# **GROUP 2 : INTERNET OF THINGS(IOT)**

# TRAFFIC MANAGEMENT SYSTEM

# SMART TRAFFIC MANAGRMENT SYSTEM

## **Introduction :**

In this day & age, the conventional systems to manage urban mobility are proving incompetent. And there’s a growing need for an efficient traffic management system. Cities big and small are in dire need of technology-led digital solutions to manage & monitor traffic. They can help regulate heavy traffic, road blockages at signals & congested networks.

An Internet of Things (IoT)-enabled intelligent traffic management system can solve pertinent issues by leveraging technologies like wireless connectivity & intelligent sensors. Considered a cornerstone of a smart city, they help improve the comfort and safety of drivers, passengers & pedestrians.

**Key Contents   :**

* [Role of IoT in Smart City Traffic Management](https://www.rishabhsoft.com/blog/smart-traffic-control-using-iot" \l "role-of-iot)
* [Advantages of a Smart Traffic Management System](https://www.rishabhsoft.com/blog/smart-traffic-control-using-iot" \l "advantages)
* [Functioning of Traffic Monitoring System Using IoT Capabilities](https://www.rishabhsoft.com/blog/smart-traffic-control-using-iot" \l "functioning)
* [Application of IoT in Traffic Management](https://www.rishabhsoft.com/blog/smart-traffic-control-using-iot" \l "use-cases)
* [Key Features of a Smart Traffic Management System](https://www.rishabhsoft.com/blog/smart-traffic-control-using-iot" \l "features)
* [Implementation of a Smart Traffic Management System – Key Elements](https://www.rishabhsoft.com/blog/smart-traffic-control-using-iot" \l "implementation)

## **Role of IoT in Smart City Traffic Management :**

With cities worldwide experiencing ongoing population growth – it results in stressed municipal infrastructure. And the problem of traffic congestion across smart cities is continuously increasing. [INRIX suggests](https://inrix.com/press-releases/2021-traffic-scorecard/" \t "https://www.rishabhsoft.com/blog/_blank) that the average American driver lost 36 hours due to congestion, costing $564 in wasted time. This increasing growth in cities leads to the demand to meet sustainability goals while evaluating traffic management strategies.

Integrating innovative traffic technology helps achieve phenomenal cost savings in smart cities’ infrastructure expenses while improving system reliability. [Juniper research suggests](https://www.juniperresearch.com/press/smart-traffic-management-to-significantly-reduce" \t "https://www.rishabhsoft.com/blog/_blank) that smart traffic management systems could save cities $277 billion. It is while reducing emissions and congestion by 2025.

With the pressing demand for advanced communication & network technologies, digitalization is the driving force that stimulates the implementation of smart traffic control using IoT capabilities.

It enables them to;

* Expand the capacity of city streets without having to build new roads.
* Optimize the traffic flow and keep the drivers safe. It would include cameras, sensors, and cellular technologies that automatically adjust traffic lights, expressway lanes, speed limits, and highway exit counters.
* Transmit accurate information about available parking spaces to citizens in real-time
* Collect data on congestion and improve traffic signaling to reduce blockages and optimize commute
* Locate incidents and report them to emergency rooms immediately with road sensors and video surveillance
* Employ real-time data feeds to ensure the streetlights turn dim or brighten up per the changing weather conditions and the onset of day and night

## **Advantages of a Smart Traffic Management System :**

Cleaner, greener, safer, and more accessible roads are a few benefits of implementing IoT and intelligent technology.

It helps with the following:

* Reducing traffic jams and accidents on the streets
* Ensuring immediate clearance for emergency vehicles
* Facilitating safer and shorter commute times
* Reducing congestion & energy consumption at intersections
* Offering significant productivity benefits with real-time monitoring of crucial infrastructures
* Reducing operating costs with efficient traffic management processes
* Ensuring compliance with the regulations for reducing the carbon footprint
* Saving billions of gallons of fuel wasted every year
* Accurate tracking & quick recovery of lost and stolen vehicles

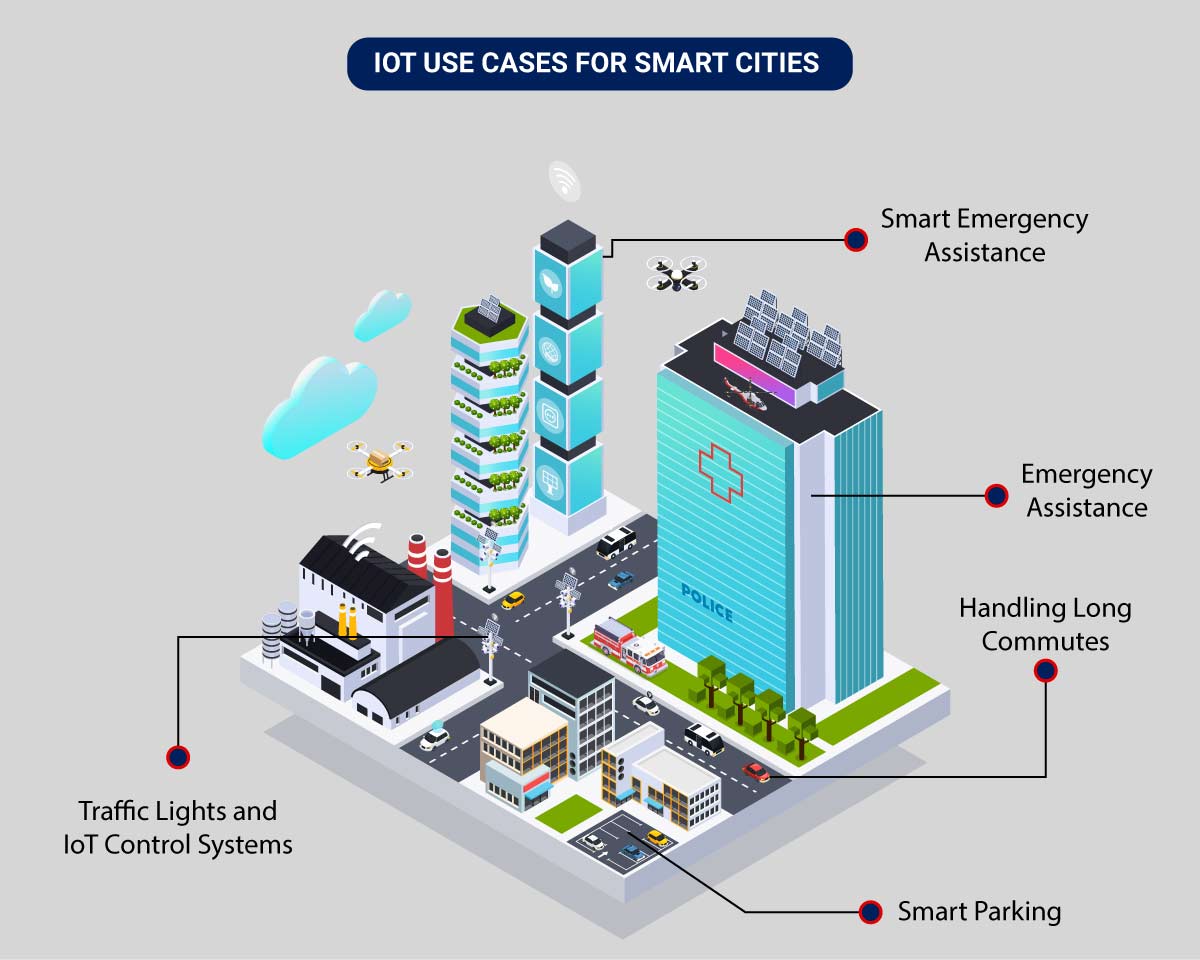
## **Functioning of Traffic Monitoring System Using IoT Capabilities :**

This intelligent system comprises several components, including wireless sensors, RFID tags, and BLE beacons installed at the traffic signals to monitor the movement of vehicles. A real-time data analytics tool connects the Geographic Information System (GIS-enabled) digital roadmap with control rooms for real-time traffic monitoring.

The smart traffic management system captures the images of vehicles at the signals using the digital image processing technique. This data is then transferred to the control room via wireless sensors. The system also leverages BLE beacons or RFID tags to track the movement of vehicles and keep traffic congestion in control, track down stolen vehicles and even clear the road for emergency vehicles that are installed with RFID readers.

## **Application of IoT in Traffic Management :**

City governments can improve their operations & infrastructure by placing IoT sensors and tracking devices on roads and highways for recording, analyzing, and sharing data in real-time.



An intelligent traffic monitoring system using IoT capabilities has so many factors & use cases, including;

* **Traffic Lights and IoT Control Systems**: Smart traffic signals may look like a typical stoplight, yet they utilize an array of sensors to monitor real-time traffic. Usually, the goal is to help cars reduce the amount of time spent idle. And IoT technology enables the various signals to communicate with each other. This is while adapting to changing traffic conditions in real time. The outcome is less time spent in traffic jams and even reduced carbon emissions.
* **Parking Enabled through IoT**: Smart meters and mobile apps make on-street parking spaces easily accessible with instant notifications. Drivers receive alerts whenever a parking spot is available to reserve it instantly. The app gives easy directions to the parking spot with a convenient online payment option.
* **Emergency Assistance through IoT**: A traffic monitoring system using IoT technology enables emergency responders to speed up the care mechanism in case of accidents late at night or in isolated locations. The sensors on the road detect any accident, and the problem is immediately reported to the traffic management system. This request is passed on to relevant authorities to take corrective action. Emergency response personnel would include medical technicians, police officers, and fire departments for enhanced responsiveness and timely intervention.
* **Commute Assistance:**With every vehicle acting as an IoT sensor, a dedicated app can make suggestions, determine optimal routes & provide advance notice of accidents or traffic jams. Further, it can even suggest the best time to leave. It is all because of a robust algorithm that helps reduce driving time with intelligent traffic lights.

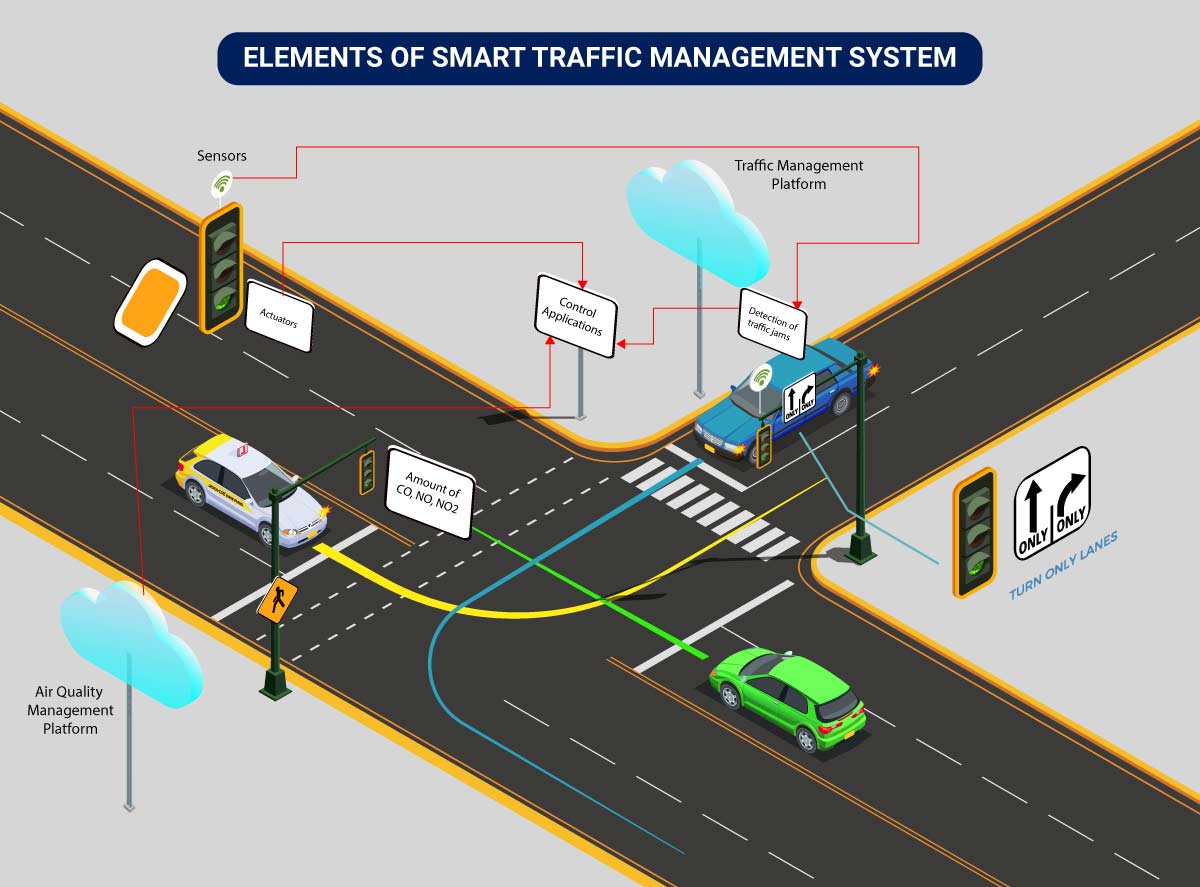
## **Key Features of a Smart Traffic Management System :**

The key features are listed below depending on the city’s size and the scope of the governmental policies. It can be integrated into an intelligent traffic management system. They include:

* **Traffic Jam Detection:**With cloud connectivity, sensors, and CCTV cameras tracking intersections 24×7, technicians can remotely monitor all the streets in real-time from the city’s traffic control room.
* **Connected Vehicles:**A smart traffic system using IoT technology can connect with roadside tracking devices to enable direct communication between intelligent vehicles & intersections.
* **Modular Control:**Real-time detection of congestion triggers dynamic adjustments in the systems meant for controlling traffic lights, express lanes, and entry alarms.
* **Emergency Navigation:**A system with edge data processing & programmatic alerting capabilities can alert response units (police, ambulance & tow trucks) in case of a car crash or collision. It reduces the crucial time an injured driver or passenger remains unattended.
* **Road Safety Analytics:** Systems with pattern detection capabilities can immediately flag high cruising speeds and reckless driver or inappropriate pedestrian behavior.
* **Digital Payments:** Commercial traffic management systems enable quick and convenient electronic transactions in real time while ensuring financial data safety.

## **Implementation of a Smart Traffic Management System – Key Elements :**

Whether municipalities want to improve their traffic management approach, expand public services, or upgrade existing infrastructure – it all starts with a smart city solution!



Here’s an implementation plan for building a scalable traffic control system using IoT capabilities:

A basic architecture that serves as a launchpad for feature enhancements and service upgrades will integrate the following components:

* **Sensors**for collecting data and sending it to a centralized cloud platform
* **Actuators**for physical devices to make necessary adjustments like – restricting the water supply in pipelines with leakages or dimming & brightening streetlights based on weather conditions.
* **Field gateways**to collect & compress data before moving it to a cloud platform.
* **Cloud gateways**enable secure data transfer between field gateways & the cloud storage of the traffic management system
* **A data lake**to store the raw, unstructured information before it is cleansed, processed, transformed & moved to a data warehouse for extracting actionable insights
* **Data warehouse**stores contextual information about connected objects and devices installed with sensors and actuators.
* **Data analytics**for analyzing the data from streetlight sensors on a centralized dashboard to adjust the intensity of lights
* **ML algorithms**to analyze traffic patterns & trends from historical data – stored in the data warehouse. The identified trends are then used to build predictive models for control apps. These apps modify the average vehicle speed to avoid congestion.
* **Rules**to enable actuators to automate the functioning & control of smart city objects and devices. These rules are manually defined to tell actuators what needs to be done to solve a specific problem.
* **User applications** that allow citizens to receive instant notifications in case of traffic jams and congested routes. Desktop user apps for control rooms send commands to actuators for altering traffic signals. It helps to relieve congestion and optimize routes.
* **Cross-solution integrations** with traffic lights or streetlight management systems. Control apps apply ML models or predefined rules to prompt appropriate output action if the air quality is poor.

Cities of all sizes can leverage this approach. Depending on the budgetary and procurement constraints, they can start small. It would be with solutions like – a littering offense ticketing system or a smart parking app. Later they can expand the range of service.

**Explanation :**

## ****Traffic Management System :****

****Traffic Management System**** python project source code – The number of vehicles has increased drastically in the last few decades making it difficult to monitor each and every vehicle for the ****traffic management system****and law enforcement purposes. We proposed a computer vision-based solution using deep learning that automatically detects traffic violators.

The main objective is to detect vehicles that do not follow the rules of traffic, ****like****overspeeding, overloading, not wearing a helmet, and running on the incorrect side of the road. We use Yolov3 for object detection and DeepSort for tracking vehicles and pedestrians. The system detects the sort of violation along with the vehicle information, maintains a log of violations, provides an in-depth dashboard, and provides alerts to the traffic police personnel. The logs ****also can****be used for forensic purposes.

## **Datasets and Models :**

I have used the dataset available on the internet for helmet images and trained the model, the [Model](https://drive.google.com/file/d/1P1klU95d1ltZq3Wfsu6gU02RIKoy6gre/view) is required to run the project. Kindly download it.

## **Working :**

I have done the detection of ****traffic management system**** & violations in real time using our own trained haar cascades for vehicle and pedestrians detection, our own trained helmet detection model written in YOLO v3, and got satisfactory results.

**Source code :**

**The main file to run the project…**

from time import sleep

import cv2 as cv

import argparse

import sys

import numpy as np

import os.path

from glob import glob

#from PIL import image

frame\_count = 0 # used in mainloop where we're extracting images., and then to drawPred( called by post process)

frame\_count\_out=0 # used in post process loop, to get the no of specified class value.

# Initialize the parameters

confThreshold = 0.5 #Confidence threshold

nmsThreshold = 0.4 #Non-maximum suppression threshold

inpWidth = 416 #Width of network's input image

inpHeight = 416 #Height of network's input image

# Load names of classes

classesFile = "obj.names";

classes = None

with open(classesFile, 'rt') as f:

classes = f.read().rstrip('\n').split('\n')

# Give the configuration and weight files for the model and load the network using them.

modelConfiguration = "yolov3-obj.cfg";

modelWeights = "yolov3-obj\_2400.weights";

net = cv.dnn.readNetFromDarknet(modelConfiguration, modelWeights)

net.setPreferableBackend(cv.dnn.DNN\_BACKEND\_OPENCV)

net.setPreferableTarget(cv.dnn.DNN\_TARGET\_CPU)

# Get the names of the output layers

def getOutputsNames(net):

# Get the names of all the layers in the network

layersNames = net.getLayerNames()

# Get the names of the output layers, i.e. the layers with unconnected outputs

return [layersNames[i-1] for i in net.getUnconnectedOutLayers()]

# Draw the predicted bounding box

def drawPred(classId, conf, left, top, right, bottom, frame):

global frame\_count

# Draw a bounding box.

#cv.rectangle(frame, (left, top), (right, bottom), (255, 178, 50), 3)

label = '%.2f' % conf

# Get the label for the class name and its confidence

if classes:

assert(classId < len(classes))

label = '%s:%s' % (classes[classId], label)

#Display the label at the top of the bounding box

labelSize, baseLine = cv.getTextSize(label, cv.FONT\_HERSHEY\_SIMPLEX, 0.5, 1)

top = max(top, labelSize[1])

#print(label) #testing

#print(labelSize) #testing

#print(baseLine) #testing

label\_name,label\_conf = label.split(':') #spliting into class & confidance. will compare it with person.

if label\_name == 'Helmet':

#will try to print of label have people.. or can put a counter to find the no of people occurance.

#will try if it satisfy the condition otherwise, we won't print the boxes or leave it.

#cv.rectangle(frame, (left, top - round(1.5\*labelSize[1])), (left + round(1.5\*labelSize[0]), top + baseLine), (255, 255, 255), cv.FILLED)

#cv.putText(frame, label, (left, top), cv.FONT\_HERSHEY\_SIMPLEX, 0.75, (0,0,0), 1)

frame\_count+=1

#print(frame\_count)

if(frame\_count> 0):

return frame\_count

# Remove the bounding boxes with low confidence using non-maxima suppression

def postprocess(frame, outs):

frameHeight = frame.shape[0]

frameWidth = frame.shape[1]

frame\_count\_out=0

classIds = []

confidences = []

boxes = []

# Scan through all the bounding boxes output from the network and keep only the

# ones with high confidence scores. Assign the box's class label as the class with the highest score.

classIds = [] #have to fins which class have hieghest confidence........=====>>><<<<=======

confidences = []

boxes = []

for out in outs:

for detection in out:

scores = detection[5:]

classId = np.argmax(scores)

confidence = scores[classId]

if confidence > confThreshold:

center\_x = int(detection[0] \* frameWidth)

center\_y = int(detection[1] \* frameHeight)

width = int(detection[2] \* frameWidth)

height = int(detection[3] \* frameHeight)

left = int(center\_x - width / 2)

top = int(center\_y - height / 2)

classIds.append(classId)

#print(classIds)

confidences.append(float(confidence))

boxes.append([left, top, width, height])

# Perform non maximum suppression to eliminate redundant overlapping boxes with

# lower confidences.

indices = cv.dnn.NMSBoxes(boxes, confidences, confThreshold, nmsThreshold)

count\_person=0 # for counting the classes in this loop.

for i in indices:

i = i[0]

box = boxes[i]

left = box[0]

top = box[1]

width = box[2]

height = box[3]

#this function in loop is calling drawPred so, try pushing one test counter in parameter , so it can calculate it.

frame\_count\_out = drawPred(classIds[i], confidences[i], left, top, left + width, top + height, frame)

#increase test counter till the loop end then print...

#checking class, if it is a person or not

my\_class='Helmet' #======================================== mycode .....

unknown\_class = classes[classId]

if my\_class == unknown\_class:

count\_person += 1

#if(frame\_count\_out > 0):

#print(frame\_count\_out)

if count\_person >= 1:

path = 'test\_out/'

# frame\_name=os.path.basename(fn) # trimm the path and give file name.

#cv.imwrite(str(path)+frame\_name, frame) # writing to folder.

#print(type(frame))

#cv.imshow('img',frame)

#cv.waitKey(800)

return 1

else:

return 0

#cv.imwrite(frame\_name, frame)

#======================================mycode.........

# Process inputs

winName = 'Deep learning object detection in OpenCV'

cv.namedWindow(winName, cv.WINDOW\_NORMAL)

def detect(frame):

#frame = cv.imread(fn)

frame\_count =0

# Create a 4D blob from a frame.

blob = cv.dnn.blobFromImage(frame, 1/255, (inpWidth, inpHeight), [0,0,0], 1, crop=False)

# Sets the input to the network

net.setInput(blob)

# Runs the forward pass to get output of the output layers

outs = net.forward(getOutputsNames(net))

# Remove the bounding boxes with low confidence

# Put efficiency information. The function getPerfProfile returns the overall time for inference(t) and the timings for each of the layers(in layersTimes)

t, \_ = net.getPerfProfile()

#print(t)

label = 'Inference time: %.2f ms' % (t \* 1000.0 / cv.getTickFrequency())

#print(label)

#cv.putText(frame, label, (0, 15), cv.FONT\_HERSHEY\_SIMPLEX, 0.5, (0, 0, 255))

#print(label)

k=postprocess(frame, outs)

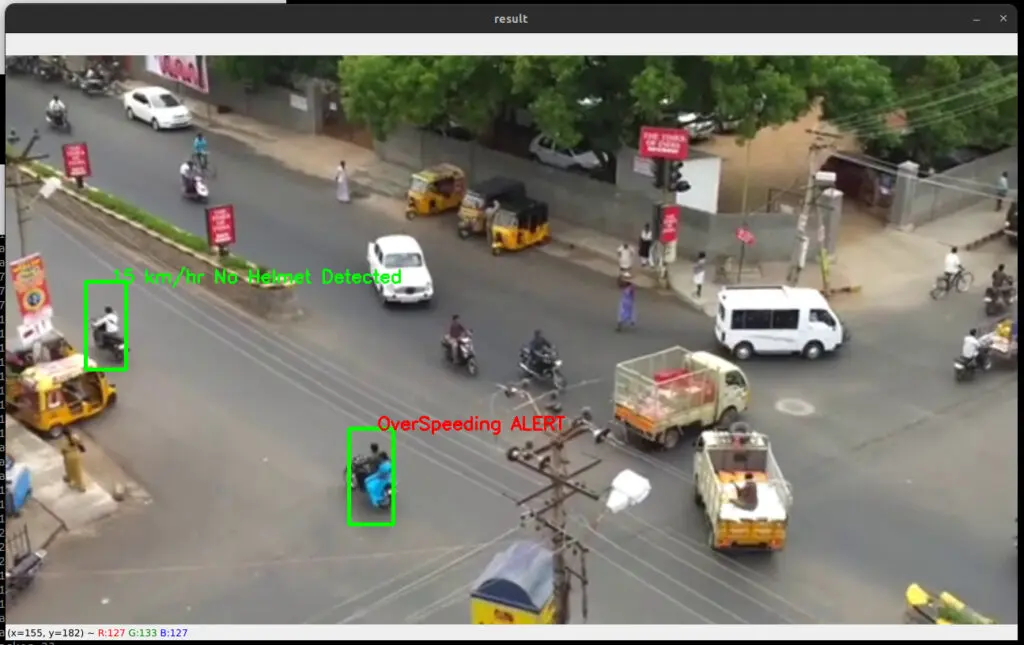
if k:

return 1

else:

return 0

## **Output :**



## **INTRODUCTION :**

The intelligent Traffic Expert Solution for road traffic control System offers the ability to acquire real-time traffic information. Traffic Expert enables operators to perform real-time data analysis on the information gathered. Traffic management measures are aimed at improving the safety and flow of traffic utilizing traffic capacity more effectively.

## **PURPOSE :**

## Smart Traffic Management is mainly improvised for looking after the Set off data of a region to manage the Traffic along that area and implement various useful technologies which are been required by various persons like vehicle owners, pedestrians, police officers etc….Mainly the purpose of Smart traffic management system is to give the details which can be used and they can be implemented in their daily life. The problems which have been occurred at their presence can be solved by this Smart Traffic.

## **SCOPE :**

Smart Traffic is a Video Analytics Module and provides ****Traffic Incident Detection****, and real time ****Traffic Flow Metrics**** & statistical analysis. Smart ****Traffic Monitoring**** can integrate with third party ****traffic management**** and ****smart roadway systems****and hosts a feature rich product scope itself. The system can be used for incident detection or for statistical metrics of a roadway.

****Related work :****

* Traffic lights, traffic signals, stoplights or robots are signalling devices positioned at ****road intersections, pedestrian crossings****, and other locations to control flows of traffic.
* The world’s first traffic light was a manually operated gas lit signal installed in ****London****in December 1868. It exploded less than a month after it was implemented, injuring its policeman operator. ****Earnest Sirrine**** from Chicago patented the first automated traffic control system in1910. It used the words “STOP and “PROCEED”, although neither word was illuminated.
* Traffic lights followed a universal ****color code**** which alternates the ****right of way**** accorded to users with a sequence of illuminating lamps or LEDs of three standard colours.
* ****GREEN light****: allows traffic to proceed in the direction denoted, if it is safe to do so and there is room on the other side of the intersection.
  + ****RED light****: prohibits any traffic from proceeding. A flashing red indication requires traffic to stop and then proceed when safe.
  + ****YELLOW light****: warns that the signal is about to change to red, with some jurisdiction requiring drivers to stop if it is safe to do so and others allowing drivers to go through the intersection if safe to  do so.

## **ADVANTAGES :**

* ****TRAFFIC CONTROL****
  + Existing centralised traffic control system go someway towards alleviating traffic congestion and ensuring the smooth flow of vehicles through a road network.
  + Intelligent transportation systems, however, allow traffic lights to respond to changing pattern themselves.
* ****TIME SAVING****
  + At certain junctions, sometimes even if there is no traffic, people have to wait. Because the traffic light remains red for the preset time period, the road users should wait until the light turn to green.
  + In smart traffic management system, if there is no vehicle present on a particular lane, then we can bypass through that lane, thereby saving our time.
* ****DETECTION AND MANAGEMENT OF TRAFFIC CONGESTION****
  + In addition to the earlier method of traffic congestion detection, a server can be maintained which can receive certain crucial data such as the density of vehicles present on a particular lane.
  + The main aim is to implement a system that would trace the travel time of individual cars as they pass and if congestion is sensed then system would control traffic signals and generate automatic re-routing of vehicles.

**SOURCE CODE :**

#Function to simulate a traffic light#It is required to make 2 user defined functions trafficLight() and light().def trafficLight():

signal = input("Enter the colour of the traffic light: ")

if (signal not in ("RED","YELLOW","GREEN")):

print("Please enter a valid Traffic Light colour in CAPITALS")

else:

value = light(signal) #function call to light()

if (value == 0):

print("STOP, Your Life is Precious.")

elif (value == 1):

print ("PLEASE GO SLOW.")

else:

print("GO!,Thank you for being patient.")#function ends here

def light(colour):

if (colour == "RED"):

return(0);

elif (colour == "YELLOW"):

return (1)

else:

return(2)#function ends here

trafficLight()print("SPEED THRILLS BUT KILLS")

#### **Logic:**

* Define the function trafficLight().
* Prompt the user to enter the color of the traffic light.
* If the input is not “RED”, “YELLOW”, or “GREEN”, display an error message.
* If the input is valid, call the function light() with the input as the argument.
* Store the return value from the function light() in the variable “value”.
* Based on the value of “value”, display the appropriate message.
* Define the function light() which takes the color as an argument.
* If the color is “RED”, return 0.
* If the color is “YELLOW”, return 1.
* For any other color, return 2.
* Call the function trafficLight().
* Print the safety message “SPEED THRILLS BUT KILLS”.

#### **Output:**

* >> Enter the colour of the traffic light: RED
* >> STOP, Your Life is Precious.
* >> SPEED THRILLS BUT KILLS

CONCLUSIONS :

In this way we are developing a very smart traffic control system which can be able to detect and monitor the traffic. It can take decision according to the density of traffic. The proposed work guarantees that it will give an efficient and dynamic management of traffic considering emergency vehicles. The speed detection system implemented can help reduce accidents and hence save lives.