

---

# **TRAFFIC ANOMALY INTRUSION DETECTION SYSTEM**

Submitted in partial fulfillment of the requirements for the award of  
Bachelor of Technology degree in Information Technology

by

Anand Kumar Ranjan (37120008)  
Md. Farhan (37120701)



**DEPARTMENT OF INFORMATION TECHNOLOGY  
SCHOOL OF COMPUTING**

**SATHYABAMA**  
INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)  
Accredited with Grade "A" by NAAC

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

**MARCH – 2021**



# **SATHYABAMA**

INSTITUTE OF SCIENCE AND TECHNOLOGY

**(DEEMED TO BE UNIVERSITY)**

Accredited with "A" grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

[www.sathyabama.ac.in](http://www.sathyabama.ac.in)



## **DEPARTMENT OF INFORMATION TECHNOLOGY**

### **BONAFIDE CERTIFICATE**

This is to certify that this Project Report is the bonafide work of ANAND KUMAR RANJAN (37120008), MD. FARHAN (37120701) under my supervision from November 2019 to April 2020.

#### **Internal Guide**

Dr.S.GOWRI,M.E.,Ph.D.,

#### **Head of the Department**

Dr. R. SUBHASHINI M.E., Ph.D.,

**Submitted for Viva voce Examination held on \_\_\_\_\_**

**Internal Examiner**

**External Examiner**

## **DECLARATION**

I ANAND KUMAR RANJAN (37120008), MD. FARHAN (37120701) here by declare that the Project Report entitled TRAFFIC ANOMALY DETECTION SYSTEM done by me under the guidance of Dr.S.GOWRI,M.E.,Ph.D.,(Internal) is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering / Technology degree in Information Technology.

**DATE:**

**PLACE:**

**SIGNATURE OF THECANDIDATE**

## ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D, Dean**, School of Computing , **Dr.R.Subhashini M.E., Ph.D. , Head** of the Department of Information Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.S.GOWRI,M.E.,Ph.D.**, for his valuable guidance, suggestions and constant encourage me ntpaved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

## **ABSTRACT**

Managing traffic rule violations has always been a tedious and compromising task. This system contains a camera that has the capability to grab an image and to find location of number in the image and then extract characters from character recognition tool. Since motorcycles are affordable and a daily mode of transport rules must be strict in order to avoid any accidents. As helmet is mandatory as per traffic rules violation of which attract heavy fines. This system presents Automatic Number Plate Recognition (ANPR) techniques and other image manipulation techniques for plate localization and character recognition which makes it faster and easier to identify the number plates. After recognizing the vehicle number from number plate the SMS based module is used to notify the vehicle owners about their traffic rule violation. An additional SMS is sent to Regional Transport Office (RTO) for tracking the report status.

## TABLE OF CONTENT

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	vi
	LIST OF FIGURE	ix
	LIST OF ABBREVIATIONS	x
1	<b>INTRODUCTION</b>	<b>1</b>
	1.1 ABOUT DOMAIN	1
	1.2 OVERVIEW	2
	1.3 OBJECTIVES	3
	1.4 PROBLEM STATEMENT	3
	<b>1.4.1 Existing System</b>	<b>4</b>
	<b>1.4.2 Proposed System</b>	<b>5</b>
2	<b>LITERATURE SURVEY</b>	<b>5</b>
3	<b>SYSTEM REQUIREMENTS</b>	<b>8</b>
	2.1 FUNCTIONAL REQUIREMENTS:	8
	2.2 NON-FUNCTIONAL REQUIREMENTS:	8
	2.4 HARDWARE REQUIREMENTS	9
	2.4 SOFTWARE REQUIREMENTS	9
4	<b>METHODOLOGY:</b>	<b>10</b>
	4.1 SYSTEM ARCHITECTURE	10
	4.2 STEPS :	11
	4.3 ALGORITHM USED	15
	4.4 USED CASE DIAGRAM	19
	4.5 CLASS DIAGRAM	19
5	<b>RESULT AND DISCUSSION</b>	<b>21</b>
	5.1 RESULT AND DISCUSSION	21

<b>6</b>	<b>CONCLUSION AND FUTURE WORK</b>	<b>23</b>
	6.1 FUTURE WORK	23
	<b>REFERENCES</b>	<b>24</b>
	<b>APPENDIX</b>	
	A.SOURCE CODE	25
	B.SAMPLE SCREEN	30

## LIST OF FIGURES

<b>SNO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
4.1	Architecture of Traffic Intrusion System	10
4.2	Analysis Model of Traffic Intrusion System	11
4.3	Vehicle detection	13
4.4	Traffic movements at junctions (tmj)	14
4.5	Image Segmentation	16
4.6	Gray Scaling	16
4.7	Canny Edge Image	17
4.8	An object detection model	18
4.9	use case diagram	19
4.10	class diagram	20

## LIST OF ABBREVIATIONS

### ACRONYM

### EXPANSION

ANPR	Automatic Number Plate Recognition(ANPR)
CNN	Convolutional neural network
ML	Machine learning
BPA	Business process automation
TMJ	Traffic Movements at junctions
OCR	Optical character recognition
UML	Unified Modeling Language
API	Application programming Interface
IDE	Integrated Development Environment
OS	Operating System

# CHAPTER 1

## INTRODUCTION

Automation in day to day life has gained importance in recent years. The number of accidents on the roads is due to the rule violations such as breaking traffic signals, over speeding, driving on wrong sides etc. To avoid such traffic violations, traffic police has to be present on the road and has to continuously check if some vehicle is violating the rule. A certain automated solutions were developed to eliminate the violations; however each of them had certain limitations. For example, the video capturing cameras eliminated need of a authority to be present to check rule violation. Managing traffic rule violations has always been a tedious and compromising task. This system presents Automatic Number Plate Recognition (ANPR) techniques and other image manipulation techniques for plate localization and character recognition which makes it faster and easier to identify the number plates. After recognizing the vehicle number from number plate the SMS based module is used to notify the vehicle owners about their traffic rule violation. An additional SMS is sent to Regional Transport Office (RTO) for tracking the report status. In this project number plate detection is simply the ability to automatically extract and recognition a vehicle number plates characters from the image. This system contains a camera that has the capability to grab an image and to find location of number in the image and then extract characters from character recognition tool. Since motorcycles are affordable and a daily mode of transport rules must be strict in order to avoid any accidents. As helmet is mandatory as per traffic rules violation of which attract heavy fines. This system using convolutional neural network(CNN) instead of hand crafted feature, to improve the classification performance.

### 1.1 ABOUT DOMAIN

Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values. Recommendation engines are a common use case for machine learning. Other popular uses include fraud detection, spam filtering,

malware threat detection, business process automation (BPA) and predictive maintenance.

## 1.2 OVERVIEW

Automation in day to day life has gained importance in recent years. The number of accidents on the roads is due to the rule violations such as breaking traffic signals, over-speeding, driving on wrong sides etc. To avoid such traffic violations, traffic police has to be present on the road and has to continuously check if some vehicle is violating the rule. A certain automated solutions were developed to eliminate the violations; however each of them had certain limitations. For example, the video capturing cameras eliminated need of a authority to be present to check rule violation. However, whole stored video had to be checked manually for the rule violation scenario. In this proposed system, a solution for signal breaking violation is given .The system includes an automated system by using IR sensor, camera and number plate recognition application. In this system IR sensor will be placed near zebra crossing line. If any vehicle crosses the zebra line, the desktop application will be initiated and will capture number plate image. Number plate recognition application by using image processing algorithm will recognize number plate and SMS will be sent to the offender in case of rule violation scenario. Solid In all the countries there are driving rules available for people to drive carefully by being honest and respecting. When these rules are broken it is defined as a road violation. There road violations that happens in day to day traffic. For an example most common violations such as red light violation, over-speeding, overtaking other vehicles through double white lines and single white lines. These road protocols are applied thoroughly for the places where most of the accidents happen. Research team have implemented an Open CV and python based system upon a PC . This system is implemented to ease the work to the police using a user interface. Image processing technology is used to detect the lanes, vehicles and will identify vehicles who break rules. The team have implemented the system with the ultrasonic sensors in order to system to receive conditions to identify when a violation occur. It will be easier for the police to catch who break laws of the traffic and for the policemen who take bribes and dishonest

people will also be reduced. An image of the violation with the location, time, date and an image of the vehicle to the nearest policemen devices by deriving devices around the current location that violation happened.

### 1.3 OBJECTIVES

- Main aim of this project is to replace the human traffic police by a virtual traffic police.
- It basically requires a 360-degree camera for surveillance. • It captures the vehicles which is violating the traffic rules by detecting the vehicles number plate.
- It also identifies if the driver is wearing the helmet or not for motorcyclist.

### 1.4 PROBLEM STATEMENT

To design and develop a real-time detection, tracking and license plate recognition system that will work efficiently under the conditions of slow-moving objects and the objects that are merged into the background due to a temporary stop and becoming foreground again, adaptive to different traffic environment conditions, robustness against progressive or sudden illumination changes, Occlusions, identification time of the system should be as short as possible.

#### **1.4.1 Existing System**

The first approach that relays on human participation is the traffic patrols. A traffic patrol is a police unit created primarily for the purpose of overseeing and enforcing traffic safety compliance on roads and highways. Another technology to fight the traffic violations is by using cameras installed on the traffic lights. The goal is just to detect the red-light crossing violations.

Drawbacks of existing system

- There was no proper implementation
- Accuracy was much low

### **1.4.2 Proposed System**

Without police also the camera will check the condition. Violation Detection using CNN Algorithm with respect to helmet, Signal Jumping, No parking. If any traffic violation occurs immediately send alert message to owner number. Number plate recognition for each traffic violation.

#### Advantages

- Saves traffic police time
- Makes people to follow strict traffic rules

## **CHAPTER 2**

### **LITERATURE SURVEY**

Year : 2018

Author : Amirgaliyev Beibut, Kairanbay Magzhan, Kenshimov Chingiz

Title : Effective Algorithms and Methods for Automatic Number Plate Recognition

Advanced systems for tracking and identifying stolen, unauthorized vehicles are based on automated number plate recognition technology. Genetic and HOUGH algorithms are used. The car number plate pictures were taken from different sides and the accuracy is 90%.The distance from camera to the vehicle and the weather conditions decreases the performance of the system.

Year : 2017

Author : Amirgaliyev Beibut, Kairanbay Magzhan, Kenshimov Chingiz

Title : An Efficient FPGA Implementation of Optical Character Recognition for License Plate Recognition.

The OCR is the core of an LPR system. In this work, a hardware-based OCR system has been implemented. Supervised learning algorithm is used. A robust FPGA based OCR system has been designed and tested with imperfect and noisy license plate images. It was able to maintain a 98.2% accuracy. It requires a proper system with selection of network size, and the optimized and efficient setup of the neuron activation function.

Year : 2015

Author : Farid Bounini, Denis Gingras, Herve Pollart

Title : Autonomous Vehicle And Real Time Road Lanes Detection And Tracking.

Work deals with a road boundaries and painted lines detection for intelligent and autonomous vehicles. Advanced computer vision algorithm used. Robust against exogenous perturbations and different constraints, but good enough to control the vehicle with a simple couple of fuzzylogic laws. The fuzzy controller handles the vehicle's steering, which has a limitation for a maximum speed of 70 km/h in sharp turns.

Year : 2019

Author : P.Meghana, S. SagarImambi, P. Sivateja, K. Sairam

Title : Image Recognition for Automatic Number Plate

This automatic number plate recognition system uses image processing technology for identification of the vehicles. This system can be used in highly populated areas and highly restricted areas to easily identify traffic rule violated vehicles. OCR algorithm. Efficient and first computing technique for identifying vehicle number plate. Computationally very inexpensive as compared with most of the conventional methods. Demerit is Limited area coverage.

Year : 2018

Author : Sina, A. Wibisono, A. Nurhadiyatna, B. Hardjono, W. Jatmiko

Title : Vehicle Counting and Speed Measurement Using Headlight Detection. We can use a few methods to detect and estimate vehicle speed at night by using CCTV Camera. Vehicle detection algorithm was used. Better detection accuracy than the areacentroid-difference method. The miscalculation of the vehicle counting happens because of the high density of the read.

Year : 2015

Author : Sina, A. Wibisono, A. Nurhadiyatna, B. Hardjono, W. Jatmiko

Title : Autonomous Vehicle And Real Time Road Lanes Detection And Tracking.

Work deals with a road boundaries and painted lines detection for intelligent and autonomous vehicles. Advanced computer vision algorithm used. Robust against exogenous perturbations and different constraints, but good enough to control the vehicle with a simple couple of fuzzy logic laws. The fuzzy controller handles the vehicle's steering, which has a limitation for a maximum speed of 70 km/h in sharp turns.

## **CHAPTER 3**

### **SYSTEM REQUIREMENTS**

#### **3.1 FUNCTIONAL REQUIREMENTS:**

Functional Requirements are the statements of services the system should provide and how the system reacts to particular inputs and how the system should behave in particular situation. The requirement specifies that a function that a system or component must be able to perform. These include inputs, outputs, calculations, external interfaces, communications, and special management information needs. Functional requirements are also called behavioral requirements because they address what the system does. A specification constraining the way in which a given task is to be performed, the results to be obtained (speed, accuracy, etc.) as well as the elements of the functional entities involved (initiator, source, receptor, etc.).

#### **3.2 NON-FUNCTIONAL REQUIREMENTS:**

In systems engineering and requirements engineering, non-functional requirements are requirements which specify criteria that can be used to judge the operation of a system, rather than specific behaviors. This should be contrasted with functional requirements that specify specific behavior or functions. Typical non-functional requirements are Reliability, Scalability, Performance, Usability, Maintainability, Portability and Cost. Other terms for non-functional requirements are "quality attributes" and "quality of service requirements". Non-functional requirements are divided into several groups. The first group of categories reflects the five qualities attributes:

- Usability
- Efficiency
- Reliability
- Maintainability
- Reusability

These requirements constrain the design to meet specified levels of quality. The second group of non-functional requirements categories constrains the environment and technology of the system. Efficiency

### 3.3 HARDWARE REQUIREMENTS

System : intel i3/i5 2.4 GHz.

Hard Disk : 500 GB

Ram : 4/8 GB

### 3.4 SOFTWARE REQUIREMENTS

Operating system : Windows XP/ Windows 7.

Software Tool : Open CV Python.

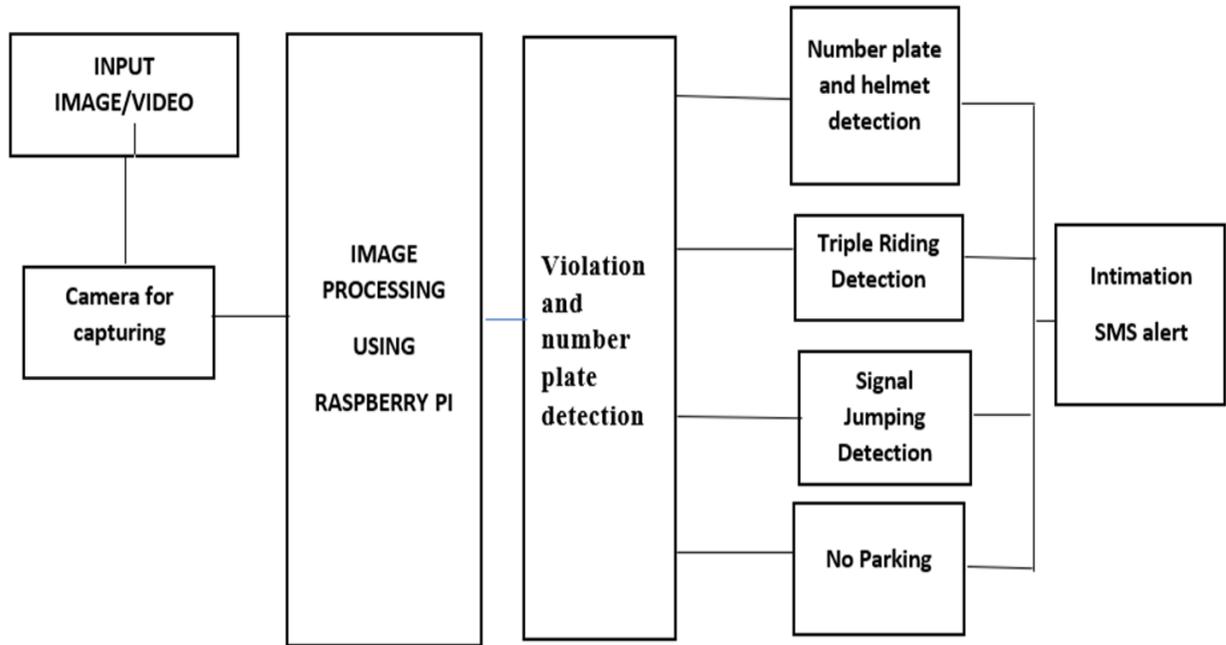
Coding Language : Python .

Toolbox : Image processing toolbox.

# CHAPTER 4

## SYSTEM DESIGN

### 4.1 SYSTEM ARCHITECTURE OF TRAFFIC INTEUTION SYSTEM



*Fig 4.1 : System Architecture of Traffic Intrusion System*

A video is taken from a camera, and then each frame of the video is processed as the image. In this stage the license plate region from the given image is located and isolated. Quality of the image plays an important part hence prior to this stage pre-processing of the image is necessary. So first each frame pre-processed by binarization, noise reduction and edge detection. Then, the license plate is located by different image processing technique.

## 4.2 STEPS :

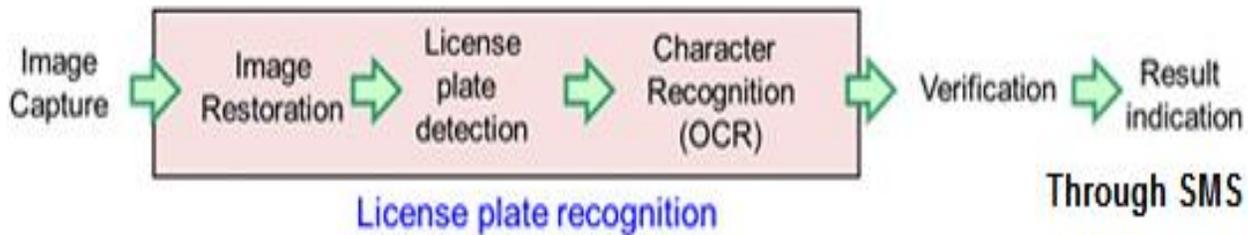


Fig 4.2 : Analysis Model of Traffic Intrusion System.

### **1) Input image from webcam.**

- Capture image from webcam.

### **2) Convert image into binary.**

- Identify the intensity of the image.
  - Calculate appropriate threshold value for the image.
- Convert the image into binary image using the calculated threshold.

### **3) Detect number plate area.**

- Determine width and height of the image.
- Fill small holes including numbers of Number plate so that number plate area will be large to isolate from figure.

### **4) Segmentation**

- Filter the noise level present in the image.
- Clip the plate area in such a way that only numbers of plate area extracted.

### **5) Number identification**

- Create the template file from the stored template images.
- Resize image obtained from segmentation to the size of template.

## **6) Save to file in given format**

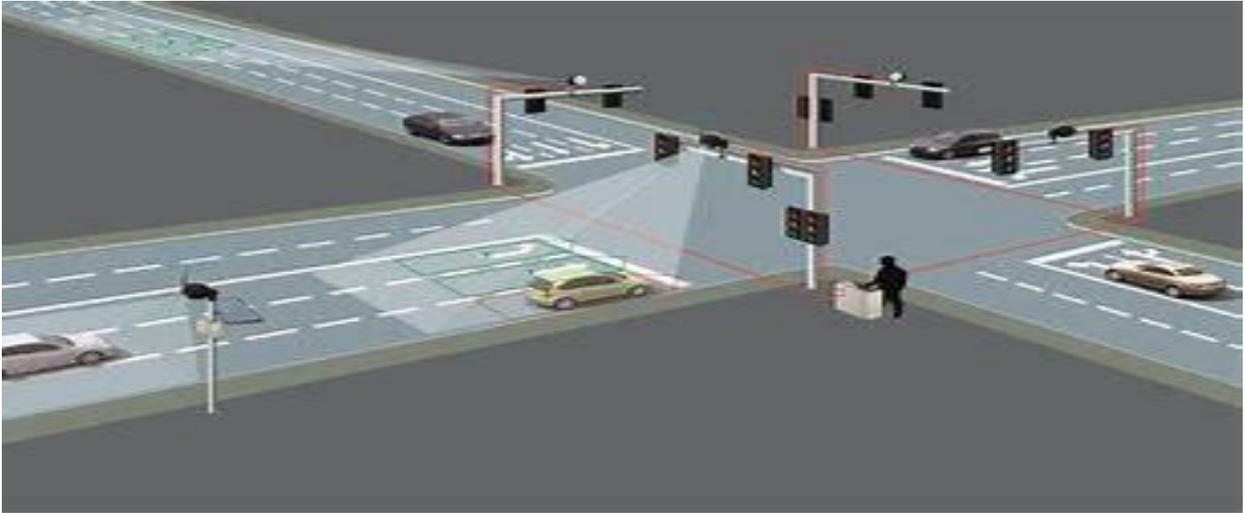
- Open a text file in write mode.
- Store the character obtained from the number identification process to text file in given format.

**MOTION DETECTION OPERATION:** When there is motion, the differences of the profiles are larger than the case when there is no motion. The motion can be detected by selecting a threshold value. Motion detectors most commonly use an infrared detection sensor, which can be incorporated into various devices found around us. This sensor provides data about motion, using which the device can alert the user of motion in the area or perform certain tasks. For example, a motion detector attached to a burglar alarm detects motion and enables the alarm to alert the home owners, or to activate a camera.

**VEHICLE DETECTION ALGORITHM:** A vehicle detection operation is applied on the profile of the unprocessed image. The method used here is based on applying edge detector operators to a profile of the image edges are less sensitive to the variation of ambient lighting and are used in full frame applications (detection). Different from those shallow models, deep learning has the ability of learning multiple levels of representation and abstraction that helps to make sense of image data. From another point of view, deep learning can be viewed as one kind of multilayer neural networks with adding a novel unsurprised pretraining process.

Although deep learning for object recognition has been an area of great interest in the machine-learning community, no prior research study has been reported that uses deep learning to establish an on-road vehicle detection method. In this paper, a 2D-DBN based vehicle detection algorithm is proposed.

The main novelty and contribution of this work include the following. A deep learning architecture of 2D-DBN which preserves discriminative information for vehicle detection is proposed. A deep learning based on-road vehicle detection system has been implemented and thorough quantitative performance analysis has been presented.



*Fig 4.3: VEHICLE DETECTION*

TRAFFIC MOVEMENTS AT JUNCTIONS (TMJ): The first step to measure the TMJ parameters using the key region method is to cover the boundary of the junction by a polygon in such a way that all the entry and exit paths of the junction cross the polygon. However, the polygon should not cover the pedestrian marked lines. This step is shown in the figure given below. The second step of the algorithm is to define a minimum number of key regions inside the boundary of the polygon, covering the junction.

An important safety aim is to match the speed at which drivers negotiate the junction with the complexity of the decisions to be made. This can be done, for example, by only allowing simple merging manoeuvres on high speed roads or by ensuring that drivers reduce speed on the junction approach (e.g. by deflection of path through a roundabout). Sight lines should provide drivers with sufficient information to make

safe decisions, but not tempt them to try to select short gaps in conflicting traffic flows. The majority of the various forms of channelization appear to have a more favourable effect on the number of accidents at crossroads than at t-junctions. There is a weak tendency that the more comprehensive the channelization methods are, the more favourable the effect on accidents.

### Redesigning junctions

Redesigning junctions includes:

- Change to the angle between roads
- Changes to the gradients of roads approaching the junction
- Other measures to improve sight conditions at junctions

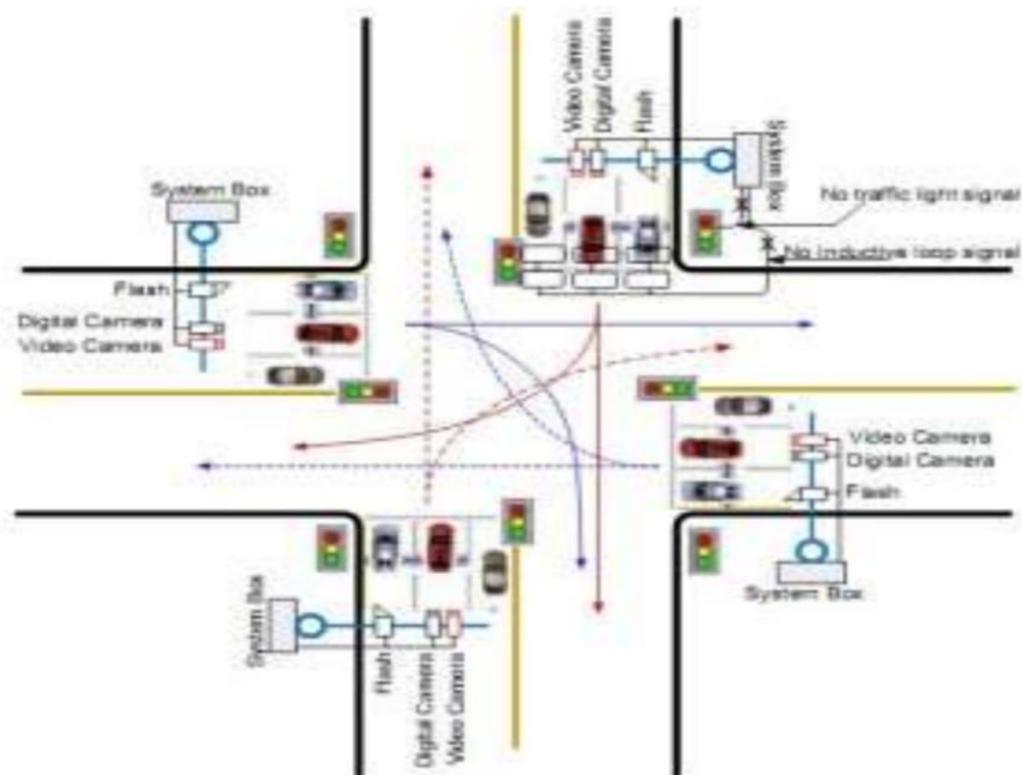


Fig 4.4: TRAFFIC MOVEMENTS AT JUNCTIONS (TMJ)

## 4.3 ALGORITHM USED

TESSERACT OCR : Number Plate Recognition

Gaussian Blur Algorithm

Canny Edge Detection : Edge Detection

CNN (Convolutional neural network) : Used for Customized Object Detection .

Tesseract OCR: The number plate recognition using Tesseract OCR starts with image acquisition, then the image is being pre-processed to remove any distortions and noise from the image. Then number plate is then localized irrelevant part and characters in the number plate are segmented and then the recognized and output. The license plate recognition detects number plates of all types of vehicles like car, truck and motorcycle. It also works on different number plate styles and fonts with high accuracy. Tesseract up to and including version 2 could only accept TIFF images of simple one-column text as inputs. These early versions did not include layout analysis, and so inputting multi-columned text, images, or equations produced garbled output. Since version 3.00 Tesseract has supported output text formatting, suggest positional information and page-layout analysis.

Gaussian Blur method:

Gray scaling and blurring: As the part of pre-processing the input frame got from the CCTV footage, the image is Gray scaled and blurred with Gaussian Blur method. In practice, it is best to take advantage of the Gaussian blur's separable property by dividing the process into two passes. In the first pass, a one-dimensional kernel is used to blur the image in only the horizontal or vertical direction. In the second pass, the same one-dimensional kernel is used to blur in the remaining direction. The resulting effect is the same as convolving with a two-dimensional kernel in a single pass, but requires fewer calculations.

Discretization is typically achieved by sampling the Gaussian filter kernel at discrete points, normally at positions corresponding to the midpoints of each pixel. This

reduces the computational cost but, for very small filter kernels, point sampling the Gaussian function with very few samples leads to a large error.

In these cases, accuracy is maintained (at a slight computational cost) by integration of the Gaussian function over each pixel's area.



*Fig. 4.5: Image Segmentation*



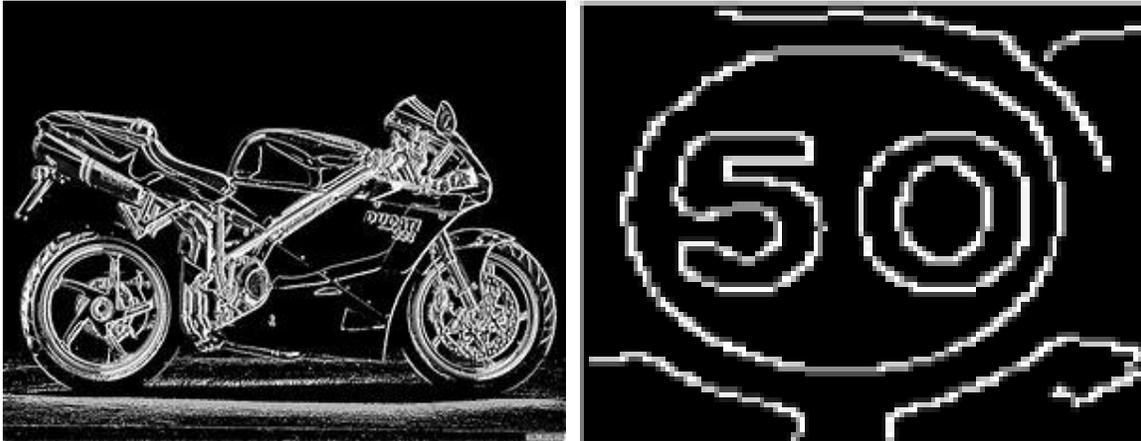
*Fig. 4.6: Gray Scaling*

Canny Edge Detection: The canny edge detector is an multi stage algorithm is used to detect wide range of edges in image. Edge detection is the process of identifying the edge in a digital image where the intensity of the image changes sharply or has discontinuities. The unwanted region is removed other than the number plate section.

Canny edge detection is a technique to extract useful structural information from different vision objects and dramatically reduce the amount of data to be processed. It has been widely applied in various computer vision systems. Canny has found that the requirements for the application of edge detection on diverse vision systems are relatively similar. Thus, an edge detection solution to address these requirements can be implemented in a wide range of situations. The general criteria for edge detection include:

1. Detection of edge with low error rate, which means that the detection should accurately catch as many edges shown in the image as possible

2. The edge point detected from the operator should accurately localize on the center of the edge.
3. A given edge in the image should only be marked once, and where possible, image noise should not create false edges.



*Fig. 4.7: Canny Edge Image*

- CNN (Convolutional neural network) :The proposed architecture of surveillance system with intelligent detection and tracking of multiple vehicles from the surveillance input video .This is done through CNN and an object detection model which are used in the classification of the moving objects into different respective classes, thus achieving vehicle classification. The construction of a convolutional neural network is a multi-layered feed-forward neural network, made by assembling many unseen layers on top of each other in a particular order.
- It is the sequential design that give permission to CNN to learn hierarchical attributes.
- In CNN, some of them followed by grouping layers and hidden layers are typically convolution layers followed by activation layers.
- The pre-processing needed in a Net is kindred to that of the related pattern of neurons in the human brain and was motivated by the organization of the Visual Cortex.

- A convolutional neural network, or CNN, is a deep learning neural network sketched for processing structured arrays of data such as portrayals.
- CNN are very satisfactory at picking up on design in the input image, such as lines, gradients, circles, or even eyes and faces.
- This characteristic that makes convolutional neural network so robust for computer vision.
- CNN can run directly on a underdone image and do not need any preprocessing.
- A convolutional neural network is a feed forward neural network, seldom with up to 20.
- The strength of a convolutional neural network comes from a particular kind of layer called the convolutional layer.
- CNN contains many convolutional layers assembled on top of each other, each one competent of recognizing more sophisticated shapes.
- With three or four convolutional layers it is viable to recognize handwritten digits and with 25 layers it is possible to differentiate human faces.
- The agenda for this sphere is to activate machines to view the world as humans do, perceive it in a alike fashion and even use the knowledge for a multitude of duty such as image and video recognition, image inspection and classification, media recreation, recommendation systems, natural language processing, etc.

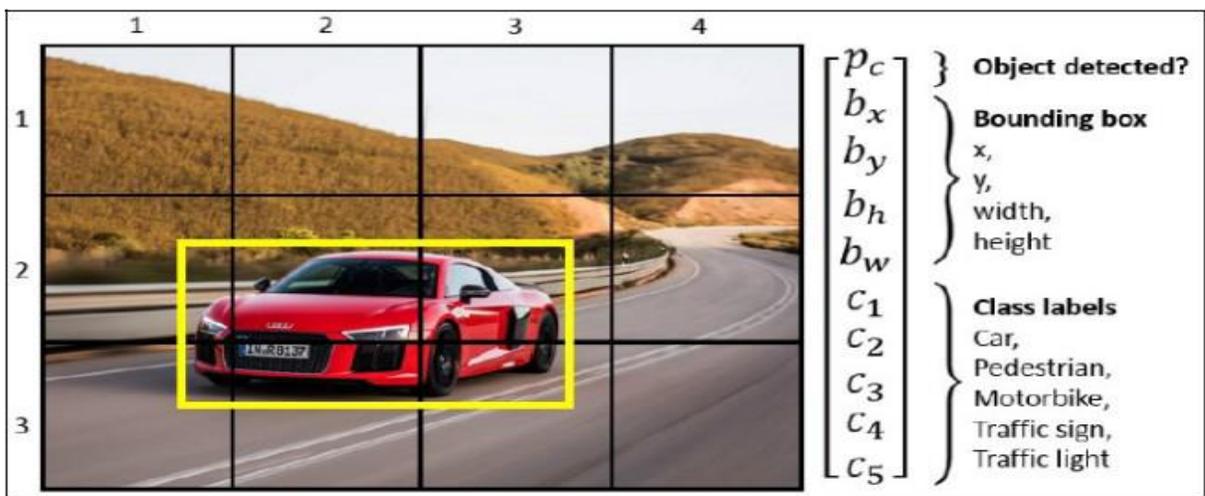
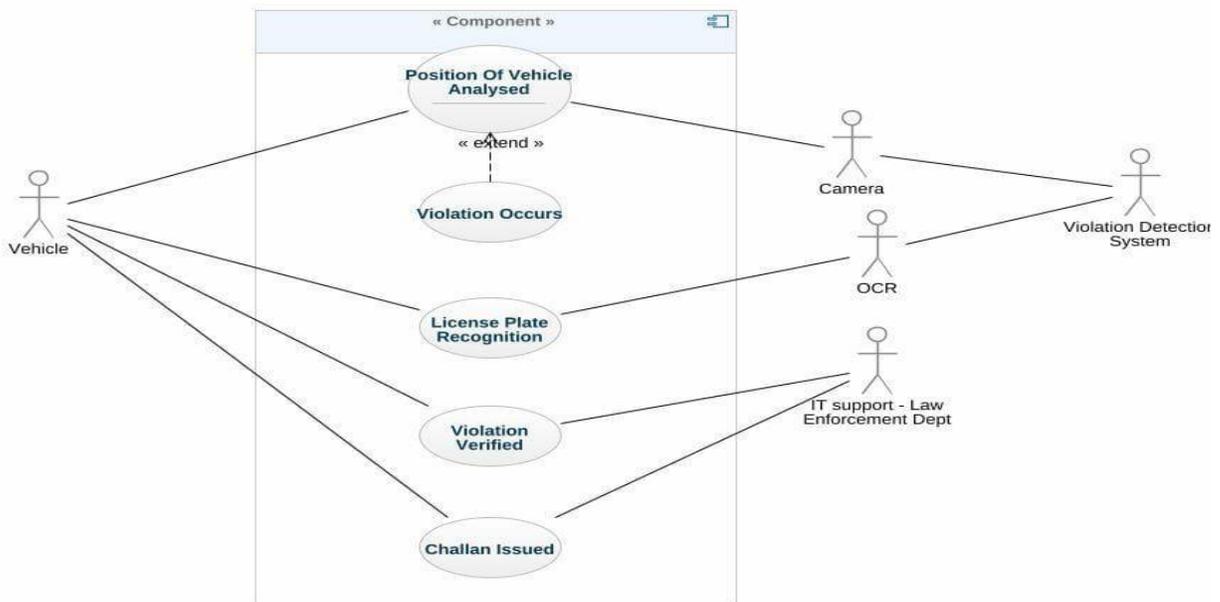


Fig 4.8: An object detection model

#### 4.4 USE CASE DIAGRAM

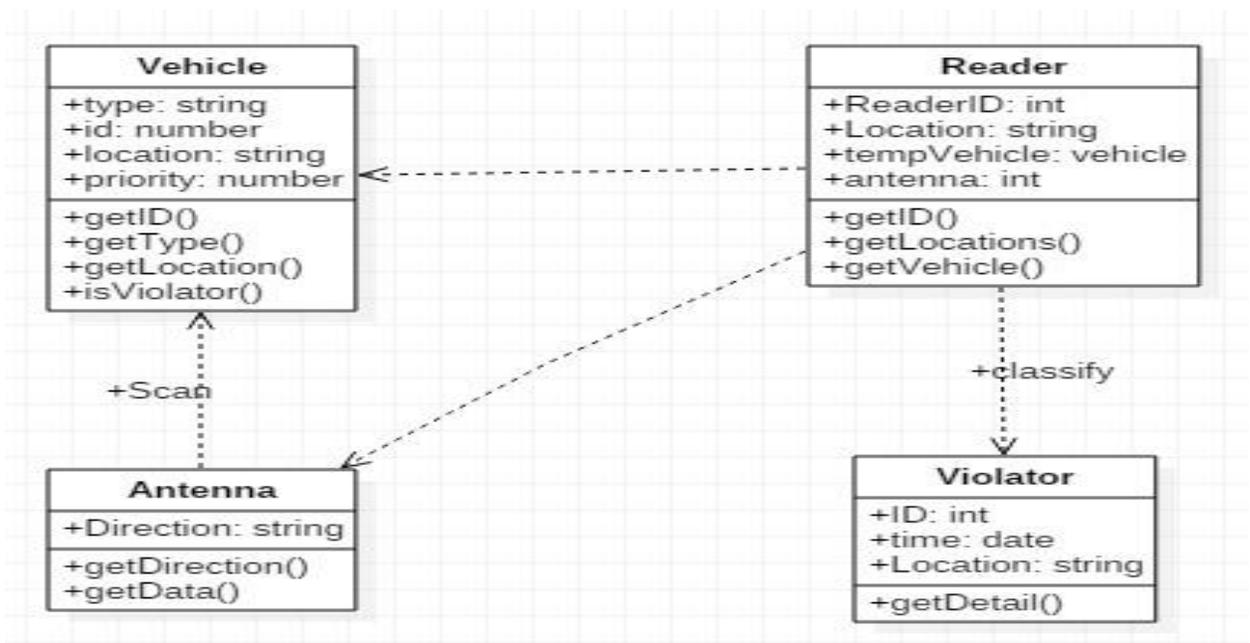
A use case diagram is a dynamic or behavior diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform. In this context, a "system" is something being developed or operated, such as a web site. The "actors" are people or entities operating under defined roles within the system.



#### 4.5 CLASS DIAGRAM

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.



Number Plate of the vehicle is authenticated and data is taken Since this model is capable of detecting vehicles in every frame it is having been enhanced to detect the speed of the vehicle in a video. Detection of vehicle crossing during a red light in the image and locate the exact location. Data of the vehicle is taken when the people riding on two-wheeler is more than two Vehicle parked in the no parking area is given five minutes time to change the vehicle from the spot if not changed violation is detected and fine is sent through SMS. Detects and reads license plates automatically 24 x 7 in real time. High accuracy for heterogeneous number plates. Alerts/Alarms for stolen or marked vehicles. Capable of detecting and recognizing different dimensions, contrast, colors license plates with variety of character font & style.

## CHAPTER 5

### RESULT AND DISCUSSION

- Automation in day to day life has gained importance in recent years.
- The number of accidents on the roads is due to the rule violations such as breaking traffic signals, over-speeding, driving on wrong sides etc.
- To avoid such traffic violations, traffic police has to be present on the road and has to continuously check if some vehicle is violating the rule.
- It will be easier for the police to catch who break laws of the traffic and for the policemen who take bribes and dishonest people will also be reduced.
- An image of the violation with the location, time, date and an image of the vehicle to the nearest policemen devices by deriving devices around the current location that violation happened. Without police also the camera will check the condition.
- Violation Detection using CNN Algorithm with respect to helmet, Signal Jumping, No parking.
- If any traffic violation occurs immediately send alert message to owner number.
- Number plate recognition for each traffic violation.
- The first approach that relays on human participation is the traffic patrols.
- A traffic patrol is a police unit created primarily for the purpose of overseeing and enforcing traffic safety compliance on roads and highways.
- Another technology to fight the traffic violations is by using cameras installed on the traffic lights.
- The goal is just to detect the red-light crossing violations.

## ADVANTAGES

- Detects and reads license plates automatically 24 x 7 in real time.
- High accuracy for heterogeneous number plates.
- Transmits real-time stamp of entry/exit of the vehicle to the database.
- Alerts/Alarms for stolen or marked vehicles.
- Capable of detecting and recognizing different dimensions, contrast, colors license plates with variety of character font & style.

## CHAPTER 6

### CONCLUSION AND FUTURE WORK

Number plate recognition: Number Plate of the vehicle is authenticated and data is taken.

Helmet detection : Since this model is capable of detecting vehicles in every frame it is having been enhanced to detect the speed of the vehicle in a video.

Signal jumping detection: Detection of vehicle crossing during a red light in the image and locate the exact location.

Triple riding recognition: Data of the vehicle is taken when the people riding on two wheeler is more than two.

No parking detection : vehicle parked in the no parking area is given five minutes time to change the vehicle from the spot if not changed violation is detected and fine is sent through SMS.

#### 6.1 FUTURE SCOPE

- With the increasing growth in traffic density all over the world, it possesses a great challenge to traffic management.
- Emphasis should be that large area is covered and the high volume of traffic monitoring and detection from a single input source using parallel computation.
- Further enhancements are required to reduce computational time at high traffic volume roads.
- Further, it can be implemented for larger datasets by training using GPUs and high-end FPGA kits

## REFERENCES

[1] "Implementing Lane Departure Warning System on Raspberry Pi", University Tunku Abdul Rahman, May 2016.

[2] "A Real-Time Vision System for Night time Vehicle Detection and Traffic Surveillance", IEEE transactions on industrial electronics, Volume-58, No-5, May 2011.

## BOOKS REFERED

1. Artificial Intelligence: A Modern Approach (Fourth edition, 2020) by Stuart Russell and Peter Norwig.
2. Understanding Machine Learning: From Theory to Algorithms c 2014 by Shai ShalevShwartz and Shai Ben-David.

## WEBSITE VISITED

1. <https://searchenterpriseai.techtarget.com/>
2. <https://www.geeksforgeeks.org/>

## A.SOURCE CODE:

```
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[0] - 1] for i in net.getUnconnectedOutLayers()]

# Draw the predicted bounding box
def drawPred(classId, conf, left, top, right, bottom):

```

```

    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(':') #splitting into class &
confidence. 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 occurrence.

```

#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]
```

```
    global frame_count_out
```

```
    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)

    #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)

    #cv.imwrite(frame_name, frame)

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

```

```
# Process inputs

winName = 'Deep learning object detection in OpenCV'
cv.namedWindow(winName, cv.WINDOW_NORMAL)

for fn in glob('images/*.jpg'):
    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
    postprocess(frame, outs)
```

```
# 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)
```





