time it takes to train the model and can help to reduce overfitting.
- Overall, YOLOv8 is a powerful and versatile object detection algorithm that can be used in various real-world scenarios to detect and classify objects with high accuracy and speed.



The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224×224 input image) and then double the resolution for detection.



YOLOv8 Object Detection from Traffic Video Footage

Models Available in YOLOv8

- There are five models in each category of YOLOv8 models for detection, segmentation, and classification. YOLOv8 Nano is the fastest and smallest, while YOLOv8 Extra Large (YOLOv8x) is the most accurate yet the slowest among them.
- **YOLOv8** comes bundled with the following **pre-trained** models:
- **Object Detection** checkpoints trained on the COCO detection dataset with an image resolution of 640.
- **Instance segmentation** checkpoints trained on the COCO segmentation dataset with an image resolution of 640.
- **Image classification** models pretrained on the ImageNet dataset with an image resolution of 224.
- The pretrained YOLOv8 model (e.g., **yolov8s.pt**) can be downloaded from the YOLO website. All YOLOv8 models for object detection are already pre-trained on the **COCO dataset,** which is a huge collection of images of 80 different types. The custom dataset, if required, can also be created by collecting a set of images that contain vehicles. The images should be labelled with the bounding boxes of the vehicles.

Modules that We Used in this Project

Module	Uses
Python-OpenCV	**OpenCV** is a Python open-source library, which is used for computer vision in Artificial intelligence, Machine Learning, face recognition, etc.
Pandas	**Pandas** is a library for python programming language. It is used for create, remove, edit and data manipulation in dataframe.
NumPy	**NumPy** is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

Contd.

Module	Uses
Ultralytics YOLOv8	YOLOv8 is designed to be fast, accurate, and easy to use, making it an excellent choice for a wide range of object detection and tracking, instance segmentation, image classification and pose estimation tasks.
Tracker	This is a custom module based on the Centroid tracking algorithm to track the vehicles.
Vehicle detection module	This module uses **YOLOv8** to detect vehicles in the video footage. YOLOv8 is a deep learning algorithm that can detect objects in images and videos. The algorithm is trained on a dataset of images that contain vehicles. When the algorithm is applied to a video footage, it can detect the vehicles in the video.

Contd.

Module	Uses
Vehicle tracking module	This module tracks the vehicles that have been detected by the vehicle detection module. The module uses a **centroid tracking algorithm** to track the vehicles. The centroid tracking algorithm tracks the centre of mass of the vehicles in the video footage.
Vehicle speed detection module	This module estimates the speed of the vehicles that are being tracked. To estimate the speed of the vehicles, it uses the distance between the two regions of interest (ROI) chosen and the elapsed time taken by the vehicles to cover that distance.

Contd.

Module	Uses
Vehicle counting module	This module counts the number of vehicles that pass through a specific area in the video footage. The module uses a region of interest (ROI) to define the specific area. The module then counts the number of vehicles that enter and exit the ROI.
Vehicle speed violation detection module:	This module detects if a vehicle is violating the speed limit. The module uses the speed of the vehicles that are being tracked to determine if they are violating the speed limit.

Conclusion

- This project proposes an approach to detect and track the moving vehicles and estimation of their speeds. The innovation of the approach lies in the selection of the Region of Interest for the vehicle detection. The traffic monitoring system developed in this project can be used for a variety of applications, such as traffic management, public safety, parking management, and smart cities. The system uses YOLOv8, a state-of-the-art object detection algorithm, which is known for its speed and accuracy. We also used the centroid tracking algorithm to track the vehicles. The centroid tracking algorithm works by tracking the centroids of the objects detected by YOLOv8. The centroids are the points at the center of the objects. The centroid tracking algorithm then tracks the movement of the objects by tracking the movement of their centroids. The system has several limitations, such as the difficulty of detecting vehicles in difficult conditions, such as occlusion or poor lighting. However, the system is a promising approach to traffic monitoring and can be improved in future work.

Future Scope

- The future scope of the project is to improve the accuracy of the system, extend its capabilities, deploy it in a real-world setting, and integrate it with other systems.
- To improve the accuracy of the system, a larger and more diverse dataset of traffic images can be used. A more powerful deep learning model can also be used. Overall, the future scope of the project is to develop a robust and reliable traffic monitoring system that can be used to improve traffic safety and efficiency

THANK YOU