Traffic Management

Project Description:

Traffic congestion is becoming a serious problem with a large number of cars on the roads. Vehicles queue length waiting to be processed at the intersection is rising sharply with the increase of the traffic flow, and the traditional traffic lights cannot efficiently schedule it.

ows at the signalized road intersection. This is done by a state-of-the-art, real-time object detection based on a deep Convolutional Neural Networks called You Only Look Once (YOLO). Then traffic signal phases are optimized according to collected data, mainly queue density and waiting time per vehicle, to enable as much as more vehicles to pass safely with minimum waiting time. YOLO can be implemented on embedded controllers using Transfer Learning technique.

YOLO uses a totally different approach. YOLO is a clever convolutional neural network (CNN) for doing object detection in real-time. The algorithm applies a single neural network to the full image, and then divides the image into regions and predicts bounding boxes and technique.

Project Implementation:

1.To build a self adaptive traffic control system based on yolo.

2.Disproportionate and diverse traffic in different lanes leads to inefficient utilization of same time slot for each of them characterized by slower speeds, longer trip times, and increased vehicular queuing .

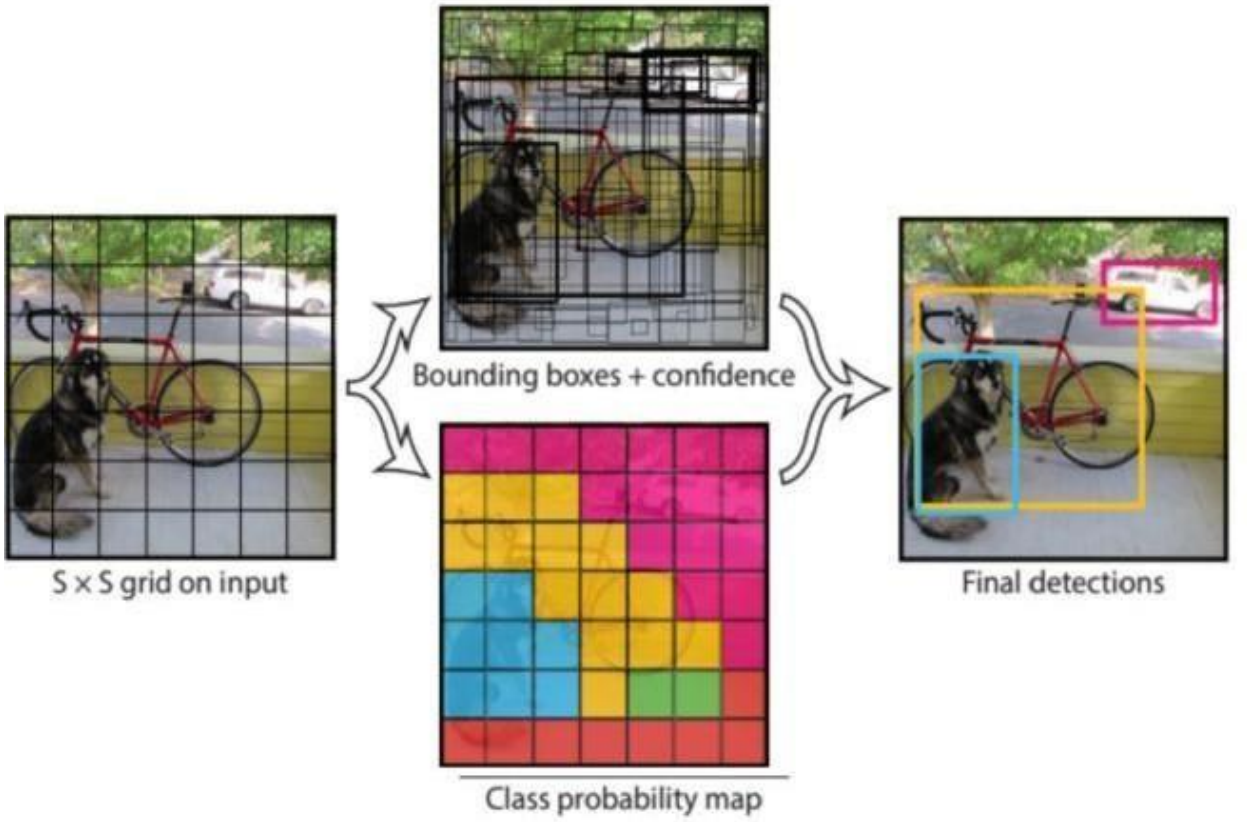
3.To create a system which enable the traffic management system to take time allocation decisions for a particular lane according to the traffic density on other different lanes .

4. Technology :

4.1 YOLO

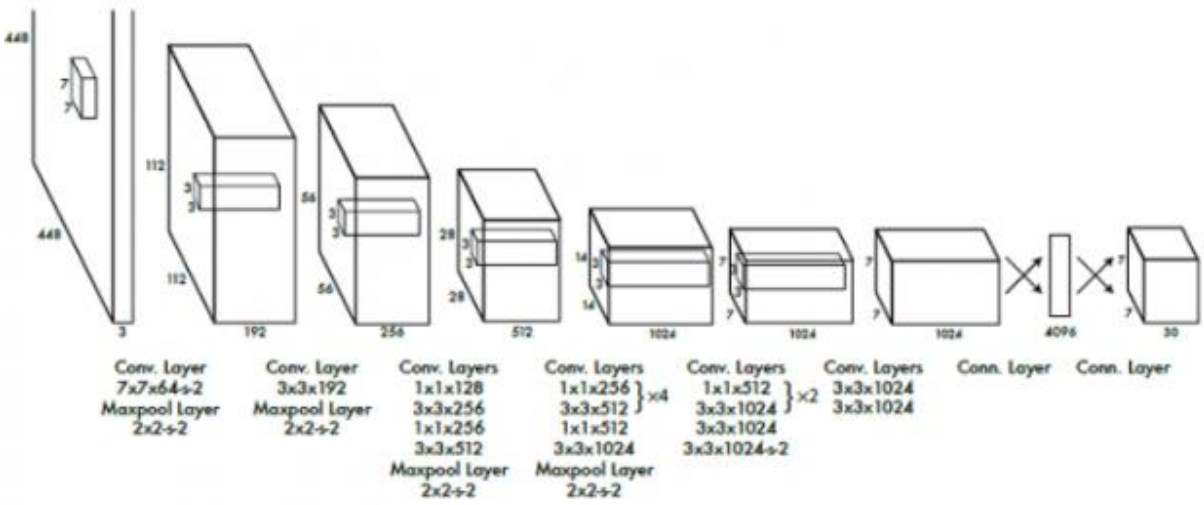
You only look once (YOLO) is a state-of-the-art, real-time object detection systemYOLO, a new approach to object detection. Prior work on object detection repurposes classifiers to perform detection. Instead, we frame object detection as a regression problem to spatially separated bounding boxes and associated class probabilities.

A single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation. Since the whole detection pipeline is a single network, it can be optimized end-to-end directly on detection performance.elp of cameras, image processing modules.



The object detection task consists in determining the location on the image where certain objects are present, as well as classifying those objects. Previous methods for this, like R-CNN and its variations, used a pipeline to perform this task in multiple steps.

This can be slow to run and also hard to optimize, because each individual component must be trained separately. YOLO, does it all with a single neural network.



4.2 YoloV3 Car Counter

This is a demo project that uses YoloV3 neural network to count vehicles on a given video. The detection happens every x frames where x can be specified. Other times the dlib library is used for tracking previously detected vehicles.

Furthermore, you can edit confidence detection level, number of frames to count vehicle as detected before removing it from trackable list .

The maximum distance from centroid (see CentroidTracker class), number of frames to skip detection (and only use tracking) and the whether to use the original video size as annotations output or the YoloV3 416x416 size.

YoloV3 model is pretrained and downloaded (Internet connection is required for the download process).

Circuit Diagram:

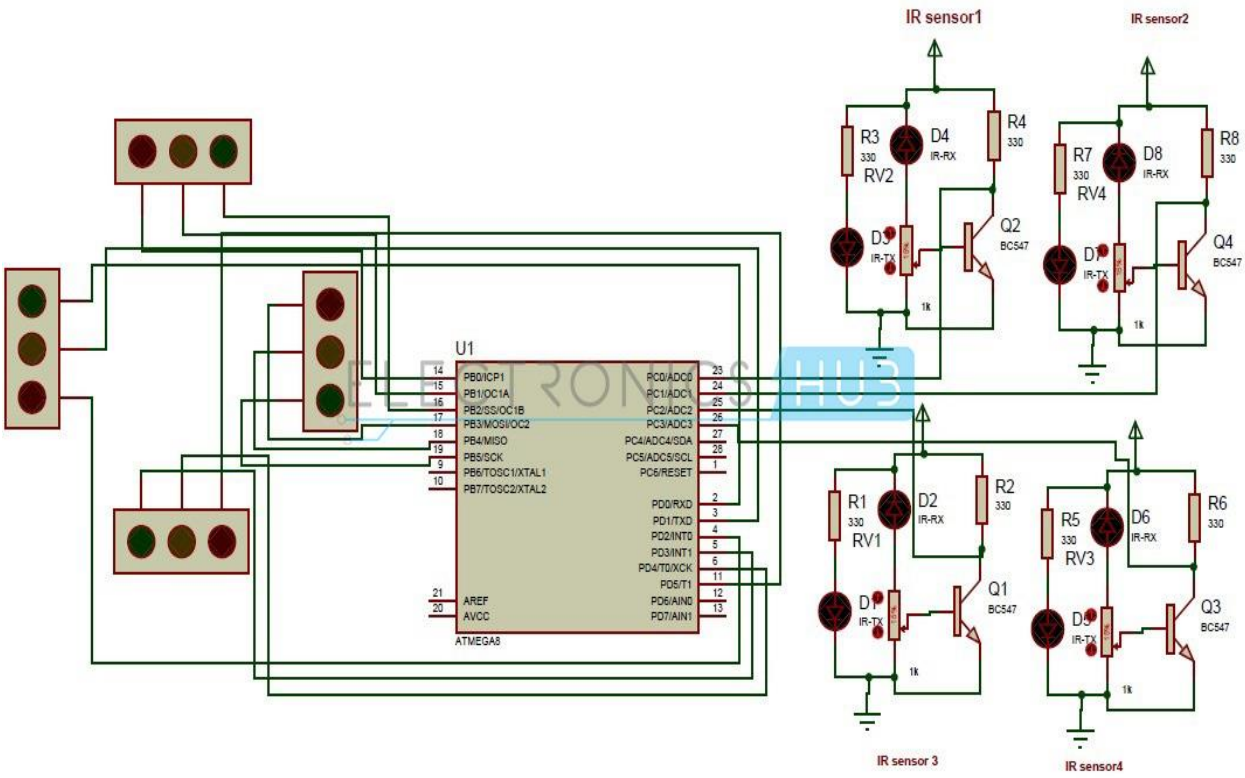


Fig: circuit diagram for traffic management system

Source Code:

```
f = open("out.txt", "r")

no_of_vehicles=[]

no_of_vehicles.append(int(f.readline()))

no_of_vehicles.append(int(f.readline()))

no_of_vehicles.append(int(f.readline()))

no_of_vehicles.append(int(f.readline()))

baseTimer = 120 # baseTimer = int(input("Enter the base timer value"))
```

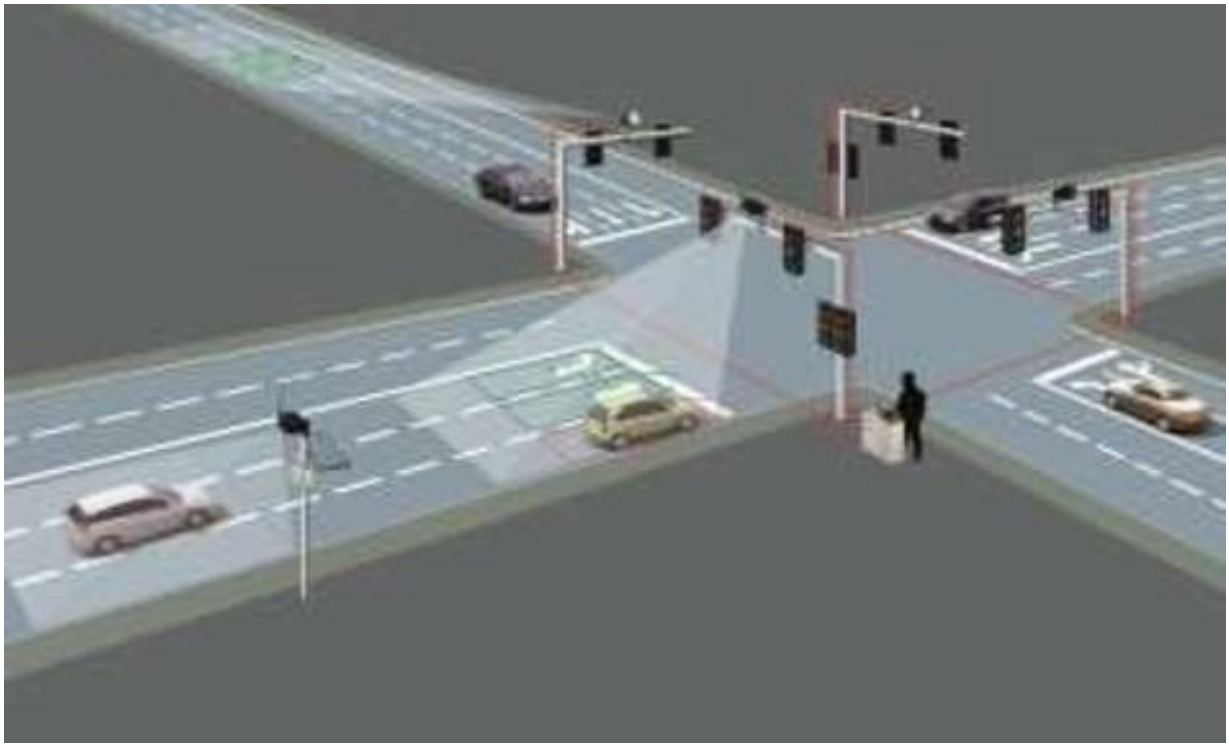
timeLimits = [5, 30] # timeLimits = list(map(int,input("Enter the time limits ").split()))

print("Input no of vehicles : ", *no_of_vehicles)

t = [(i / sum(no_of_vehicles)) * baseTimer if timeLimits[0] < (i / sum(no_of_vehicles)) * baseTimer < timeLimits[1] else min(timeLimits, key=lambda x: abs(x - (i / sum(no_of_vehicles)) * baseTimer)) for i in no_of_vehicles]

print(t, sum(t))

Working:



The solution can be explained in four simple steps:

1.Get a real time image of each lane.

2.Scan and determine traffic density.

3.Input this data to the Time Allocation module.

4.The output will be the time slots for each lane, accordingly.

The goal of this work is to improve intelligent transport systems by developing a Self-adaptive algorithm to control road traffic based on deep Learning.

This new system facilitates the movement of cars in intersections, resulting in reducing congestion, less CO2 emissions, etc.

The richness that video data provides highlights the importance of advancing the state-of-the-art in object detection, classi・cation and tracking for real-time applications.

YOLO provides extremely fast inference speed with slight compromise in accuracy, especially at lower resolutions and with smaller objects.

While real-time inference is possible, applications that utilize edge devices still require improvements in either the architecture's design or edge device's hardware.

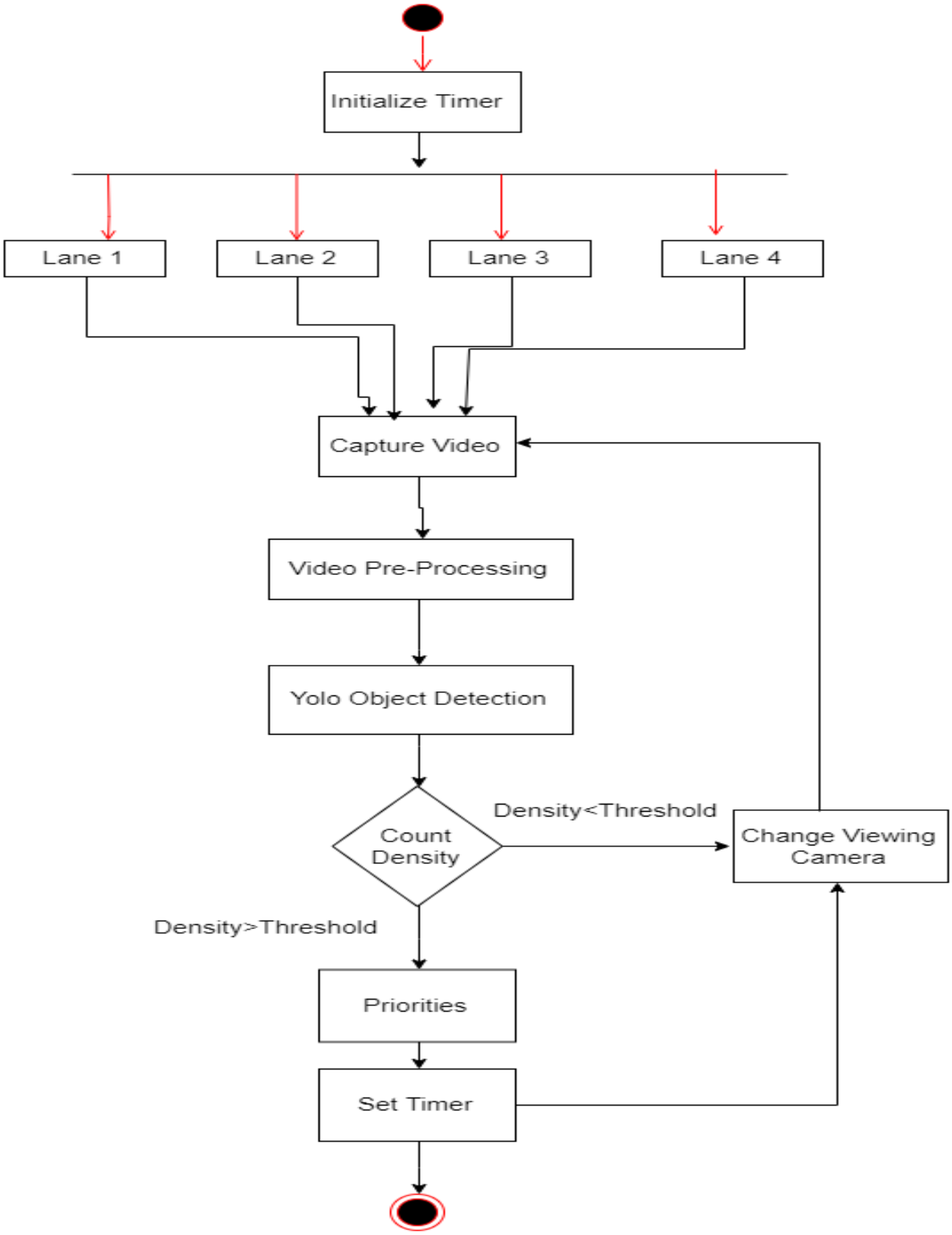
Final This is done by a state-of-the-art, real-time object detection based on a deep Convolutional Neural Networks called You Only Look Once (YOLO).

Then traffic signal phases are optimized according to collected data, mainly queue density and waiting time per vehicle, to enable as much as more vehicles to pass safely with minimum waiting time.

YOLO can be implemented on embedded controllers using Transfer Learning technique.

lly, we have proposed a new algorithm taking this real-time data from YOLO.

Yolo Object Detection



Sequence of operations performed:

1.Camera sends images after regular short intervals to our system.

2.The system determines further the number of cars in the lane and hence computes its relative density with respect to other lanes.

3.Time allotment module takes input (as traffic density) from this system and determines an optimized and efficient time slot.

4.This value is then triggered by the microprocessor to the respective Traffic Lights.

SampleOutput:

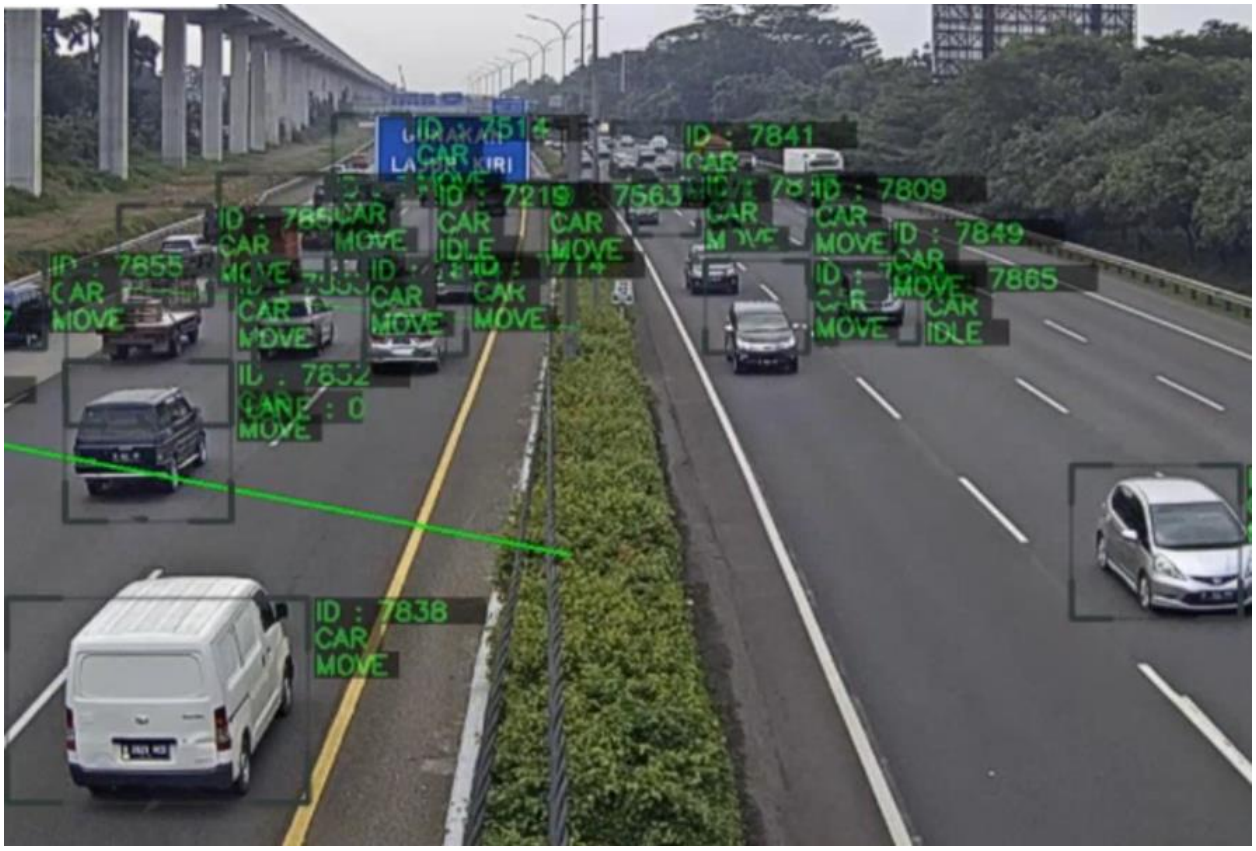
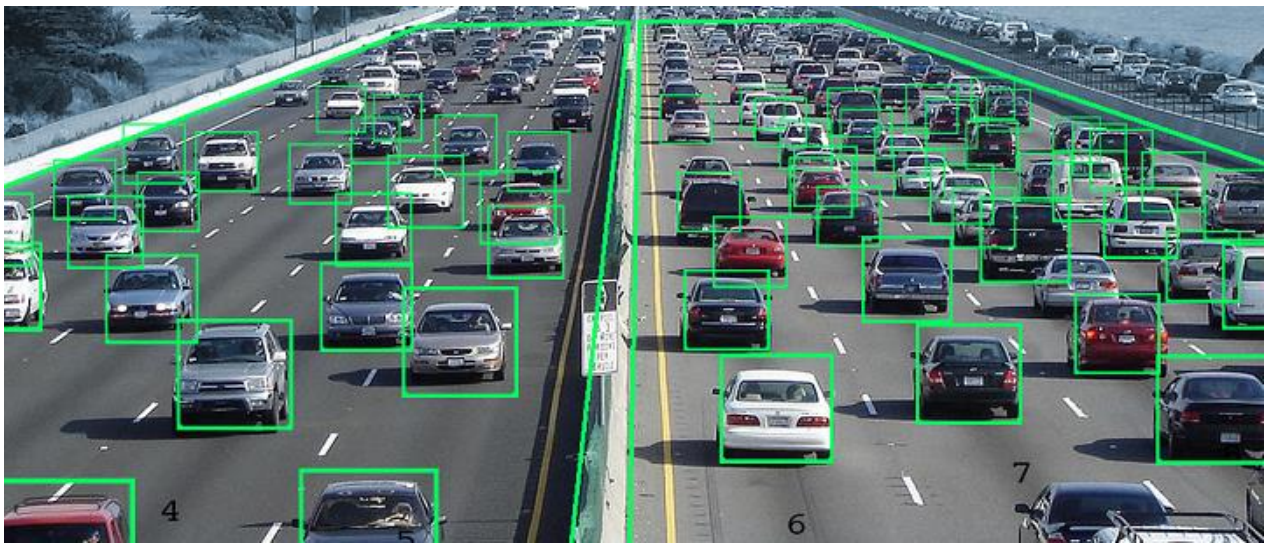


Fig: Object Detection for Traffic Management System



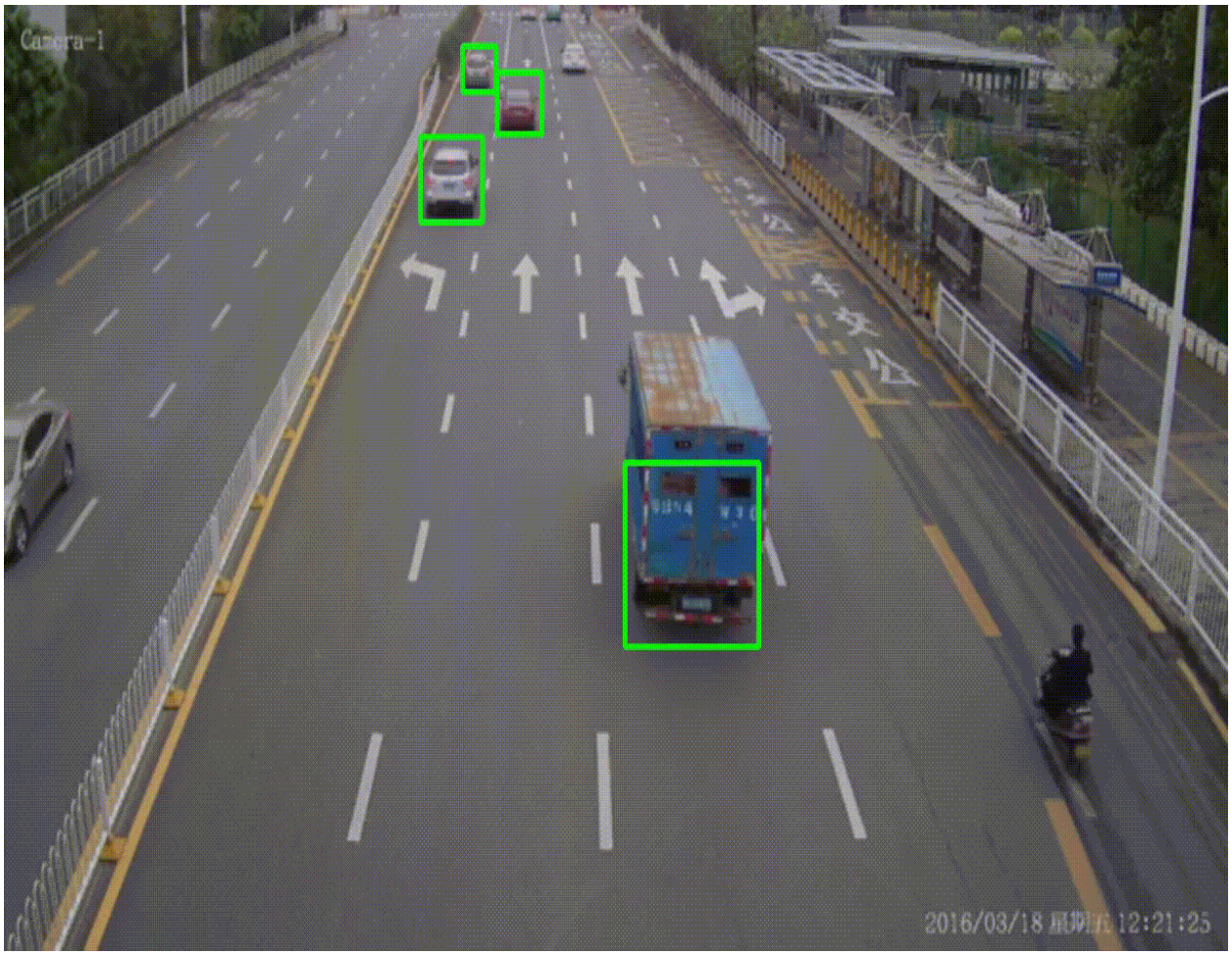


Fig: Object Detection for Traffic Management System