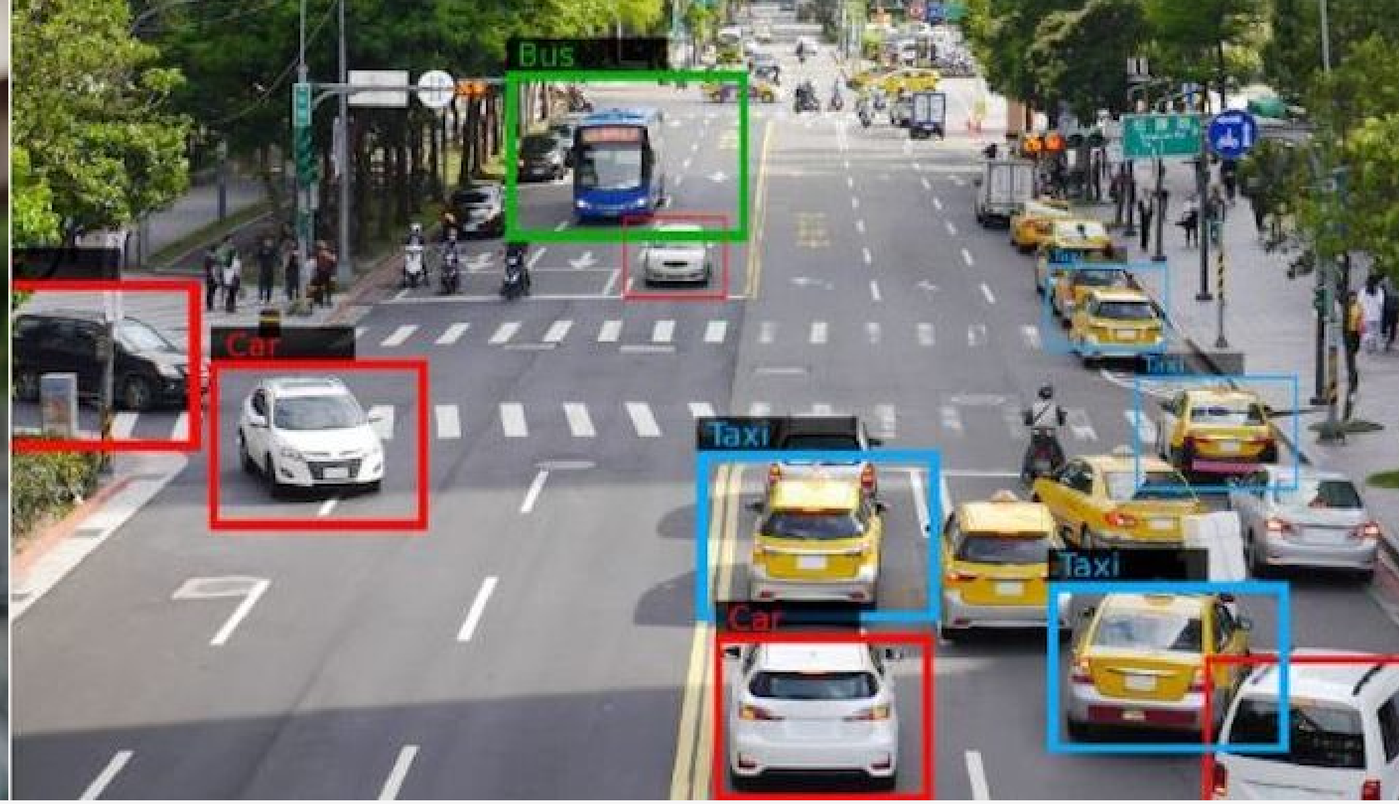
TRAFFIC SURVEILLANCE SYSTEM

Team Members:

Tushar Rawat
Vaibhav Banga
Rahul Kadam
Prashant Kumar
Akash Sarkar
Mrinal Shahi
Kumar Shubham
Vikas Rathod

Team Leader:

Vivek Chaudhary



Problem Statement

How to build a Custom Object Detection Model with respect to Traffic Surviellance using Computer Vision.

Flowchart

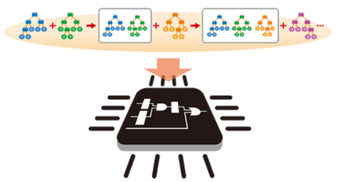
Data Collection

LabelImg

YOLO v3

YOLO

MODEL TRAINING



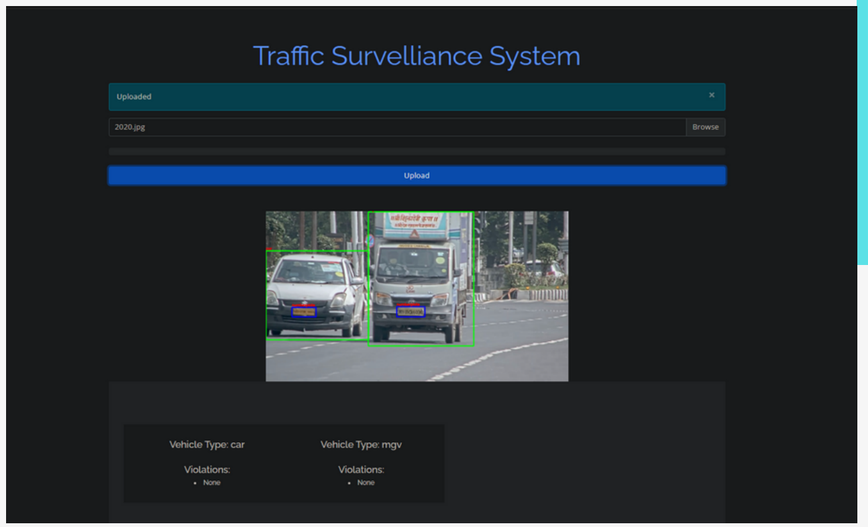
DEPLOYMENT

Flask

node.js

TESTING WITH API

POSTMAN



DATASET AND CHALLENGES FACED

Dataset

Dataset collected by team members in real time.

Collecting large amount of real-time data was challenging

MODELS WE REVIEWED..

SSD

RCNN

You Only Look Once

YOLO requires a small amount of data to train and provide us results.

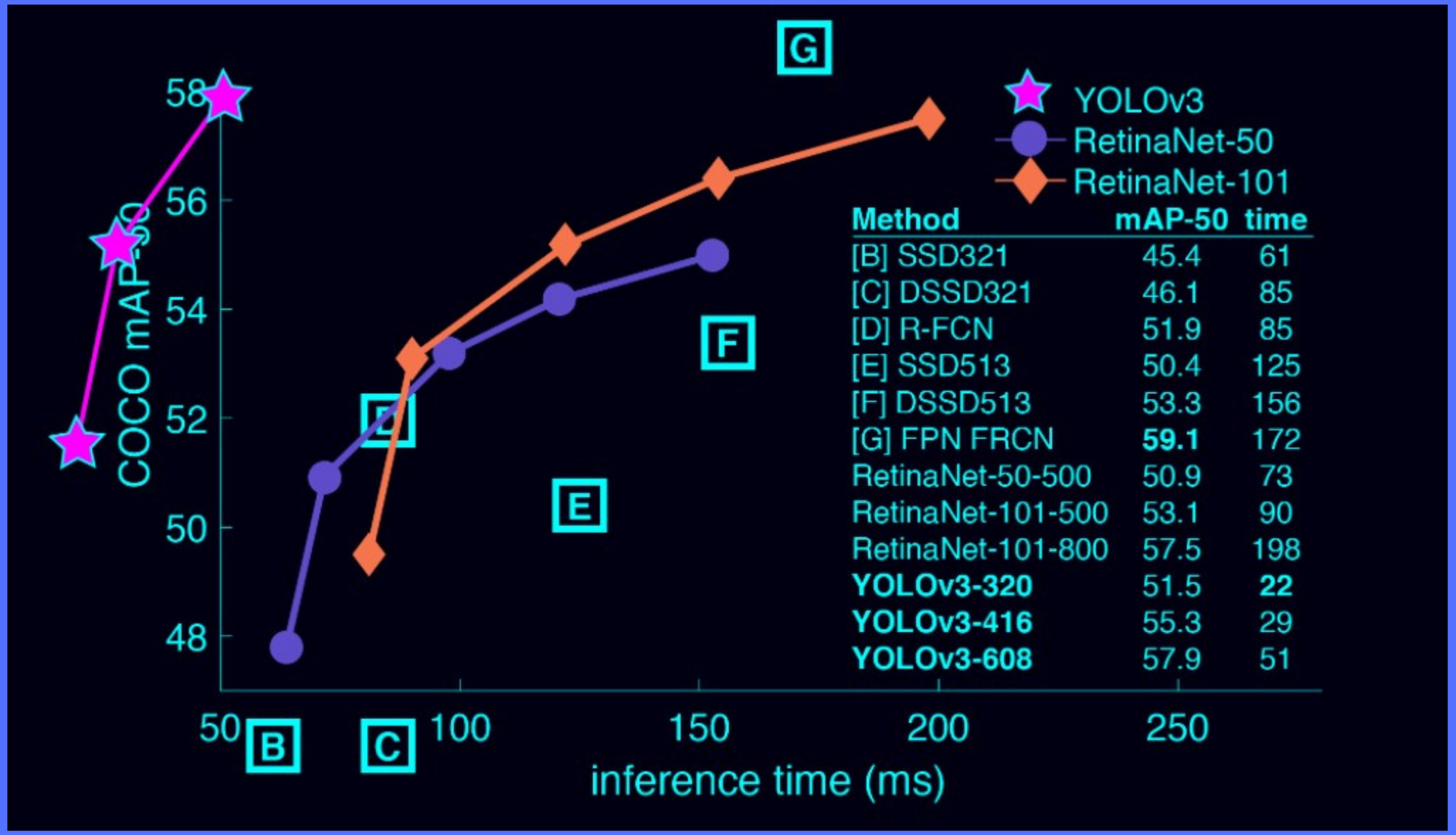
Single-Shot Detector

SSD requires large amount of data to train and give results (Greedy Algorithm)

Regional based convolutional network

RCNN is slower than YOLO because YOLO trained to do classification and bounding box regression at the same time.

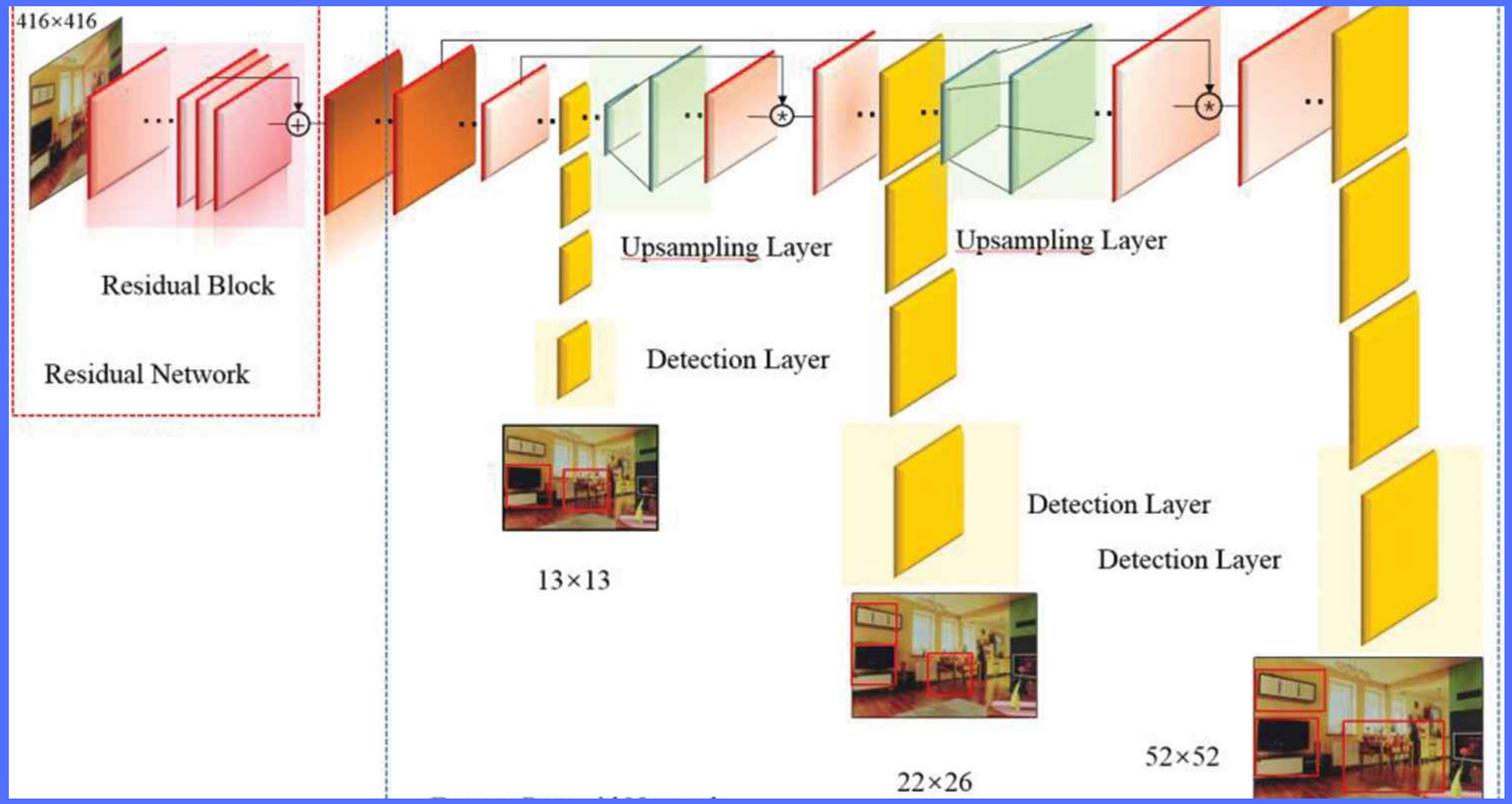
COMPARISON



Method	mAP-50	time
[B] SSD321	45.4	61
[C] DSSD321	46.1	85
[D] R-FCN	51.9	85
[E] SSD513	50.4	125
[F] DSSD513	53.3	156
[G] FPN FRCN	**59.1**	172
RetinaNet-50-500	50.9	73
RetinaNet-101-500	53.1	90
RetinaNet-101-800	57.5	198
YOLOv3-320	51.5	**22**
YOLOv3-416	55.3	29
YOLOv3-608	57.9	51

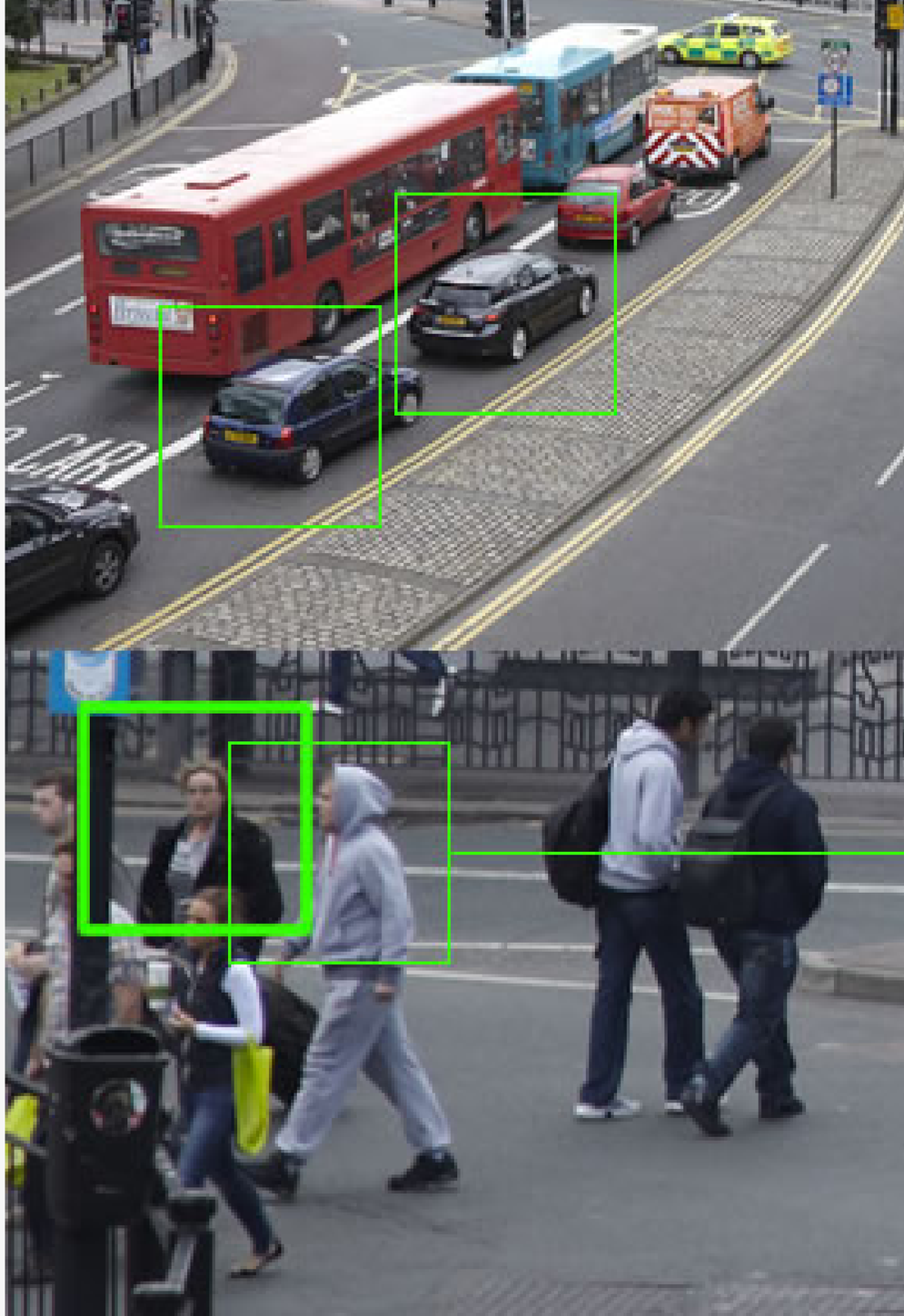
ARCHITECTURE OF YOLO

The architecture of various versions of YOLO can be accessed by their respective cfg files.



YOLO ADVANTAGES

- Process Frames at Rates
- Generalize the image better.
- Fast and Accurate



Yolov3 Code

```
!echo -e 'Number Plate' > data/obj.names
!mkdir data/backup
!echo -e 'classes= 1\ntrain  = data/train.txt\nvalid  = data/valid.txt\nnames = data/obj.names\nbackup = data/backup/' > data/obj.data
```

MagicPython

```
!cp cfg/yolov3_training.cfg data/backup/yolov3_training.cfg
!cp data/obj.names data/backup/classes.txt
```

MagicPython

```
!./darknet detector train data/obj.data cfg/yolov3_training.cfg darknet53.conv.74 -dont_show
```

MagicPython

```
Streaming output truncated to the last 5000 lines.
 1904: 0.740037, 0.909278 avg loss, 0.001000 rate, 1.714151 seconds, 121856 images, 0.130508 hours left
Loaded: 0.290747 seconds - performance bottleneck on CPU or Disk HDD/SSD
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 16 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000000
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 23 Avg (IOU: 0.696128), count: 9, class_loss = 0.947442, iou_loss = 0.517947, total_loss = 1.465389
 total_bbox = 265974, rewritten_bbox = 0.000752 %
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 16 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000000
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 23 Avg (IOU: 0.565241), count: 11, class_loss = 2.002429, iou_loss = 0.777534, total_loss = 2.779963
 total_bbox = 265985, rewritten_bbox = 0.000752 %
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 16 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000000
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 23 Avg (IOU: 0.600935), count: 10, class_loss = 0.835737, iou_loss = 0.801129, total_loss = 1.636865
 total_bbox = 265995, rewritten_bbox = 0.000752 %
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 16 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000000
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 23 Avg (IOU: 0.585482), count: 11, class_loss = 1.103880, iou_loss = 1.662903, total_loss = 2.766783
```

Pipeline Code

```
36  ##################################################################################################
37  plate_wgh_pth = resource_path(r'checkpoint/NUm_plate/yolov4_training_final.weights')
38  plate_cfg_pth = resource_path(r'checkpoint/NUm_plate/yolov4_training.cfg')
39  plate_label_pth = resource_path(r'checkpoint/NUm_plate/objplate.names')
40
41  # Vehicle Detection Checkpoint
42  vd_wgh_pth = resource_path(r'checkpoint/vehicleDetection/yolov4_vehicle_detection-288.weights')
43  vd_cfg_pth = resource_path(r'checkpoint/vehicleDetection/yolov4_vehicle_detection-288.cfg')
44  vd_label_pth =resource_path(r'checkpoint/vehicleDetection/yolov4_vehicle_detection-288.names')
45
46  # Helemet
47  violation_wgh_pth = resource_path(r'checkpoint/helmat/yolov4-training-288-final_final.weights')
48  violation_cfg_pth = resource_path(r'checkpoint/helmat/yolov4-training-288.cfg')
49  violation_label_pth = resource_path(r'checkpoint/helmat/helmet-yolov4-288.names')
50
51  # Loading Plate Detection Model
52  platedetection = yolo.LoadYOLO(input_h=256, input_w=256, \
53      labelPath=plate_label_pth, cfgPath=plate_cfg_pth, weightPath=plate_wgh_pth)
54
55  # Loading Vehicle Detection Model
56  vehicledetection = yolo.LoadYOLO(input_h=288, input_w=288, \
57      labelPath=vd_label_pth, cfgPath=vd_cfg_pth, weightPath=vd_wgh_pth)
58  # Loading Helemet
59  helemt = yolo.LoadYOLO(input_h=288, input_w=288, \
60      labelPath=violation_label_pth, cfgPath=violation_cfg_pth, weightPath=violation_wgh_pth)
61
62  ##################################################################################################
```

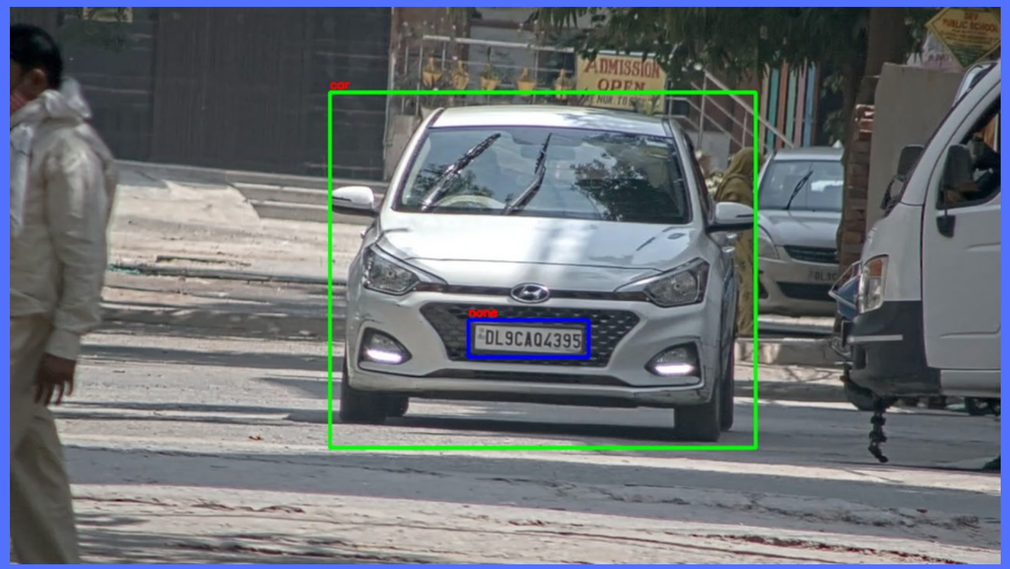
API

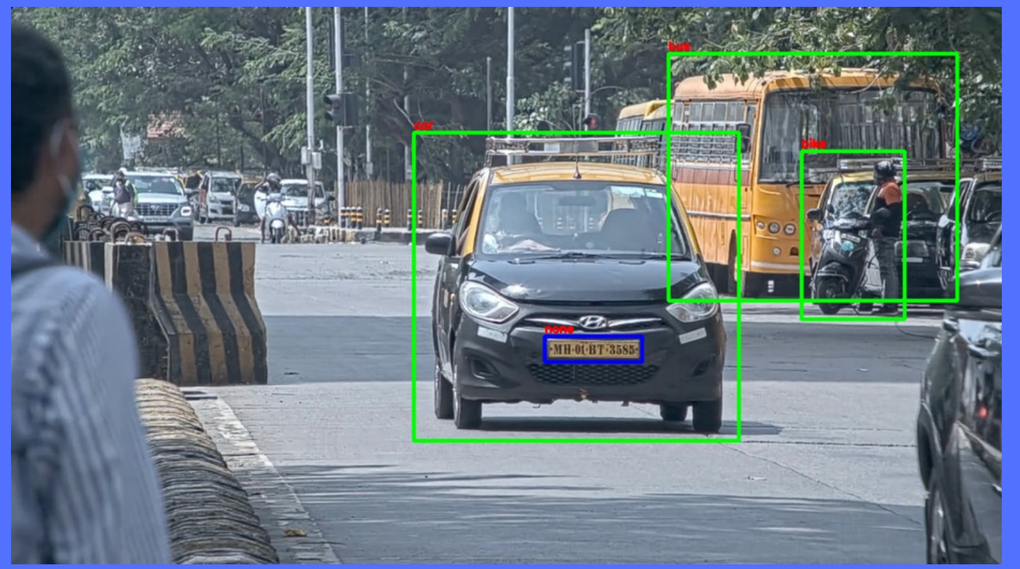
```
const [file, setFile] = useState('');
const [filename, setFilename] = useState('Choose File');
const [uploadedFile, setUploadedFile] = useState({});
const [message, setMessage] = useState('');
const [uploadPercentage, setUploadPercentage] = useState(0);
const [process,setprocess]=useState(0)
const [output_data,setoutput_data]=useState([]);
const [res,setres]=useState([]);
const onChange = e => {
  setFile(e.target.files[0]);
  setFilename(e.target.files[0].name);
};
const onSubmit = async e => {
  e.preventDefault();
  const formData = new FormData();
  formData.append('files', file);
  try {
      const res = await axios.post('http://localhost:5000/image', formData, {
       headers: {
         'Content-Type': 'multipart/form-data'
       },
       onUploadProgress: progressEvent => {
         setUploadPercentage(
```

OUTPUT


car
car
none
MH01BT2646
none
MH.01.BD.2930

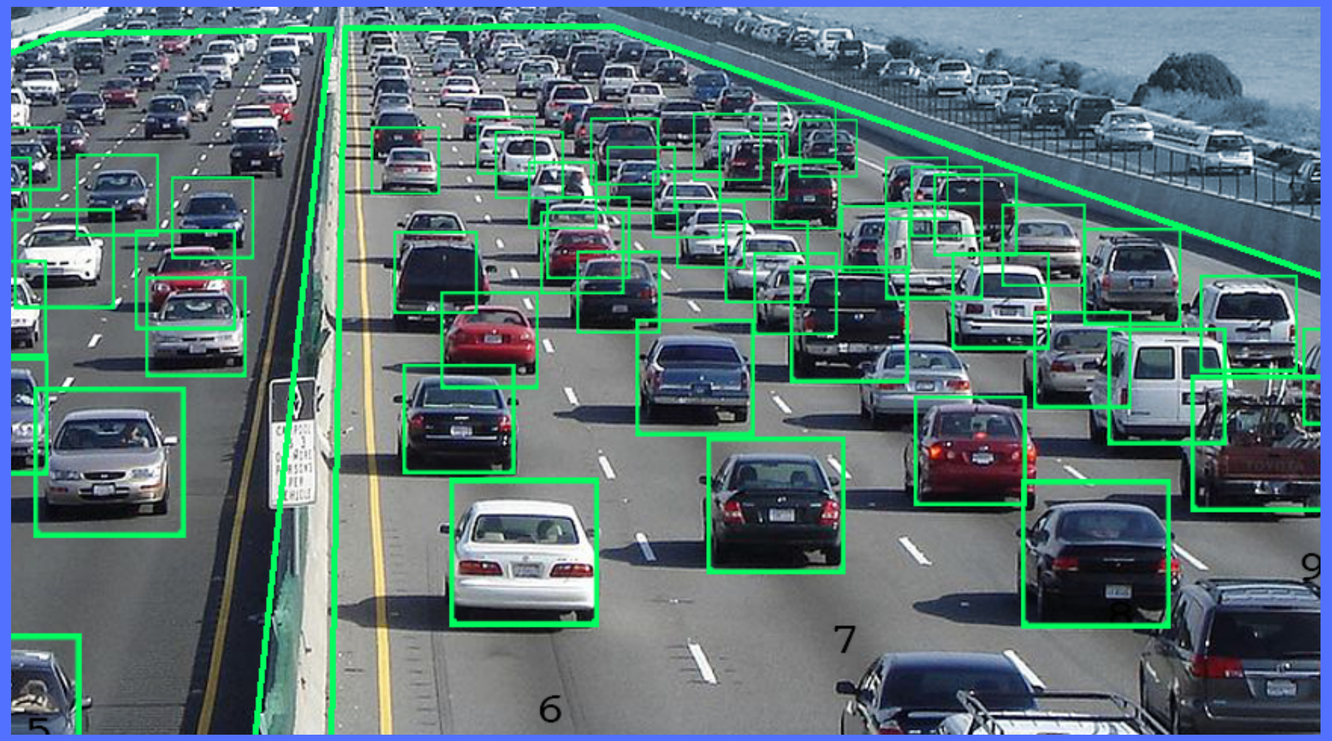

bike
none


car
none
DL9CAQ4395


bus
car
bike
none
MH-01-BT-3582

CONCLUSION

This Model will help all the traffic policemen to save their work and time.
It will reduce the violations and accidents.



FUTURE SCOPE

1. It can work as the eye of a traffic policeman.
2. We can install this at minimal cost on the roads.
3. We can add more models to it :
   Wrong lane detection, Signal jumping
   Seatbelt Detection, Overspeed

THANKS FOR ATTENDING!