PROJECT BASED LEARNING LAB
(ECE)

B.TECH V-Semester (IARE-R18)

By,
Dr. B. Surekha Reddy, Assistant Professor, ECE

Table of content:

Title:

TRAFFIC CONTROL SYSTEM USING OPENCV

Abstract:

- In-country like India, billions of people start and end each working day stuck in traffic or commuting on congested trains and buses. To overcome the problems of traditional traffic control systems, there is a shift in adaption to an Adaptive traffic control system. This proposed model of camera-based traffic monitoring and processing system which reduces the cycle time and possesses special provisions for vehicles.

- Keywords— OpenCV, Traffic monitoring; Traffic congestion detection; Traffic Management System

Introduction:

- Congestion of traffic has been a matter of routine in any existing metropolitan region with an overwhelming population but minimal infrastructure. Traffic Congestion is a critical problem with dire causes and consequences on the road. Radical population growth and low public quality transportation have caused vehicles to expand massively. Poorly controlled traffic, apart from infrastructure, creates congestion that could survive for hours. Only to a certain degree can a pre-defined timing scheme for traffic control ease the problem. It must surpass its value as a fixed delay unit irrespective of the traffic volume on the pre-defined timer signal that leads to more traffic build-up on other lanes of the intersections. So our proposed model reduce the time complexity of operation and based on detection of density of vehicles going to perform the traffic signal operations .

Applications :

- Traveler information services
- Emergency management services
- Vehicle safety
- Control system
- Public transport services

Advantages and Disadvantages

- Pros:
- The traffic controlling get easier
- Emergency vehicles
- Vehicle detection etc.
- Cons:
- Failure of camera system
- The accuracy may not appropriate

Literature survey:

1. Real-Time Adaptive Traffic Control System For Smart Cities
2. Density Based Smart Traffic Control System for Congregating Traffic Information
3. Optimal LQG Control of Networked Systems Under Traffic-Correlated
4. Two-Level Hierarchical Model-Based Predictive Control for Large-Scale Urban Traffic Networks
5. Urban Traffic Cooperative Control based on Regional Division

Problem statement and objectives:

- PROBLEM STATEMENT:
  - If the density gets heaver the operation may difficult to perform.
  - The manual operation can not be accurate
  - And the density detection using OpenCV may not be suitable for all cases
- OBJECTIVE:
  - Our main objective of this project is to detect the density of the vehicles in the particular direction where the density is more and perform the manual operation to overcome the traffic problems and also to it is use full for the emergency vehicles to ger exist at faster from the traffic. The applications like emergency vehicles , object detection and tracking can be performed with this proposed method.

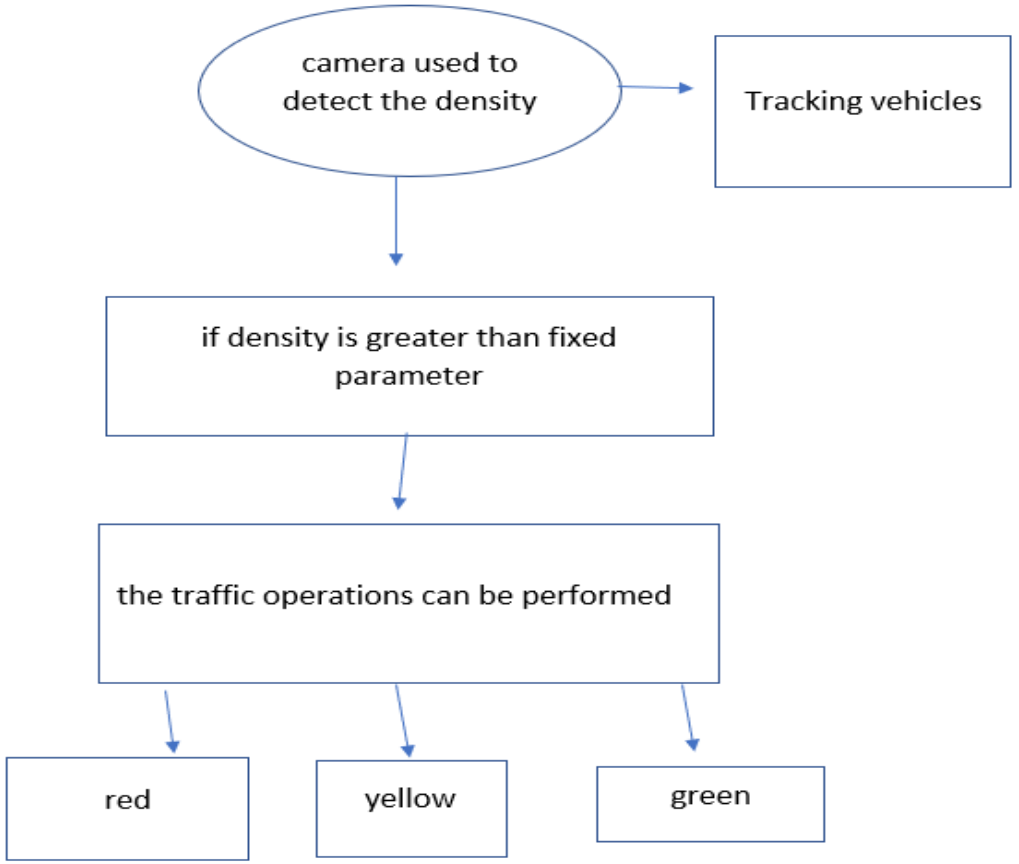
Existing model and method:

- IOT BASED AUTOMATED TRAFFIC LIGHT CONTROL SYSTEM LORA METHOD:
- EXISTING METHOD:
- In India the siren sound of all emergency vehicles are preset and follows a similar pattern. The siren sound repeats in two tones. The tones are 960 Hz and 770 Hz, and these are repeated at every 1.3 sec period. The siren sound is affected by the Doppler Effect and varies its frequency due to the motion of the emergency vehicle. The proposed system works in two phases. First phase is about detection of emergency vehicle and second phase is all about taking the action at the intersection. The system uses the sound detection sensor, camera and microcontroller for processing the data. The proposed system uses the LoRa technology for communication. Data set of different emergency vehicle patterns will be stored at the smart object, which will be used to compare the current emergency vehicle with the existing dataset. Camera will be installed in the smart object and will be well positioned to capture only required portion of the road. In first phase the smart object detects the emergency vehicle on the road through: If the emergency vehicle is on the way towards the signal then the smart object which is placed (200 m) away from the signal junction will detect the siren sound of emergency vehicle by using sound detection sensor. Next process in the smart object is about matching the moving object on the road with the stored dataset. The camera will be set to capture the pictures of vehicles on the road as soon as the Smart object detects the sound. If both the conditions satisfy, then smart object sends the message to the Decision Support System which is centralized in the Signal junction. Second phase is about taking the decision. Signal junction will be installed with the Decision Support System. This system receives the signals from the smart objects which are placed on the different roads which are going to intersect in the junction. All the smart objects and Decision Support System will be arranged in the star topology. The Decision Support System at the center is responsible for taking the appropriate decision about clearing the traffic on the lane where the emergency vehicle is travelling. Decision Support System will be installed with Acoustic Sensors near the intersection which works on Receding Doppler Effect, to make sure that the emergency vehicle has crossed the junction so it can be reverted back to its normal functioning. Decision Support System is also responsible for receiving the data, processing the data, storing the data to the cloud, as well as transmitting the data to the next Decision Support System.

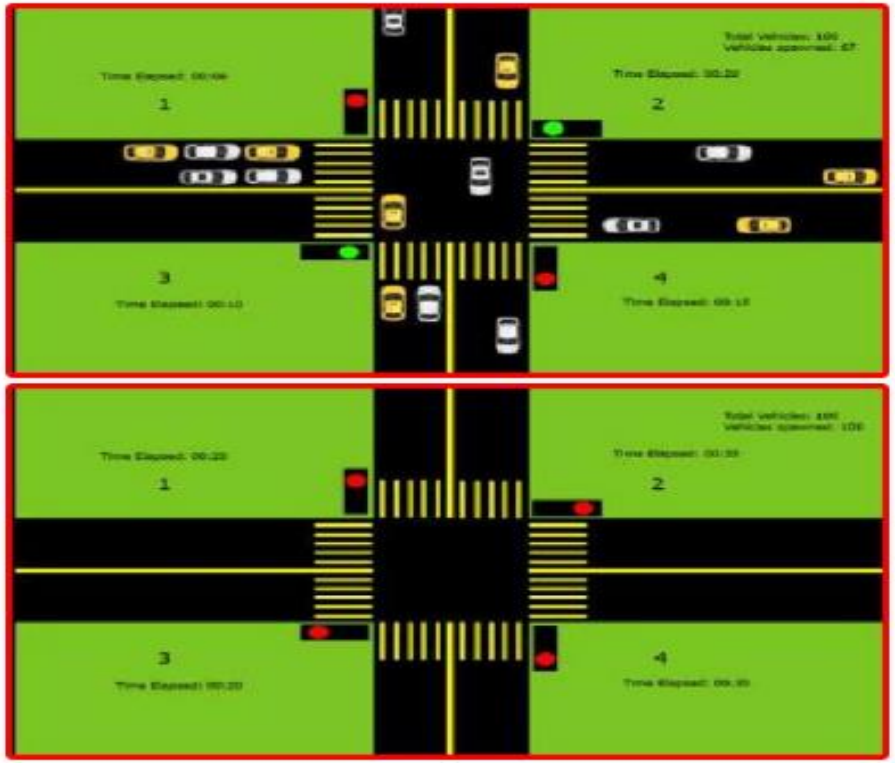
Proposed method

- This proposed model of traffic control system using opencv monitoring based on camera and processing system which reduces the cycle time and possesses special provisions for emergency vehicles.
- EQUIPMENT REQUIRED:
- Camera
- Python software
- Python library opencv

Flow chart:

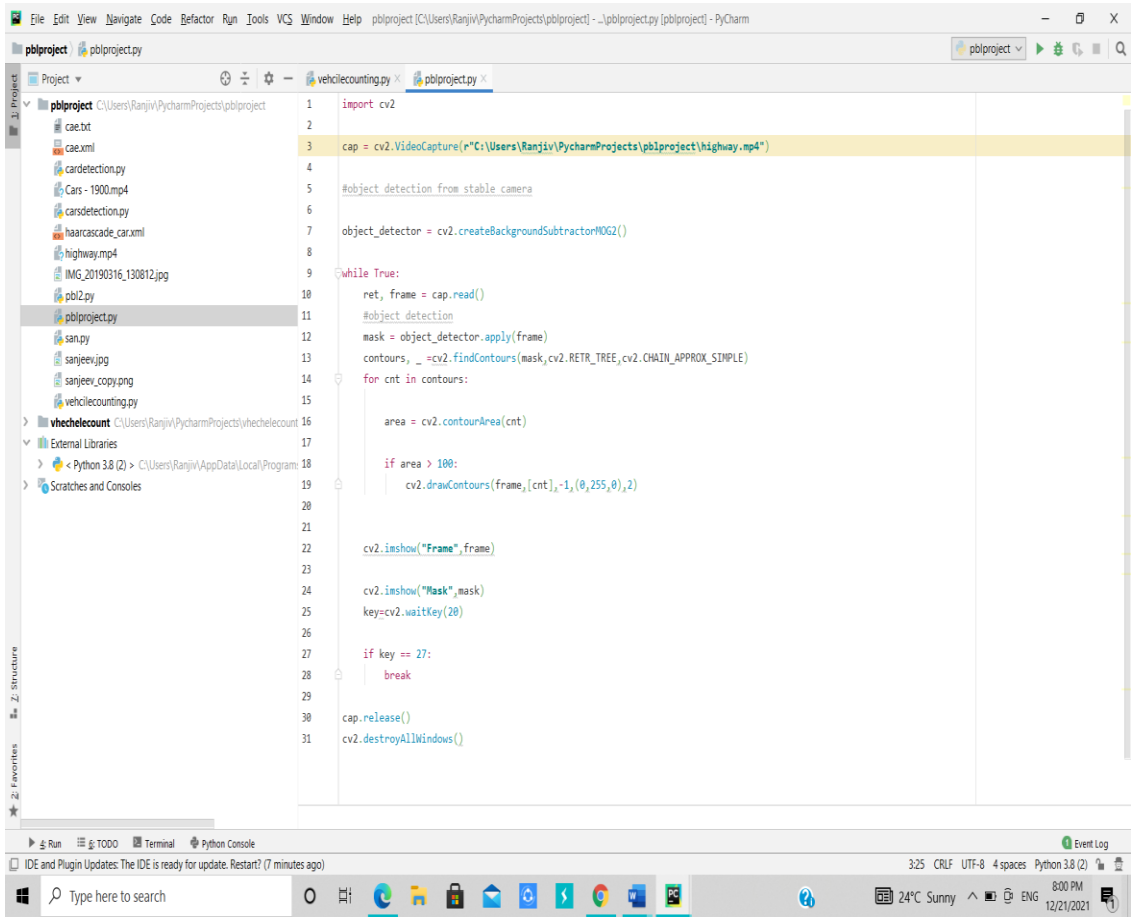

camera used to detect the density
Tracking vehicles
if density is greater than fixed parameter
the traffic operations can be performed
red
yellow
green

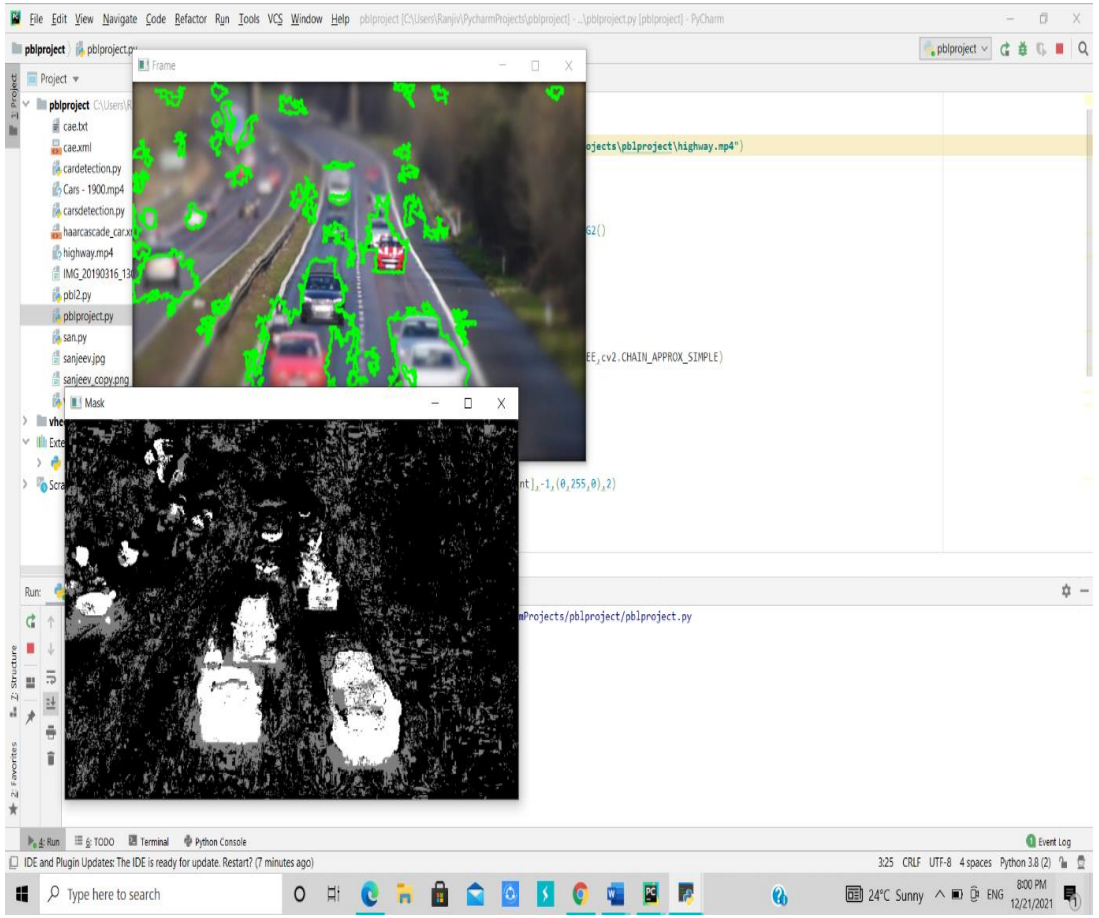
Working:



The control algorithm is intended to control 4-way and 2-way approach intersection.The video feed is obtained via the automated operations supported by OpenCV using the Direct Display (Dshow) techniques.The classifier for the identification of objects, which in this case are vehicles, has been created using the Open CV development kit. The cascaded classifier, consisting of several stages of other simple classifiers, uses Haar-like features to detect objects.based upon that the traffic signals is operated.

Result :


import cv2
cap = cv2.VideoCapture(r"C:\Users\Ranjiv\PycharmProjects\pblproject\highway.mp4")
#object detection from stable camera
object_detector = cv2.createBackgroundSubtractorMOG2()
while True:
ret, frame = cap.read()
#object detection
mask = object_detector.apply(frame)
contours, _ =cv2.findContours(mask,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
for cnt in contours:
area = cv2.contourArea(cnt)
if area > 100:
cv2.drawContours(frame,[cnt],-1,(0,255,0),2)
cv2.imshow("Frame",frame)
cv2.imshow("Mask",mask)
key=cv2.waitKey(20)
if key == 27:
break
cap.release()
cv2.destroyAllWindows()


Frame
Mask

Conclusion:

- So,This proposed model of camera-based traffic monitoring and processing system which reduces the cycle time and possesses special provisions for vehicles and traffic control can be done easier and faster way.

Appendix :

```
import cv2

# capture frames from a video
cap = cv2.VideoCapture(r'Cars - 1900.mp4')

# Trained XML classifiers describes some features of some object we want to detect
car_cascade = cv2.CascadeClassifier(r'C:\Users\Ranjiv\PycharmProjects\pblproject\cae.xml')

# loop runs if capturing has been initialized.
while True:
    ret, frames = cap.read()
    gray = cv2.cvtColor(frames, cv2.COLOR_BGR2GRAY)
    cars = car_cascade.detectMultiScale(gray, 1.1, 1)
    for (x, y, w, h) in cars:
        cv2.rectangle(frames, (x, y), (x+w, y+h), (0, 0, 255), 2)
    cv2.imshow('sKSama', frames )
    if cv2.waitKey(33) == 27:
        break

# De-allocate any associated memory usage
cv2.destroyAllWindows()
```

Future scope:

- For increases in population this method can be easily adopted and the modifications can also be done future more.
- For emergency vehicles , tracking and counting and traffic signalling ,number plates detecting