```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
!git clone https://github.com/karnoark/object-detection-on-indian-driving-dataset.git
```

```
Cloning into 'object-detection-on-indian-driving-dataset'...
remote: Enumerating objects: 116, done.
remote: Counting objects: 100% (113/113), done.
remote: Compressing objects: 100% (76/76), done.
remote: Total 116 (delta 35), reused 86 (delta 30), pack-reused 3
Receiving objects: 100% (116/116), 29.71 MiB | 29.62 MiB/s, done.
Resolving deltas: 100% (35/35), done.
```

```
!git clone https://github.com/ultralytics/yolov5
%cd yolov5
```

```
Cloning into 'yolov5'...
remote: Enumerating objects: 16582, done.
remote: Counting objects: 100% (60/60), done.
remote: Compressing objects: 100% (42/42), done.
remote: Total 16582 (delta 30), reused 39 (delta 18), pack-reused 16522
Receiving objects: 100% (16582/16582), 15.06 MiB | 24.40 MiB/s, done.
Resolving deltas: 100% (11389/11389), done.
/content/yolov5
```

```
# install dependencies as necessary


!pip install -qr requirements.txt  # install dependencies (ignore errors)
import torch

from IPython.display import Image, clear_output  # to display images



# clear_output()
print('Setup complete. Using torch %s %s' % (torch.__version__, torch.cuda.get_device_properties(0) if torch.cuda.is_available() else 'CPU'))
```

```
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 207.3/207.3 kB 2.0 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4.5/4.5 MB 24.0 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 755.6/755.6 kB 57.6 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 62.7/62.7 kB 7.8 MB/s eta 0:00:00
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source
imageio 2.31.6 requires pillow<10.1.0,>=8.3.2, but you have pillow 10.3.0 which is incompatible.
Setup complete. Using torch 2.2.1+cu121 _CudaDeviceProperties(name='Tesla T4', major=7, minor=5, total_memory=15102MB, multi_processor_c
```

```
%cd /content
!curl -L "https://app.roboflow.com/ds/Y4exlFspST?key=FBsyjEmvum" > roboflow.zip; unzip roboflow.zip; rm roboflow.zip
```

```
Streaming output truncated to the last 5000 lines.
 extracting: train/images/0009555_jpg.rf.18dd67298796a0b11dc677d1a2abc076.jpg
 extracting: train/images/000955_r_jpg.rf.4d5f23a89563bfe342119fc04be67cf0.jpg
 extracting: train/images/000955_r_jpg.rf.d214a78464c66533602bd196cc62ccf5.jpg
 extracting: train/images/0009600_jpg.rf.de217dfc17858db84094635bf5d33d9c.jpg
 extracting: train/images/000960_r_jpg.rf.f178445918260c3dabfe786b4f237dd1.jpg
 extracting: train/images/0009645_jpg.rf.63103d10c12f97e348c7ca07898b5cf1.jpg
 extracting: train/images/000964_r_jpg.rf.732962d3a529008668a6d718e2d64f08.jpg
 extracting: train/images/0009660_jpg.rf.684eb7b4756c7b776fff348c67120862.jpg
 extracting: train/images/0009660_jpg.rf.b128c1b45eee7a42aa094fac1fdbd47f.jpg
 extracting: train/images/0009662_jpg.rf.a99603f0f6a8d4699e14fdb3830115c6.jpg
 extracting: train/images/0009690_jpg.rf.30d6faae9bc54d05db7f99f5b0453fdb.jpg
 extracting: train/images/0009690_jpg.rf.5ae58ea70e82002d6865fec9fe43afe6.jpg
 extracting: train/images/000969_r_jpg.rf.0bc3804cc83f64a046c7bf2a28437673.jpg
 extracting: train/images/000969_r_jpg.rf.b8dbbfc22d560e1ebd5823ab03862968.jpg
 extracting: train/images/0009720_jpg.rf.0414b5ff4f07e451d6c36d745700b833.jpg
 extracting: train/images/0009735_jpg.rf.af52098d57ee13b72e5eab9dc709e1f6.jpg
 extracting: train/images/0009780_jpg.rf.1541c8717d8270a459c4dc0d6e281374.jpg
 extracting: train/images/0009780_jpg.rf.f508d5e78de5f757f6509690013b8b6d.jpg
 extracting: train/images/000978_r_jpg.rf.50ba571713e7085b5cb6d49a91bee8b1.jpg
 extracting: train/images/000978_r_jpg.rf.56393cdfcf81aa0b22787087c54c8203.jpg
 extracting: train/images/0009825_jpg.rf.daa0e685bf2037fd69524fc7a27bdace.jpg
 extracting: train/images/000982_r_jpg.rf.8c3080765ffd40d9e1b4ec07a0f3e290.jpg
 extracting: train/images/0009840_jpg.rf.690a66e3c65cd00cb0a63e2ac349b6a1.jpg
```

```
extracting: train/images/0009870_jpg.rf.a186fb8a9d3172c1dfe196deee28f993.jpg
extracting: train/images/000987_r_jpg.rf.7e57f495400649d32c05d1225c1fe76f.jpg
extracting: train/images/0009900_jpg.rf.46ece18b496dbe6f003e4732d249e831.jpg
extracting: train/images/0009900_jpg.rf.6b818eca58b8385e2414468faaad8b8e.jpg
extracting: train/images/000990_r_jpg.rf.03141e23461ae010e80210aa34973091.jpg
extracting: train/images/000990_r_jpg.rf.3be395b795d32f4abbbc0e93c8bb995b.jpg
extracting: train/images/0009915_jpg.rf.29b6fdfa9e00a1f5c82b7ad424534912.jpg
extracting: train/images/000991_r_jpg.rf.036a58aa39f2a89d3bb1ea8e260b174b.jpg
extracting: train/images/0009960_jpg.rf.8f4bdb69e58f438f24afd15f78c0b563.jpg
extracting: train/images/000996_r_jpg.rf.11fedd822c88960036b13aec655a8d32.jpg
extracting: train/images/000996_r_jpg.rf.29ddb5ca2b84180dd9842cc252bf2ac9.jpg
extracting: train/images/0010005_jpg.rf.0a597f1cfe08324035150bcde4ecaa3a.jpg
extracting: train/images/001000_r_jpg.rf.0f667187d25264d508d0df81451ddea8.jpg
extracting: train/images/001000_r_jpg.rf.e409076bdc432e6d05f81d6de3205861.jpg
extracting: train/images/0010020_jpg.rf.365ee70dfd23151f7628977e5ab40396.jpg
extracting: train/images/0010020_jpg.rf.bf75f518973007e4776446115a7867ac.jpg
extracting: train/images/001002_r_jpg.rf.50451b2f60909f97a63743a17cf28e22.jpg
extracting: train/images/001002_r_jpg.rf.ae9c540aa890ac84b85662ac7202f072.jpg
extracting: train/images/0010050_jpg.rf.b3416b6f8d9002f83786f910b7b7a7ff.jpg
extracting: train/images/0010050_jpg.rf.b79470298c259ef27431a1b76df060ec.jpg
extracting: train/images/001005_r_jpg.rf.a6b825f87c2cf675c9d59ac10b064918.jpg
extracting: train/images/0010068_jpg.rf.ed58878a12abcfb0002438a034120197.jpg
extracting: train/images/0010080_jpg.rf.efa6767551e881d4cd4c158830ad30e0.jpg
extracting: train/images/001009_r_jpg.rf.3db879fc5d3c1d1208de6c94792991ba.jpg
extracting: train/images/0010140_jpg.rf.6d2ec5e1b41cc290d7a1d99f16037c06.jpg
extracting: train/images/001014_r_jpg.rf.6853cf7ee8b5965ce3af707b703a9ee8.jpg
extracting: train/images/001014_r_jpg.rf.fbb80ef4a7c5dd5d54833ec1019607c9.jpg
extracting: train/images/0010185_jpg.rf.558a674754e114cdd527a8cce647ded4.jpg
extracting: train/images/001018_r_jpg.rf.1c6a5542d1a4a5a93e6b302f47d8188c.jpg
extracting: train/images/0010200_jpg.rf.edbbee72df77467696138afd94fb9a57.jpg
extracting: train/images/001020_r_jpg.rf.a5bfc366c3cb59542ecc54531b8f852a.jpg
extracting: train/images/0010230_jpg.rf.118df1289437af6691975c8d6949e8d1.jpg
extracting: train/images/001023_r_jpg.rf.351c7a1a1bdd6c50f99ee8dd4b07b6bc.jpg
extracting: train/images/0010320_jpg.rf.22d0275d256ab39b4efa1f4733dc2988.jpg
```

```
%cat data.yaml
```

```
train: ../train/images
val: ../valid/images

nc: 15
names: ['animal', 'autorickshaw', 'bicycle', 'bus', 'car', 'caravan', 'motorcycle', 'person', 'rider', 'traffic light', 'traffic sign',
```

```
# define number of classes based on YAML
import yaml
with open("data.yaml", 'r') as stream:
    num_classes = str(yaml.safe_load(stream)['nc'])
```

```
%cat /content/yolov5/models/yolov5l.yaml
```

```
# YOLOv5 🚀 by Ultralytics, AGPL-3.0 license

# Parameters
nc: 80 # number of classes
depth_multiple: 1.0 # model depth multiple
width_multiple: 1.0 # layer channel multiple
anchors:
  - [10, 13, 16, 30, 33, 23] # P3/8
  - [30, 61, 62, 45, 59, 119] # P4/16
  - [116, 90, 156, 198, 373, 326] # P5/32

# YOLOv5 v6.0 backbone
backbone:
  # [from, number, module, args]
  [
    [-1, 1, Conv, [64, 6, 2, 2]], # 0-P1/2
    [-1, 1, Conv, [128, 3, 2]], # 1-P2/4
    [-1, 3, C3, [128]],
    [-1, 1, Conv, [256, 3, 2]], # 3-P3/8
    [-1, 6, C3, [256]],
    [-1, 1, Conv, [512, 3, 2]], # 5-P4/16
    [-1, 9, C3, [512]],
    [-1, 1, Conv, [1024, 3, 2]], # 7-P5/32
    [-1, 3, C3, [1024]],
    [-1, 1, SPPF, [1024, 5]], # 9
  ]

# YOLOv5 v6.0 head
head: [
```

```
         [-1, 1, Conv, [512, 1, 1]],
         [-1, 1, nn.Upsample, [None, 2, "nearest"]],
         [[-1, 6], 1, Concat, [1]], # cat backbone P4
         [-1, 3, C3, [512, False]], # 13

         [-1, 1, Conv, [256, 1, 1]],
         [-1, 1, nn.Upsample, [None, 2, "nearest"]],
         [[-1, 4], 1, Concat, [1]], # cat backbone P3
         [-1, 3, C3, [256, False]], # 17 (P3/8-small)

         [-1, 1, Conv, [256, 3, 2]],
         [[-1, 14], 1, Concat, [1]], # cat head P4
         [-1, 3, C3, [512, False]], # 20 (P4/16-medium)

         [-1, 1, Conv, [512, 3, 2]],
         [[-1, 10], 1, Concat, [1]], # cat head P5
         [-1, 3, C3, [1024, False]], # 23 (P5/32-large)

         [[17, 20, 23], 1, Detect, [nc, anchors]], # Detect(P3, P4, P5)
       ]


#customize iPython writefile so we can write variables
from IPython.core.magic import register_line_cell_magic

@register_line_cell_magic
def writetemplate(line, cell):
    with open(line, 'w') as f:
        f.write(cell.format(**globals()))


%%writetemplate /content/yolov5/models/custom_yolov5l.yaml

# parameters
nc: {num_classes}  # number of classes
depth_multiple: 1.0  # model depth multiple
width_multiple: 1.0  # layer channel multiple

# anchors
anchors:
  - [10,13, 16,30, 33,23]  # P3/8
  - [30,61, 62,45, 59,119]  # P4/16
  - [116,90, 156,198, 373,326]  # P5/32

# YOLOv5 backbone
backbone:
  # [from, number, module, args]
  [[-1, 1, Focus, [64, 3]],  # 0-P1/2
   [-1, 1, Conv, [128, 3, 2]],  # 1-P2/4
   [-1, 3, C3, [128]],
   [-1, 1, Conv, [256, 3, 2]],  # 3-P3/8
   [-1, 9, C3, [256]],
   [-1, 1, Conv, [512, 3, 2]],  # 5-P4/16
   [-1, 9, C3, [512]],
   [-1, 1, Conv, [1024, 3, 2]],  # 7-P5/32
   [-1, 1, SPP, [1024, [5, 9, 13]]],
   [-1, 3, C3, [1024, False]],  # 9
  ]

# YOLOv5 head
head:
  [[-1, 1, Conv, [512, 1, 1]],
   [-1, 1, nn.Upsample, [None, 2, 'nearest']],
   [[-1, 6], 1, Concat, [1]],  # cat backbone P4
   [-1, 3, C3, [512, False]],  # 13

   [-1, 1, Conv, [256, 1, 1]],
   [-1, 1, nn.Upsample, [None, 2, 'nearest']],
   [[-1, 4], 1, Concat, [1]],  # cat backbone P3
   [-1, 3, C3, [256, False]],  # 17 (P3/8-small)

   [-1, 1, Conv, [256, 3, 2]],
   [[-1, 14], 1, Concat, [1]],  # cat head P4
   [-1, 3, C3, [512, False]],  # 20 (P4/16-medium)

   [-1, 1, Conv, [512, 3, 2]],
   [[-1, 10], 1, Concat, [1]],  # cat head P5
   [-1, 3, C3, [1024, False]],  # 23 (P5/32-large)
```

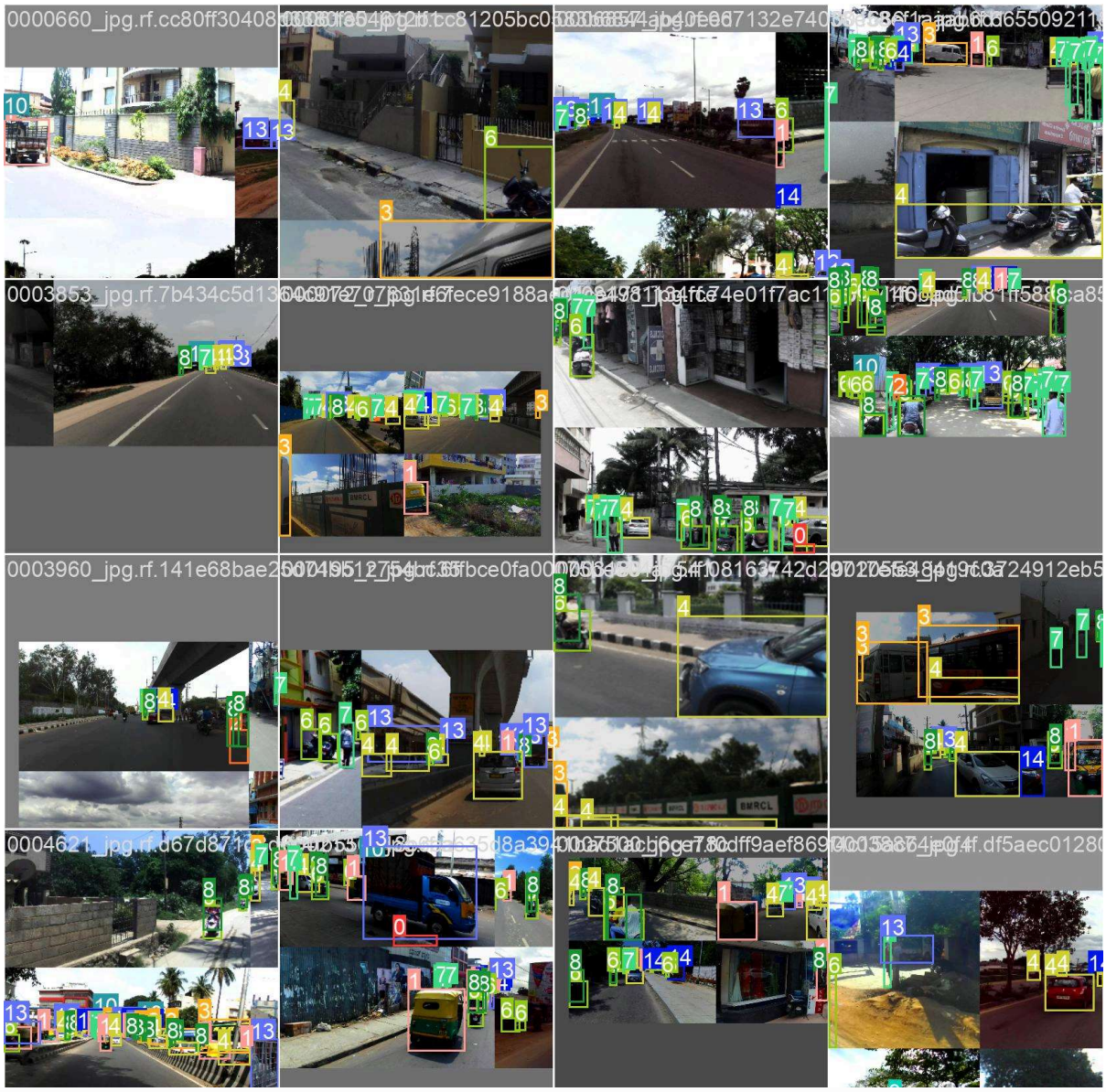
```
  [[17, 20, 23], 1, Detect, [nc, anchors]],  # Detect(P3, P4, P5)
 ]
```

```
%%time
%cd /content/yolov5/
!python train.py --img 416 --batch 64 --epochs 40 --data '../data.yaml' --cfg ./models/custom_yolov5l.yaml --weights '' --name yolov5l_resul
```

```
/content/yolov5
2024-05-08 12:38:22.585108: E external/local_xla/xla/stream_executor/cuda/cuda_dnn.cc:9261] Unable to register cuDNN factory: Attempt
2024-05-08 12:38:22.585169: E external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:607] Unable to register cuFFT factory: Attempti
2024-05-08 12:38:22.732789: E external/local_xla/xla/stream_executor/cuda/cuda_blas.cc:1515] Unable to register cuBLAS factory: Attem
train: weights=, cfg=./models/custom_yolov5l.yaml, data=../data.yaml, hyp=data/hyps/hyp.scratch-low.yaml, epochs=40, batch_size=64, i
github: up to date with https://github.com/ultralytics/yolov5 ✅
YOLOv5 🚀 v7.0-307-g920c721e Python-3.10.12 torch-2.2.1+cu121 CUDA:0 (Tesla T4, 15102MiB)

hyperparameters: lr0=0.01, lrf=0.01, momentum=0.937, weight_decay=0.0005, warmup_epochs=3.0, warmup_momentum=0.8, warmup_bias_lr=0.1,
Comet: run 'pip install comet_ml' to automatically track and visualize YOLOv5 🚀 runs in Comet
TensorBoard: Start with 'tensorboard --logdir runs/train', view at http://localhost:6006/
Downloading https://ultralytics.com/assets/Arial.ttf to /root/.config/Ultralytics/Arial.ttf...
100% 755k/755k [00:00<00:00, 17.4MB/s]

                 from  n    params  module                                  arguments
  0                -1  1      7040  models.common.Focus                     [3, 64, 3]
  1                -1  1     73984  models.common.Conv                      [64, 128, 3, 2]
  2                -1  3    156928  models.common.C3                        [128, 128, 3]
  3                -1  1    295424  models.common.Conv                      [128, 256, 3, 2]
  4                -1  9   1611264  models.common.C3                        [256, 256, 9]
  5                -1  1   1180672  models.common.Conv                      [256, 512, 3, 2]
  6                -1  9   6433792  models.common.C3                        [512, 512, 9]
  7                -1  1   4720640  models.common.Conv                      [512, 1024, 3, 2]
  8                -1  1   2624512  models.common.SPP                       [1024, 1024, [5, 9, 13]]
  9                -1  3   9971712  models.common.C3                        [1024, 1024, 3, False]
 10                -1  1    525312  models.common.Conv                      [1024, 512, 1, 1]
 11                -1  1         0  torch.nn.modules.upsampling.Upsample    [None, 2, 'nearest']
 12           [-1, 6]  1         0  models.common.Concat                    [1]
 13                -1  3   2757632  models.common.C3                        [1024, 512, 3, False]
 14                -1  1    131584  models.common.Conv                      [512, 256, 1, 1]
 15                -1  1         0  torch.nn.modules.upsampling.Upsample    [None, 2, 'nearest']
 16           [-1, 4]  1         0  models.common.Concat                    [1]
 17                -1  3    690688  models.common.C3                        [512, 256, 3, False]
 18                -1  1    590336  models.common.Conv                      [256, 256, 3, 2]
 19          [-1, 14]  1         0  models.common.Concat                    [1]
 20                -1  3   2495488  models.common.C3                        [512, 512, 3, False]
 21                -1  1   2360320  models.common.Conv                      [512, 512, 3, 2]
 22          [-1, 10]  1         0  models.common.Concat                    [1]
 23                -1  3   9971712  models.common.C3                        [1024, 1024, 3, False]
 24      [17, 20, 23]  1    107700  models.yolo.Detect                      [15, [[10, 13, 16, 30, 33, 23], [30, 61, 62, 45, 59, 119]
custom_YOLOv5l summary: 393 layers, 46706740 parameters, 46706740 gradients

AMP: checks passed ✅
optimizer: SGD(lr=0.01) with parameter groups 107 weight(decay=0.0), 110 weight(decay=0.0005), 110 bias
albumentations: Blur(p=0.01, blur_limit=(3, 7)), MedianBlur(p=0.01, blur_limit=(3, 7)), ToGray(p=0.01), CLAHE(p=0.01, clip_limit=(1,
/usr/lib/python3.10/multiprocessing/popen_fork.py:66: RuntimeWarning: os.fork() was called. os.fork() is incompatible with multithrea
  self.pid = os.fork()
train: Scanning /content/train/labels... 3304 images, 4 backgrounds, 0 corrupt: 100% 3304/3304 [00:02<00:00, 1492.58it/s]
train: WARNING ⚠️ /content/train/images/0008384_jpg.rf.8786e28ae9e6fe0af4c55789f99edcd2.jpg: 1 duplicate labels removed
train: New cache created: /content/train/labels.cache
train: Caching images (0.9GB ram): 100% 3304/3304 [01:19<00:00, 41.43it/s]
val: Scanning /content/valid/labels... 499 images, 1 backgrounds, 0 corrupt: 100% 499/499 [00:00<00:00, 545.56it/s]
val: WARNING ⚠️ /content/valid/images/0007776_jpg.rf.536b2e534be4292f1932f81cf64aff30.jpg: 1 duplicate labels removed
val: New cache created: /content/valid/labels.cache
val: Caching images (0.1GB ram): 100% 499/499 [00:20<00:00, 24.27it/s]

AutoAnchor: 3.04 anchors/target, 0.889 Best Possible Recall (BPR). Anchors are a poor fit to dataset ⚠️, attempting to improve...
```

```
print("GROUND TRUTH TRAINING DATA:")
Image(filename='/content/yolov5/runs/train/yolov5l_results/train_batch0.jpg', width=900)
```

GROUND TRUTH TRAINING DATA:




```
%cd /content/yolov5/
```

```
/content/yolov5
```

```
!python detect.py --weights runs/train/yolov5l_results/weights/best.pt --img 416 --conf 0.4 --source ../test/images
```

```
Traceback (most recent call last):
  File "/content/yolov5/detect.py", line 46, in <module>
    from ultralytics.utils.plotting import Annotator, colors, save_one_box
ModuleNotFoundError: No module named 'ultralytics'
```

```
import glob
from IPython.display import Image, display

for imageName in glob.glob('/content/yolov5/runs/detect/exp/*.jpg'): #assuming JPG
    display(Image(filename=imageName))
    print("\n")
```

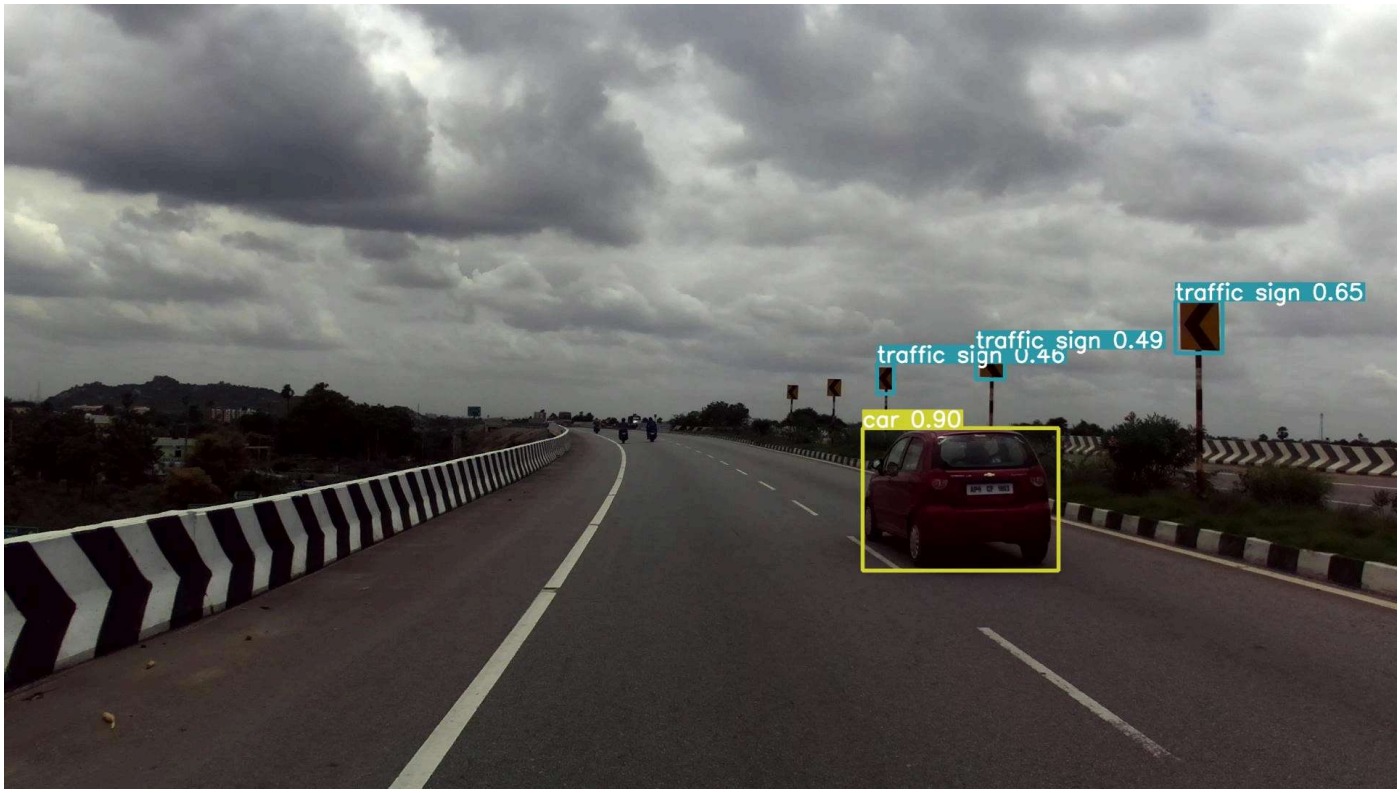
bus 0.80
TAKE DIVERSION
rider 0.42
motorcycle 0.70

rider 0.41

truck 0.55
truck 0.42
person 0.52

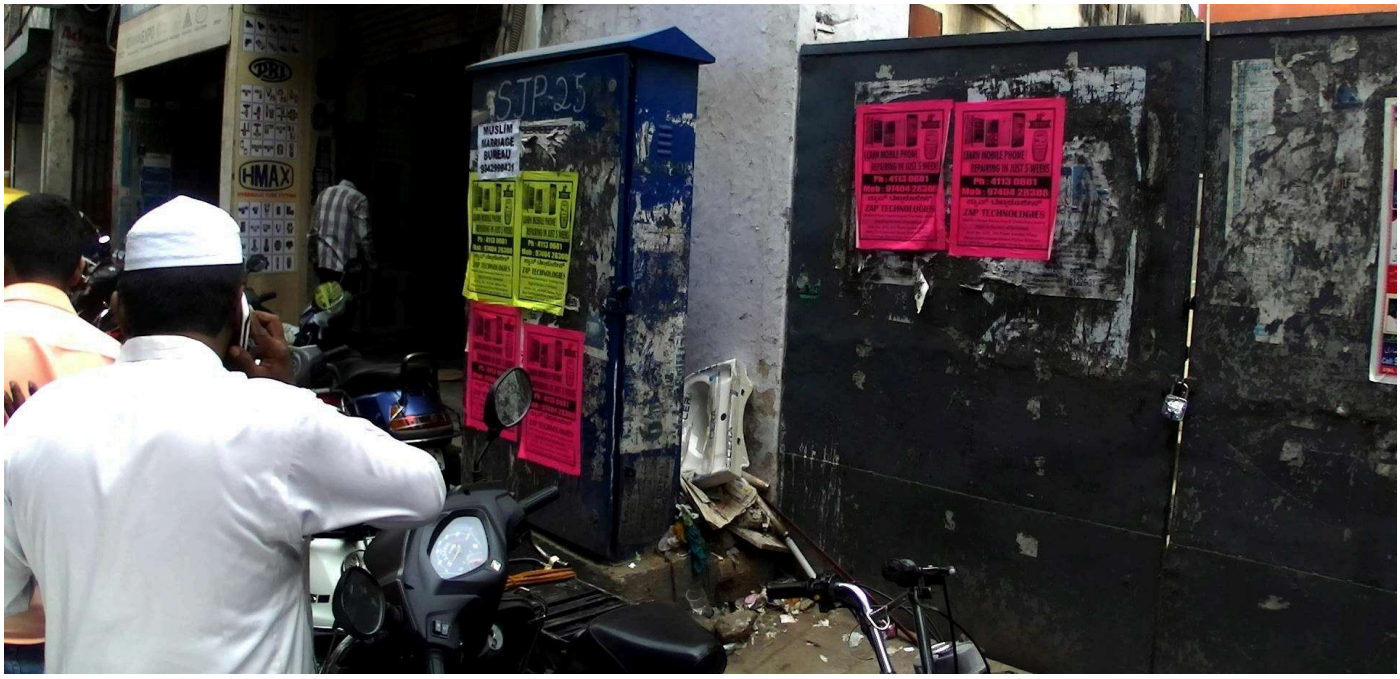
ST.DONBOSCO
HIGH SCHOOL

truck 0.58


traffic sign 0.65
traffic sign 0.49
traffic sign 0.46
car 0.90

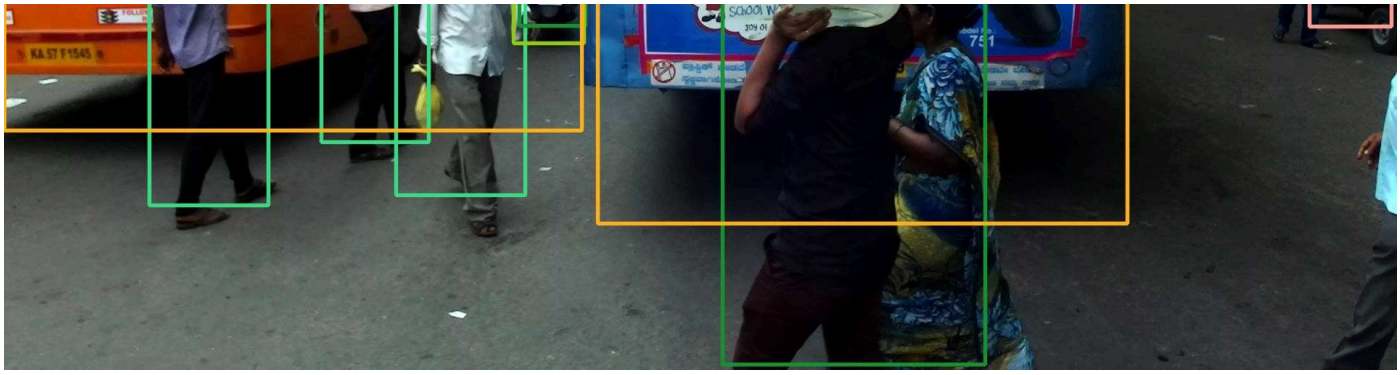
autorickshaw 0.70
person 0.45
autorickshaw 0.89
autorickshaw 0.90
motorcycle 0.63
person 0.63
Al-Zoha
Beauty Parlour
ONLY FOR LADIES

bus 0.73
bus 0.67
car 0.84
motorcycle 0.41
autorickshaw 0.93
car 0.88
car 0.46
motorcycle 0.46


HMAX
MUSLIM MARRIAGE BUREAU

autorickshaw
car 0.86
car 0.78
car 0.80
rider 0.48
motorcycle 0.45
car 0.85

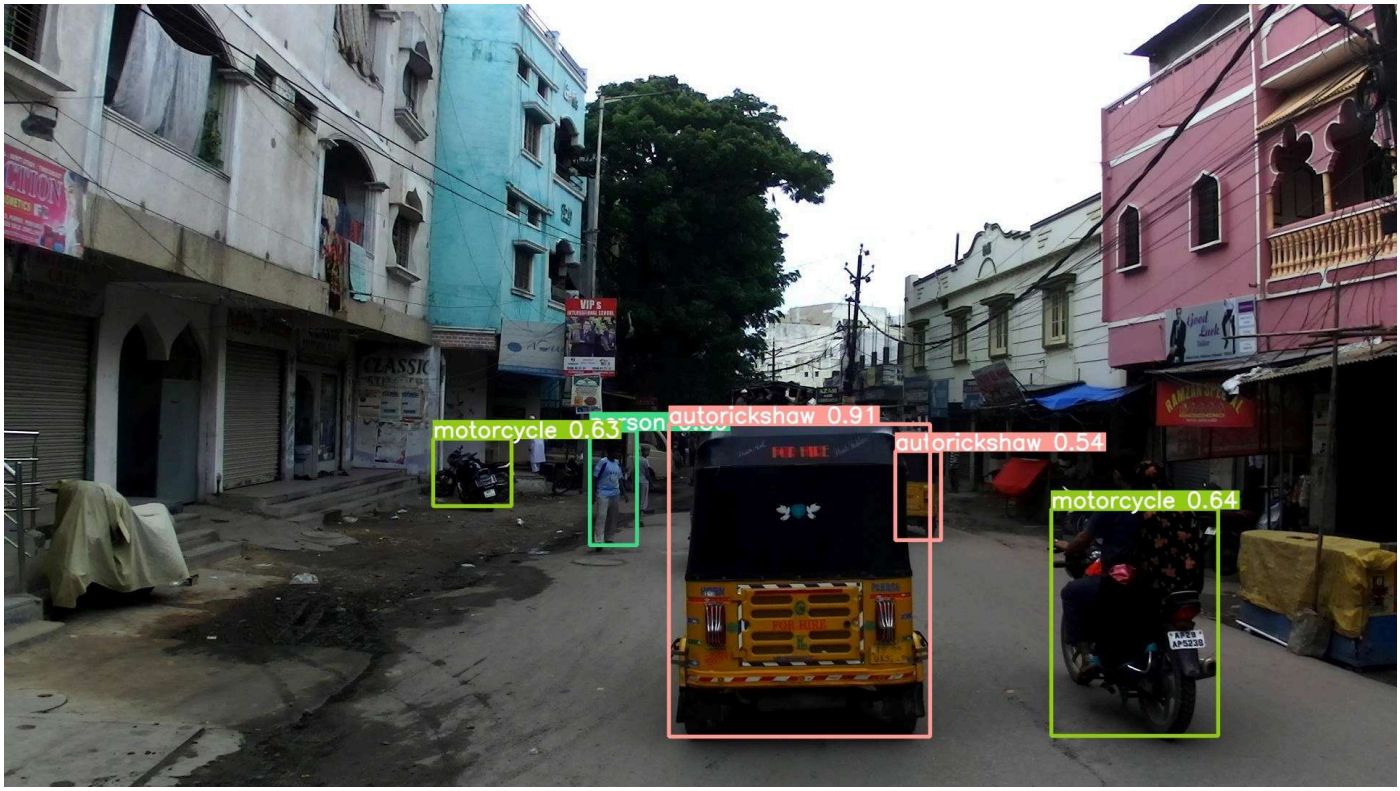
bus 0.83
bus 0.77
BMTC
person 0.64
person 0.48
person 0.41
rider 0.47
motorcycle 0.49
rider 0.58
autorick
motorcycle 0.

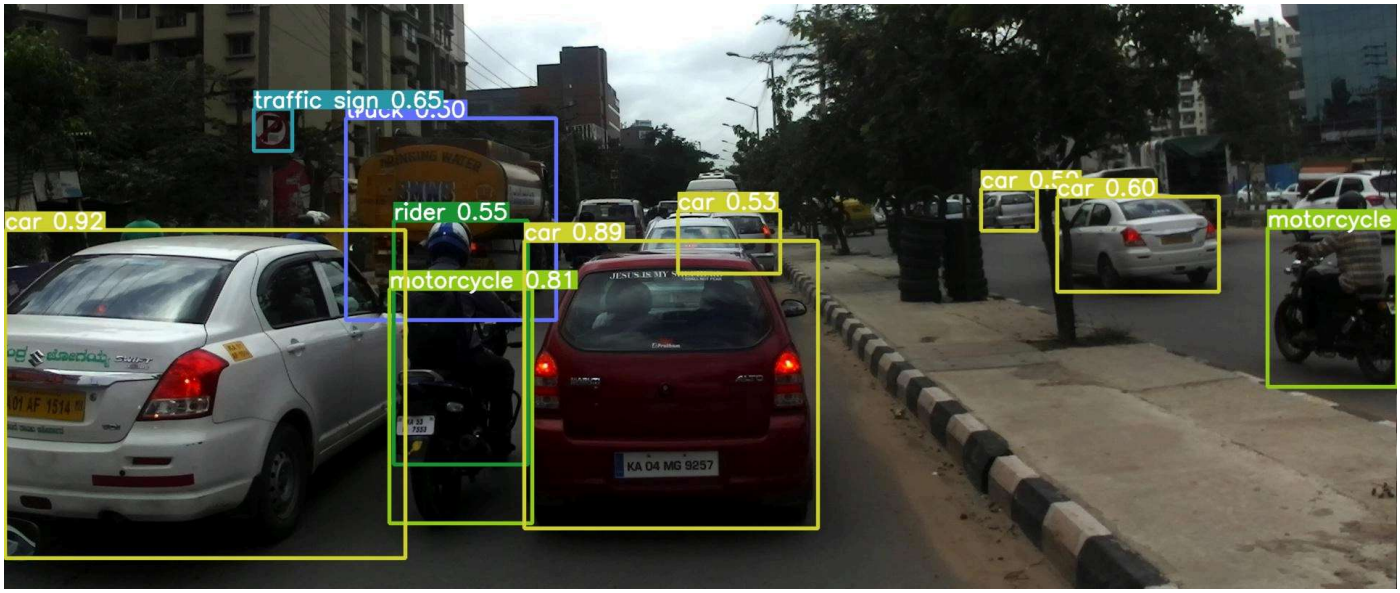


bus 0.88
bus 0.90
perperson 0.51
motorcycle 0.52

rider 0.57
motorcycle 0.74

bus 0.69
BMTC
www.mybmtc.com
Helpline No-1800 425 1663
Admissions Open
Playgroup
Nursery
Kindergarten
www.littleelly.com
Little Elly
Concept Preschool
person 0.50
rider 0.44
motorcycle 0.66
autorickshaw 0.69
bus 0.79
www.srsbook
s.R.S.
RECO

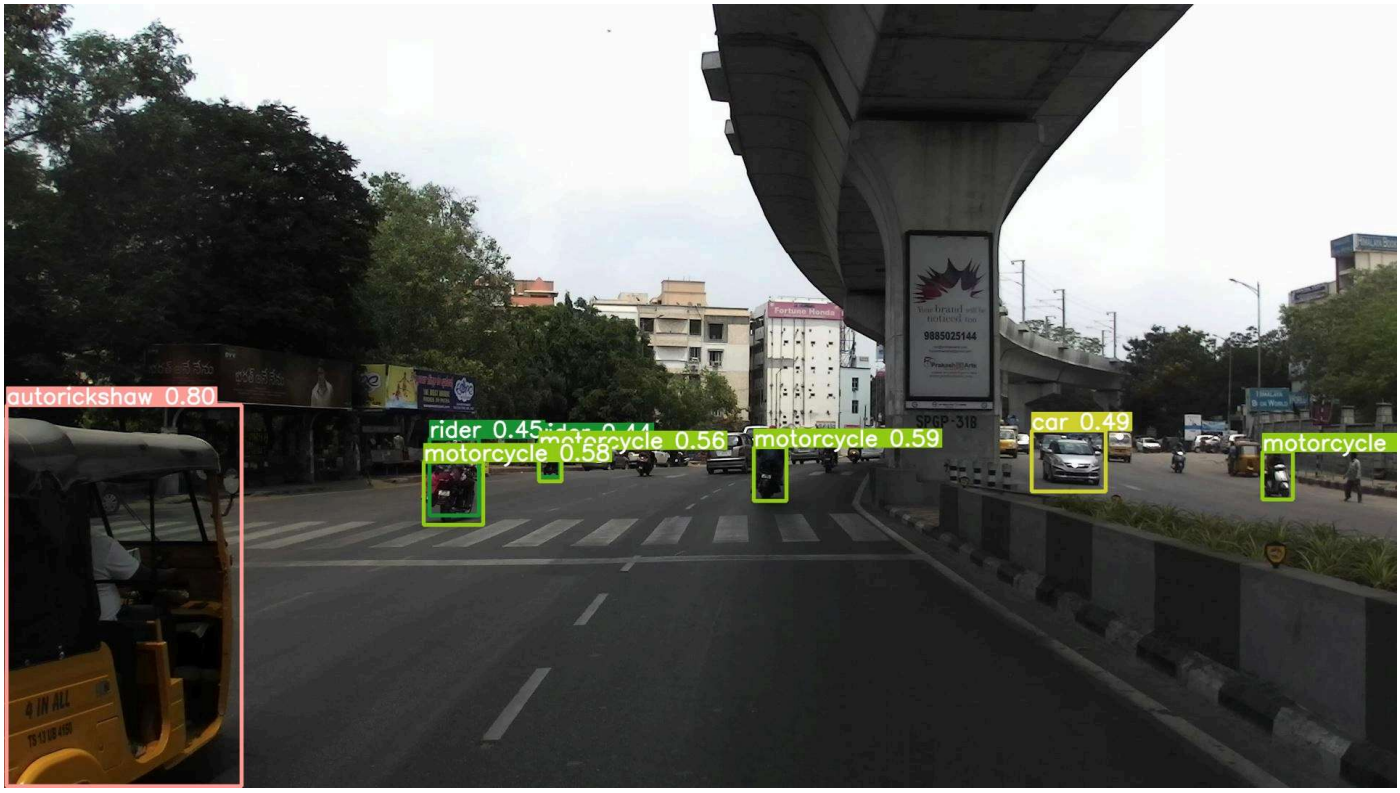

motorcycle 0.63
person
autorickshaw 0.91
autorickshaw 0.54
motorcycle 0.64
FOR HIRE
CLASSIC

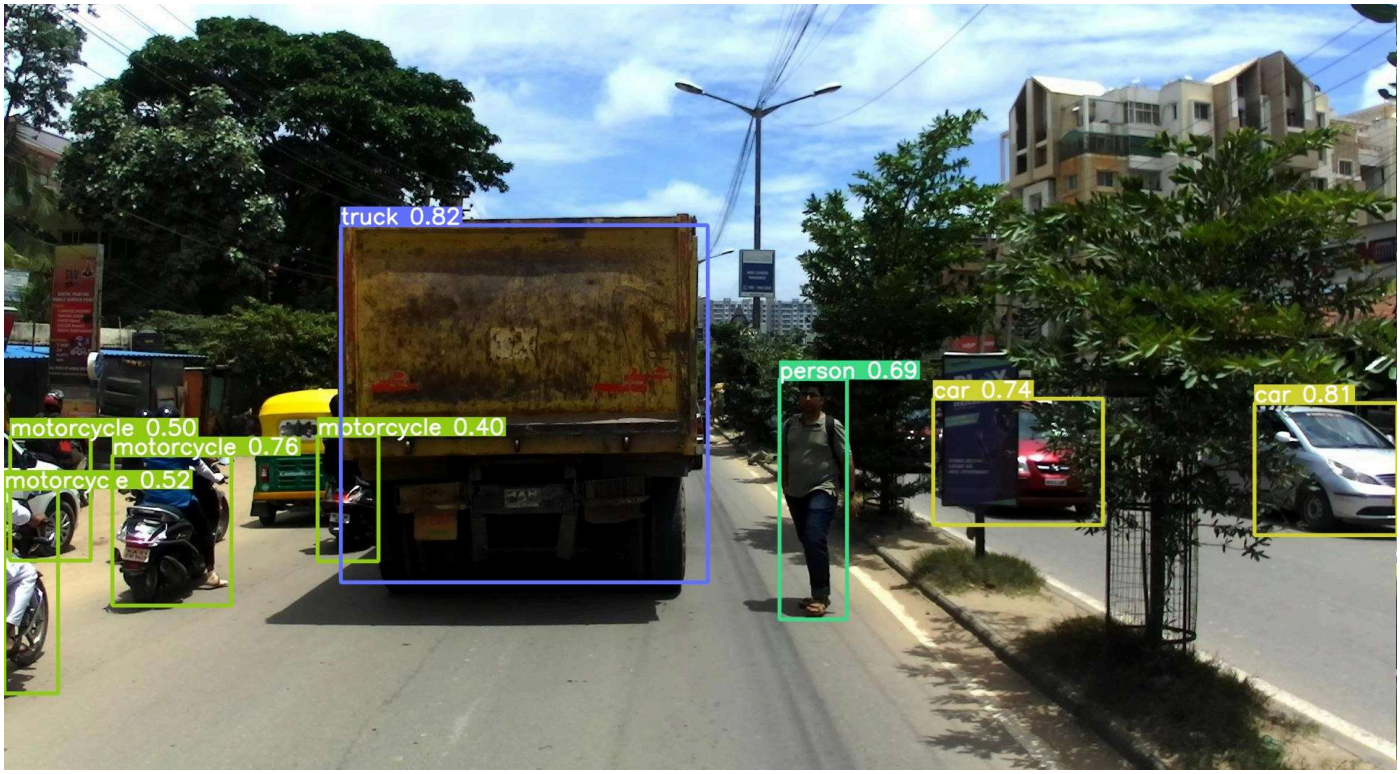

traffic sign 0.65
truck 0.50
car 0.92
rider 0.55
car 0.89
car 0.53
motorcycle 0.81
car 0.60
motorcycle

rider 0.45
car 0.87
motorcycle 0.51
truck 0.60
car 0.77

rider 0.70
motorcycle 0.67

motorcycle 0.59
SHARE
MONEY
SAVE


autorickshaw 0.80
rider 0.45
motorcycle 0.56
motorcycle 0.58
motorcycle 0.59
car 0.49
motorcycle
9885025144
SPGP-318

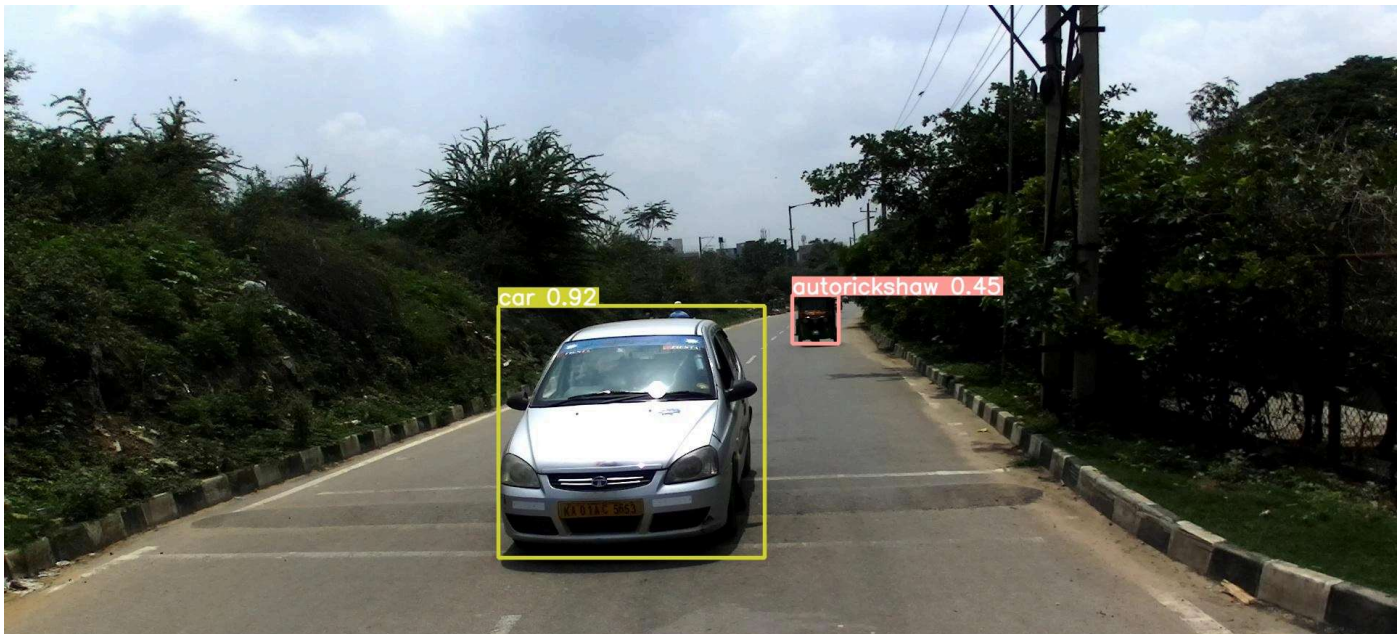

truck 0.82
person 0.69
car 0.74
car 0.81
motorcycle 0.50
motorcycle 0.76
motorcycle 0.40
motorcycle 0.52

car 0.74
Sumanapala
Gopal

PUSH YOURSELF
OR DON'T
JOCKEY
ATHLEISURE

rider 0.47
motorcycle 0.63
motorcycle 0.42
person 0.57
car 0.72
motorcycle 0.62
mo motorcycle 0.46

bus 0.66
BMTC
DIESEL
autorickshaw 0.
car 0.78
car 0


car 0.92
autorickshaw 0.45

truck 0.77
car 0.72
truck 0.66

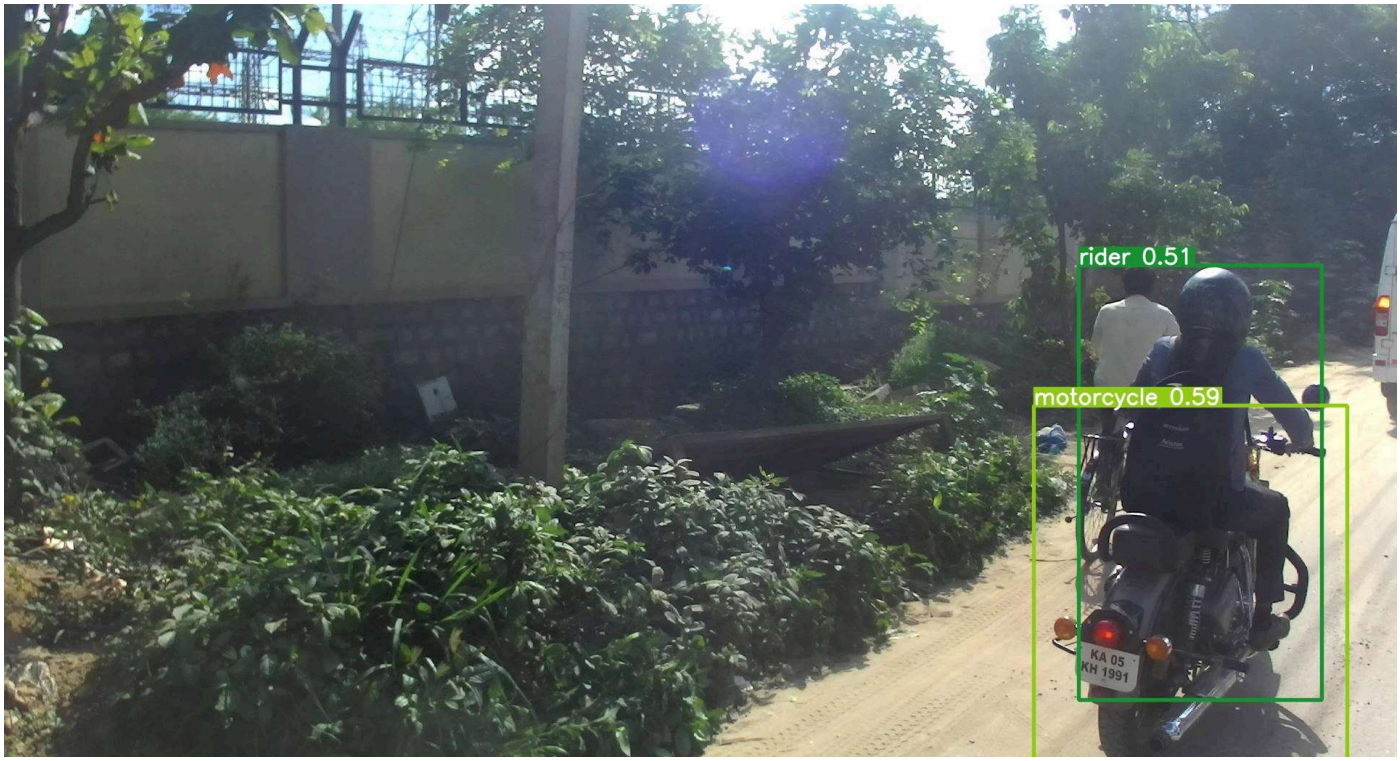
truck 0.72
car 0.93
car 0.47
motorcycle 0.45
KA03MZ1490
Admissions Op

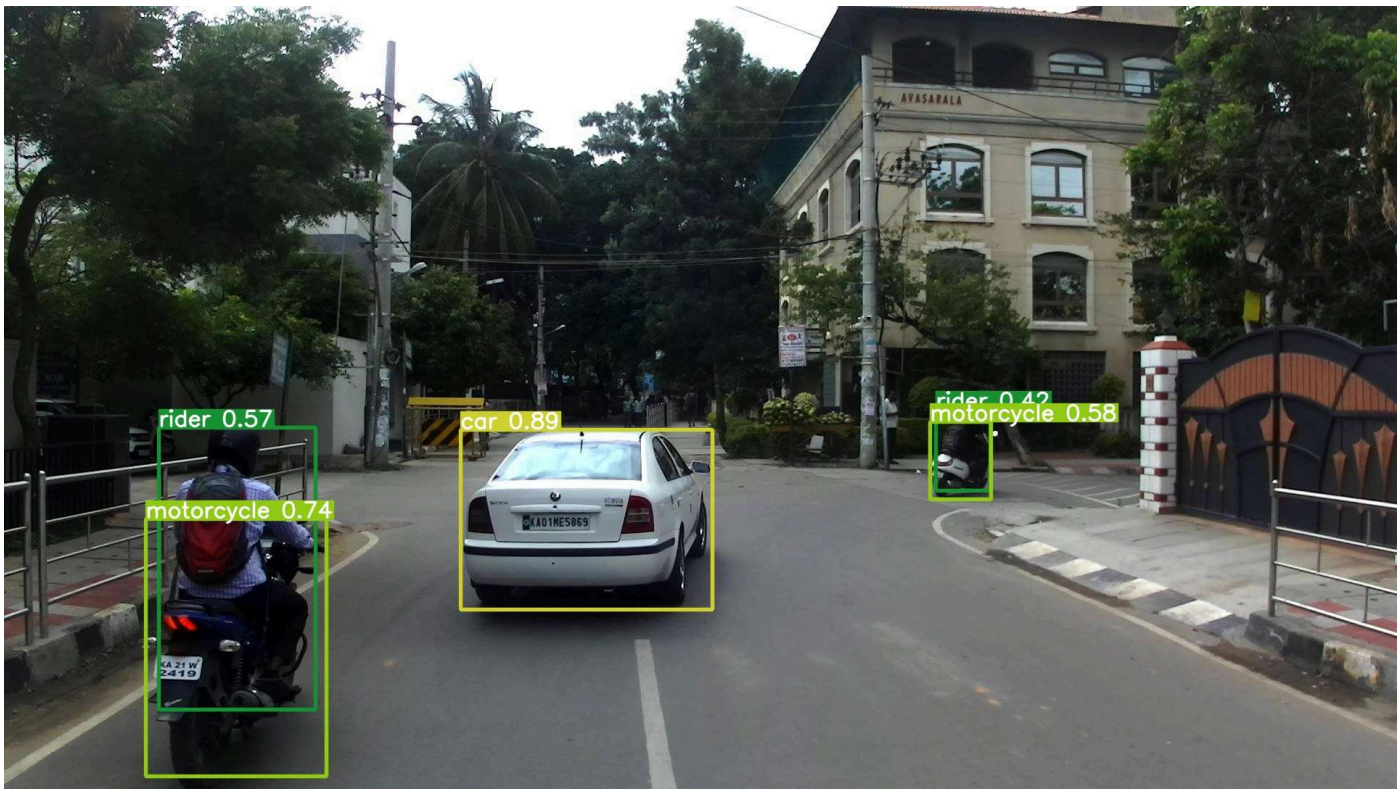
bus 0.57
travels
RECOGD. BY GO
INDIA

truck 0.85

person 0.42
person 0.55
motorcycle 0.45
motorcycle 0.68
motorcycle 0.56

S.K.PALACE
motorcycle 0.58
motorcycle 0.41
motorcycle 0.61


rider 0.51
motorcycle 0.59
KA 05
KH 1991

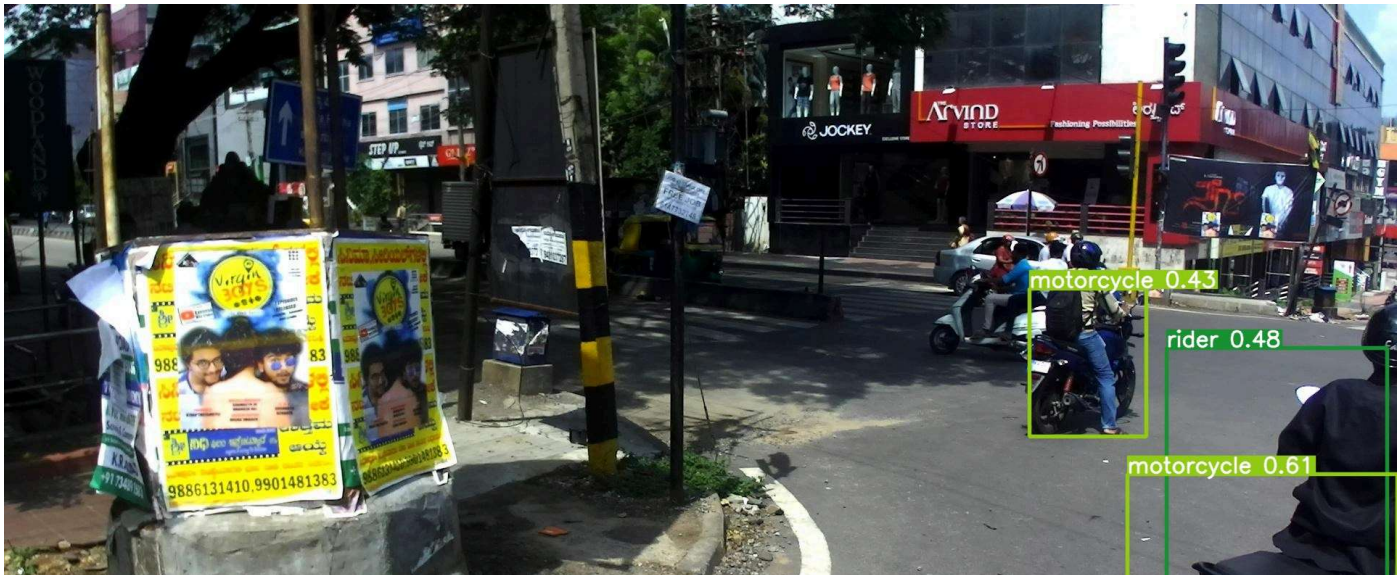

rider 0.57
motorcycle 0.74
car 0.89
rider 0.42
motorcycle 0.58
AVASARALA

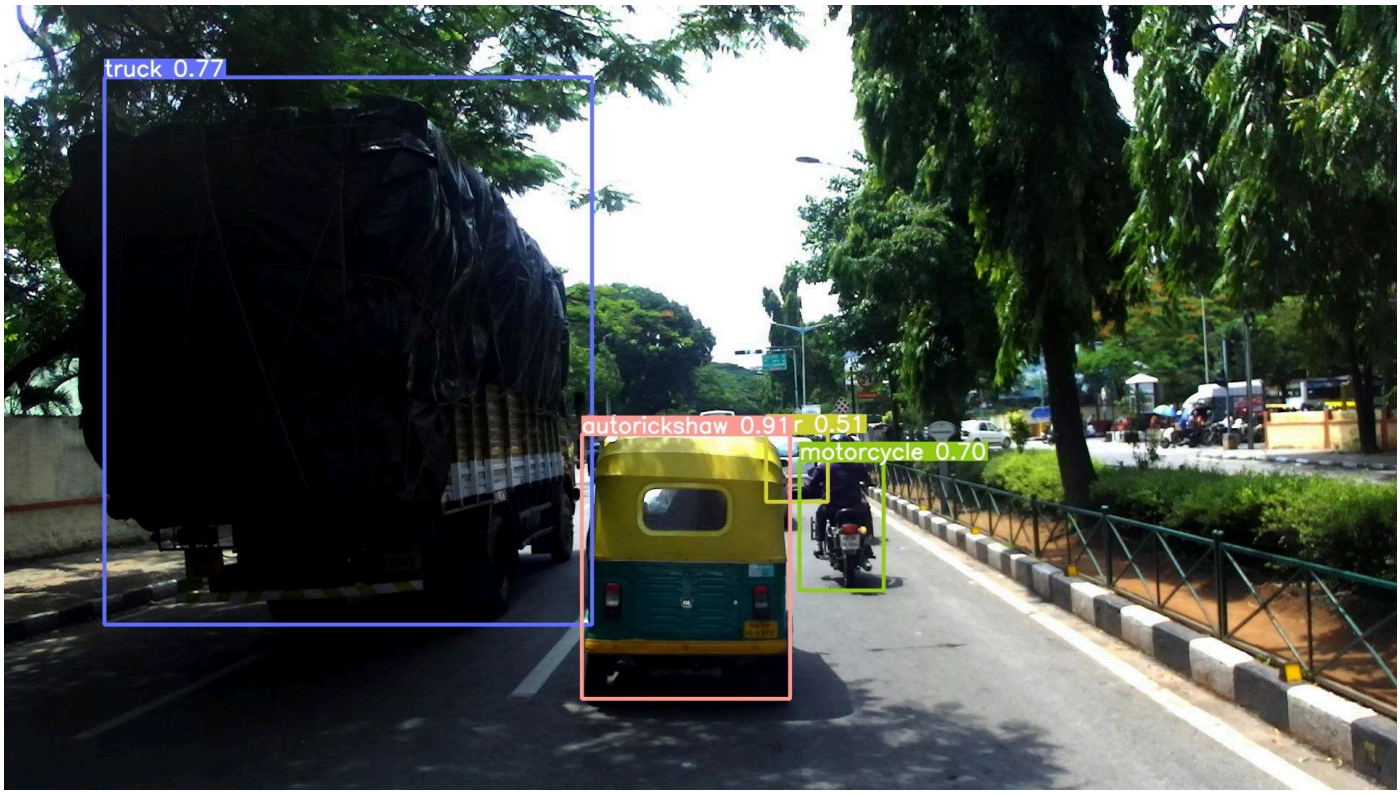
bus 0.79
car 0.88

SRI NANJUNDESHWARA BAGS
SEAT COVERS & HELMETS
Perfect Choice
For Perfect Wall..!!
ELECTRICALS
& ELECTRONICS

person 0.53

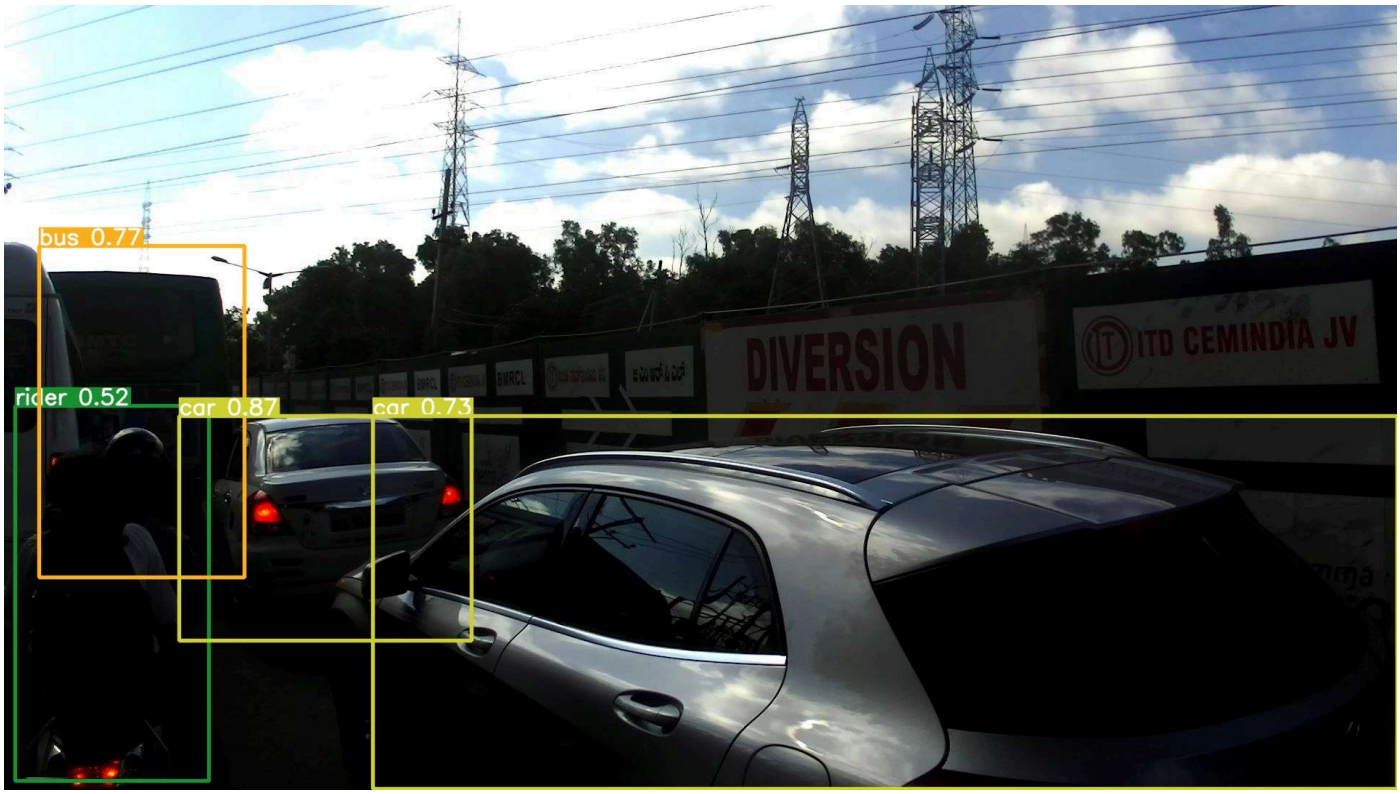
truck 0.66


JOCKEY
ARVIND
STORE
motorcycle 0.43
rider 0.48
motorcycle 0.61


truck 0.77
autorickshaw 0.91
r 0.51
motorcycle 0.70

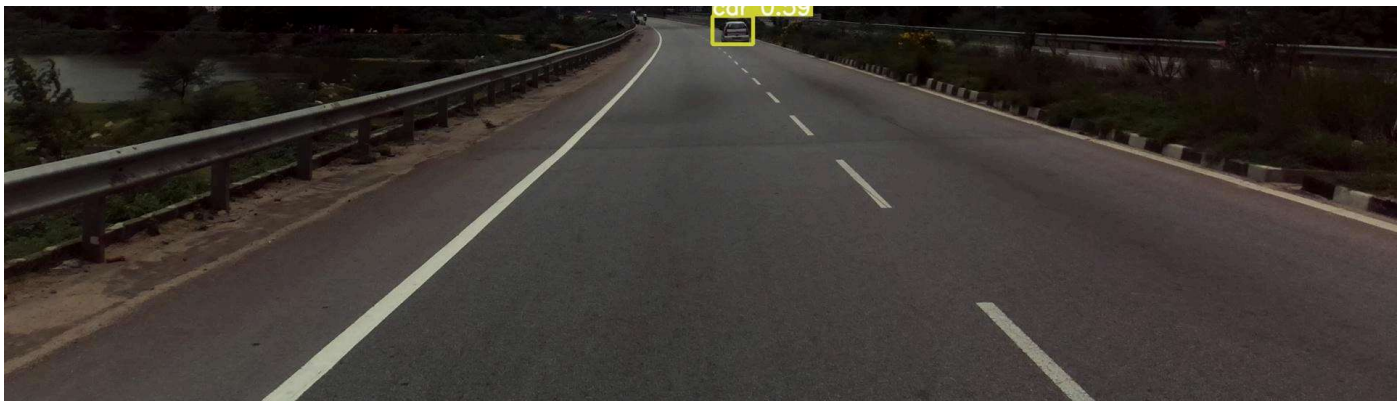
truck 0.40

motorcycle 0.60
torcycle 0.
motorcyc
car 0.49
motorcycle 0.61
autorickshaw 0.55
motorcycle 0.5

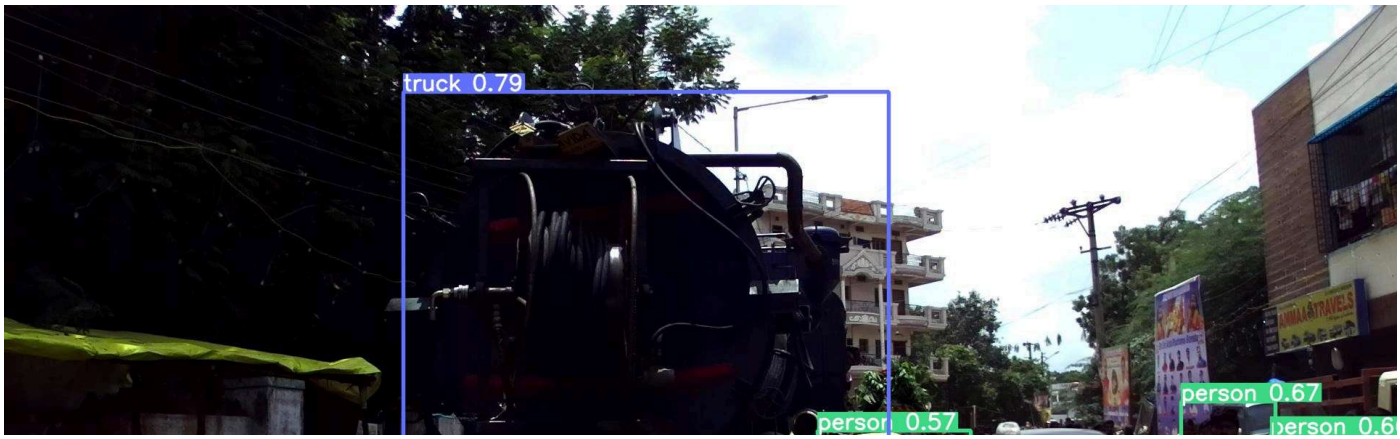

bus 0.77
rider 0.52
car 0.87
car 0.73
DIVERSION
ITD CEMINDIA JV

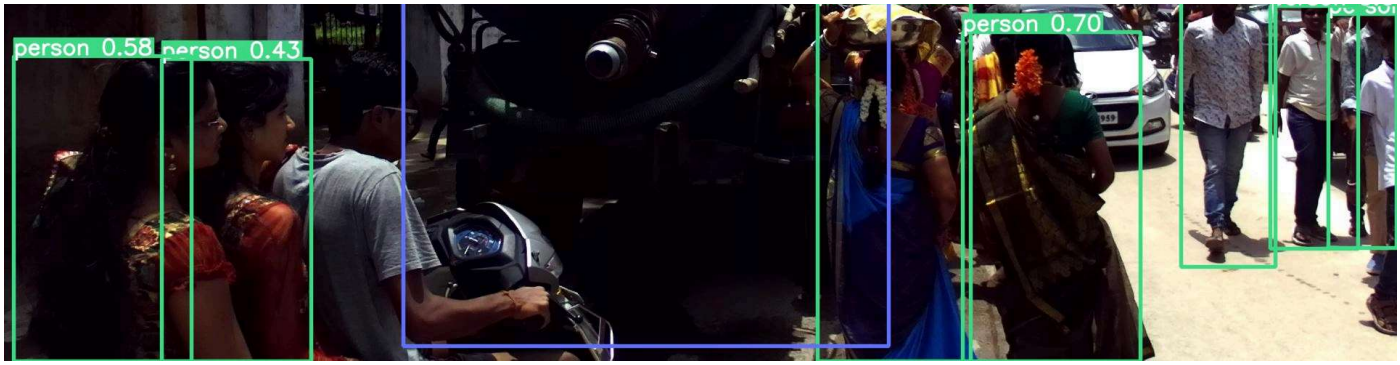
car 0.46
car 0.89
motorcycle

car 0.89
motorcycle 0.44

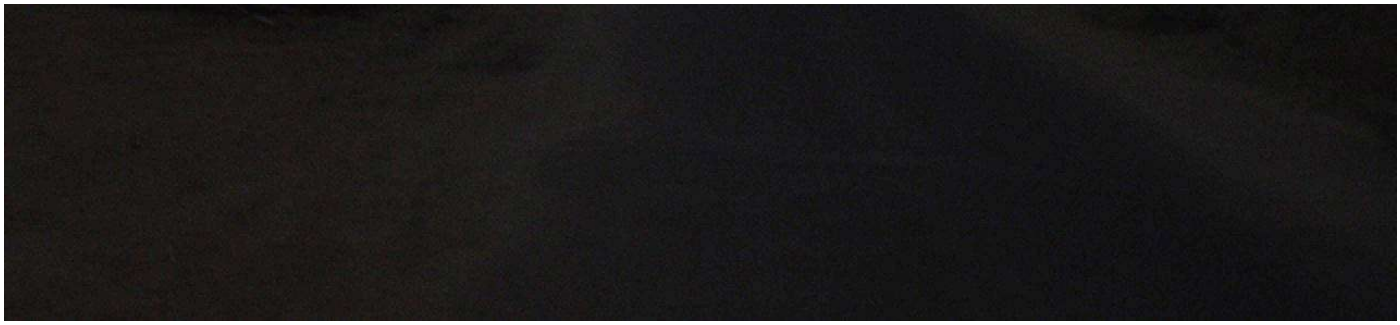


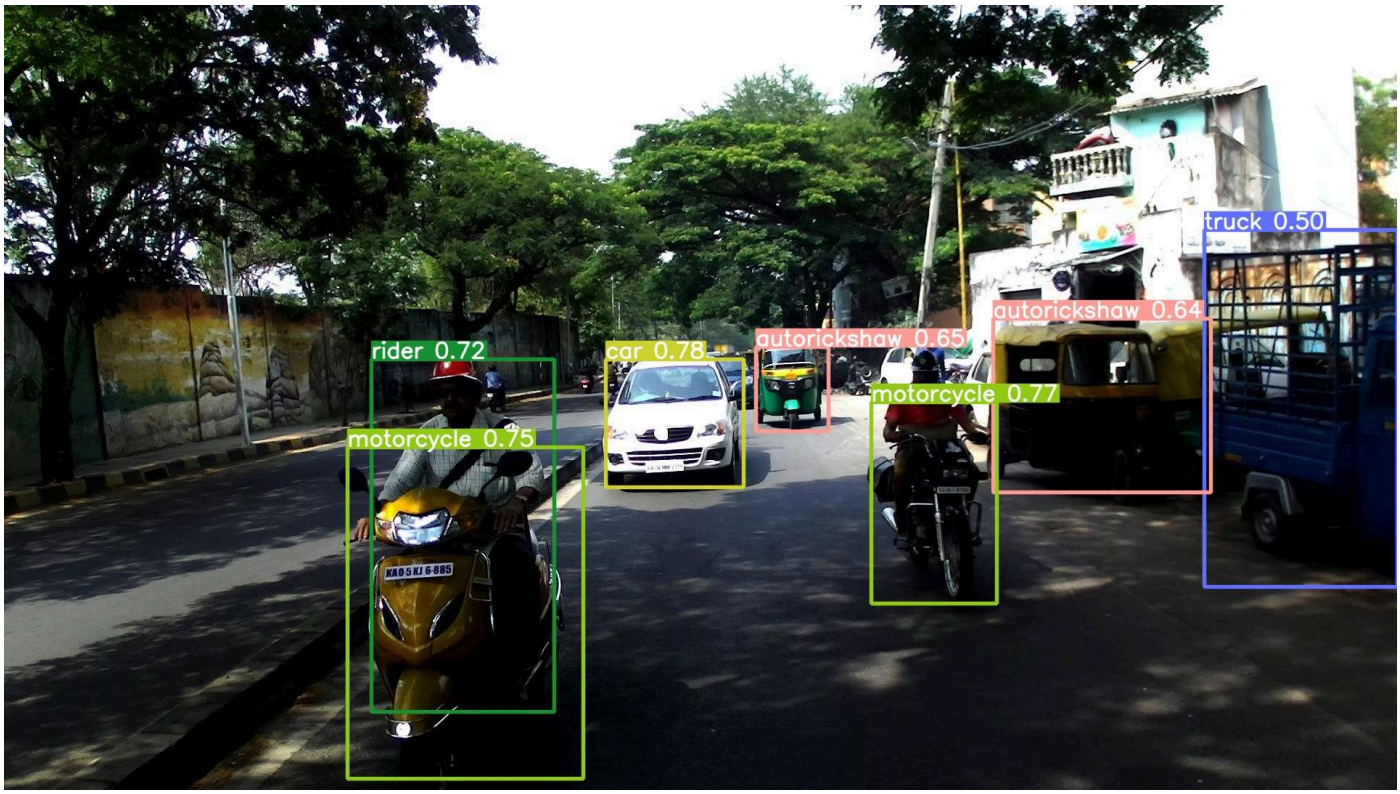
rider 0.72
person 0.45
motorcycle 0.75

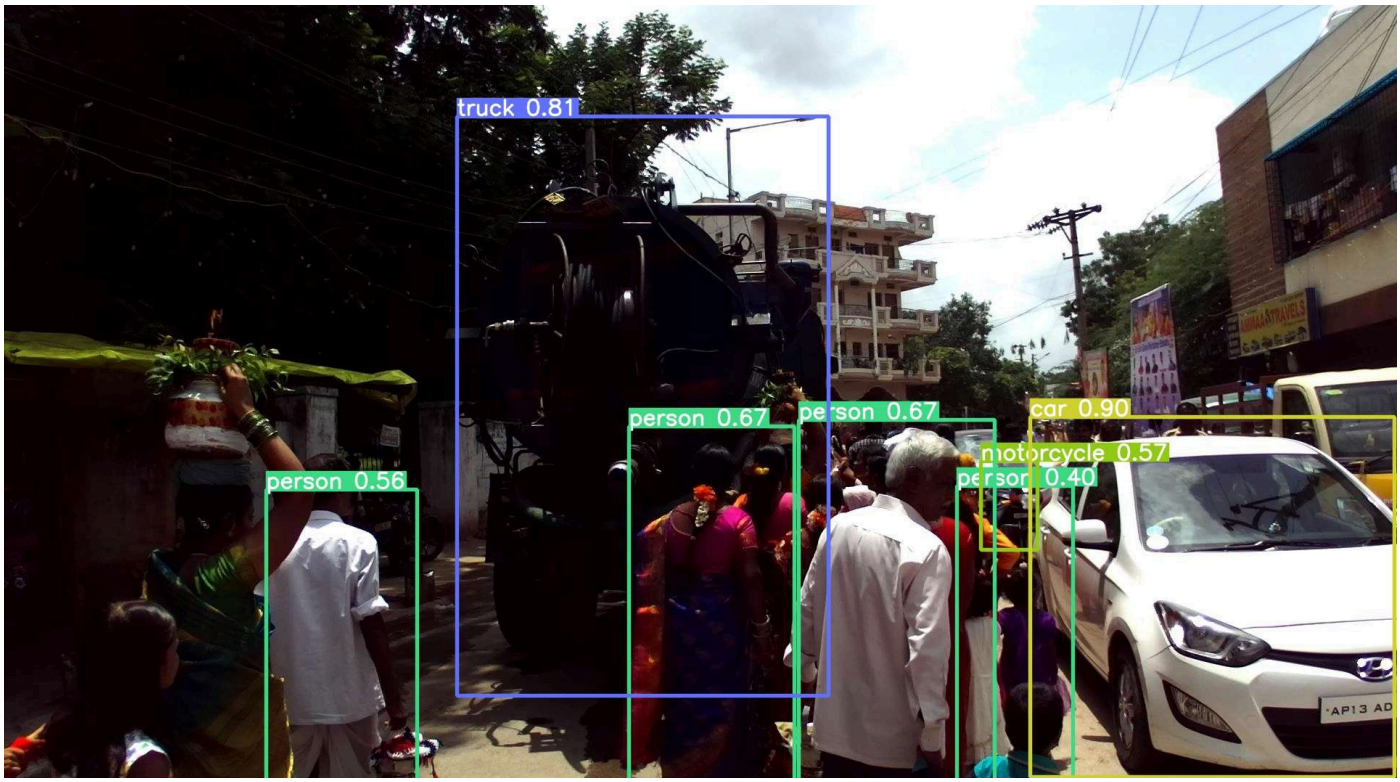

truck 0.79
person 0.67
person 0.57


person 0.58
person 0.43
person 0.70

MILK
CURD



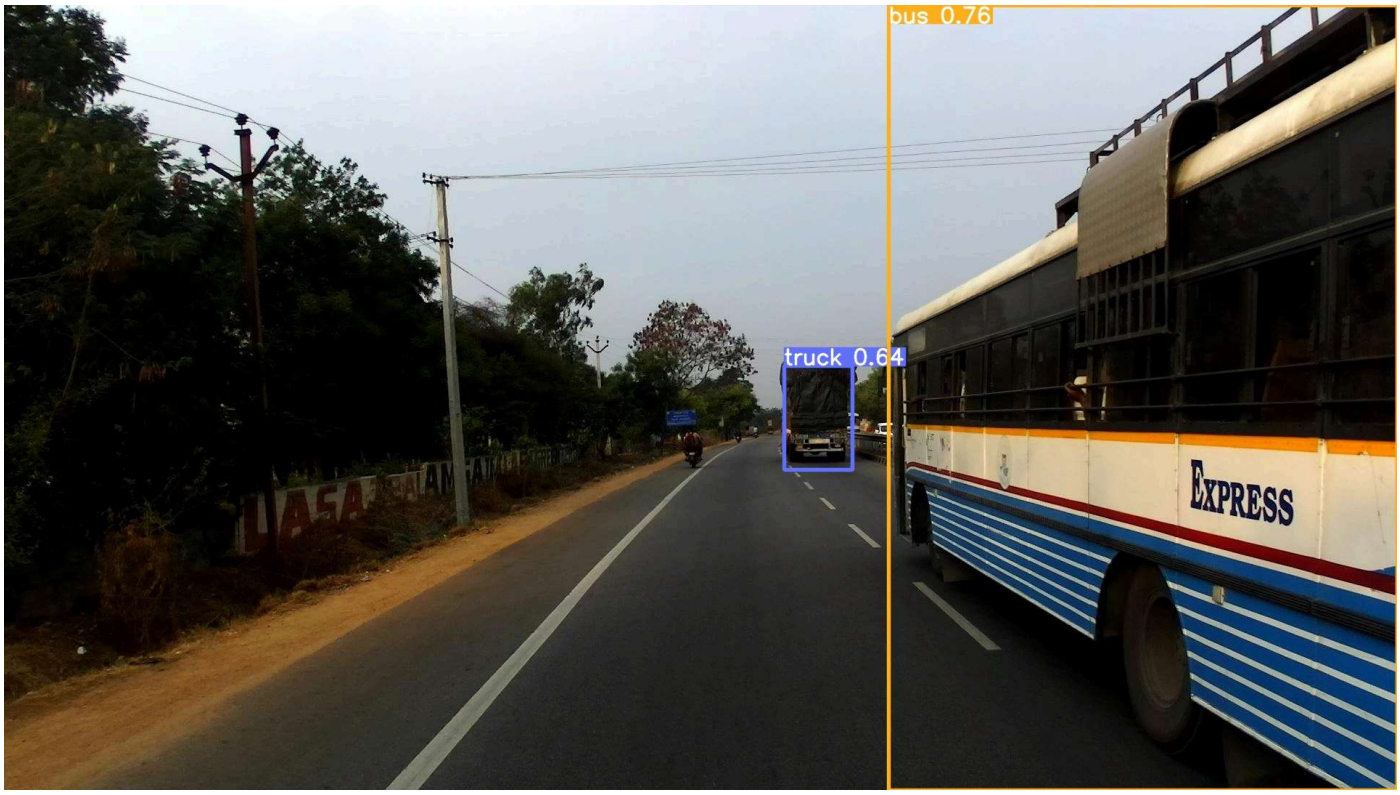

rider 0.72
motorcycle 0.75
car 0.78
autorickshaw 0.65
motorcycle 0.77
autorickshaw 0.64
truck 0.50

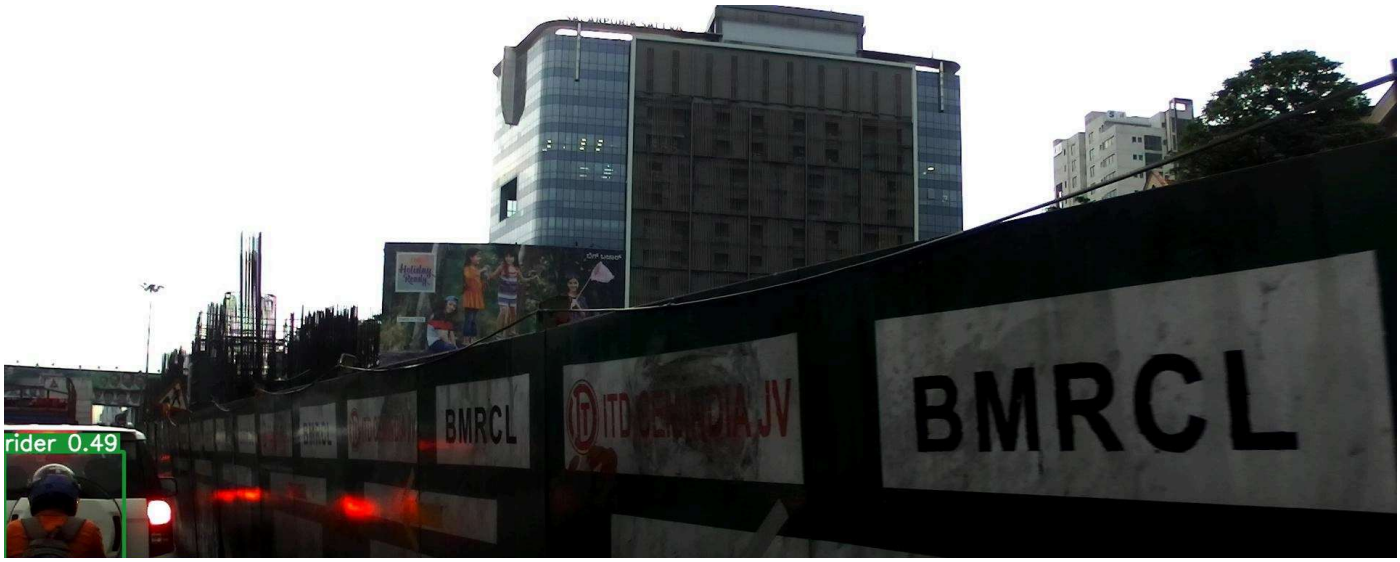

truck 0.81
person 0.56
person 0.67
person 0.67
car 0.90
motorcycle 0.57
person 0.40

bus 0.77
BMRCL

person 0.80
autoricl
rider 0.487
motorcycle

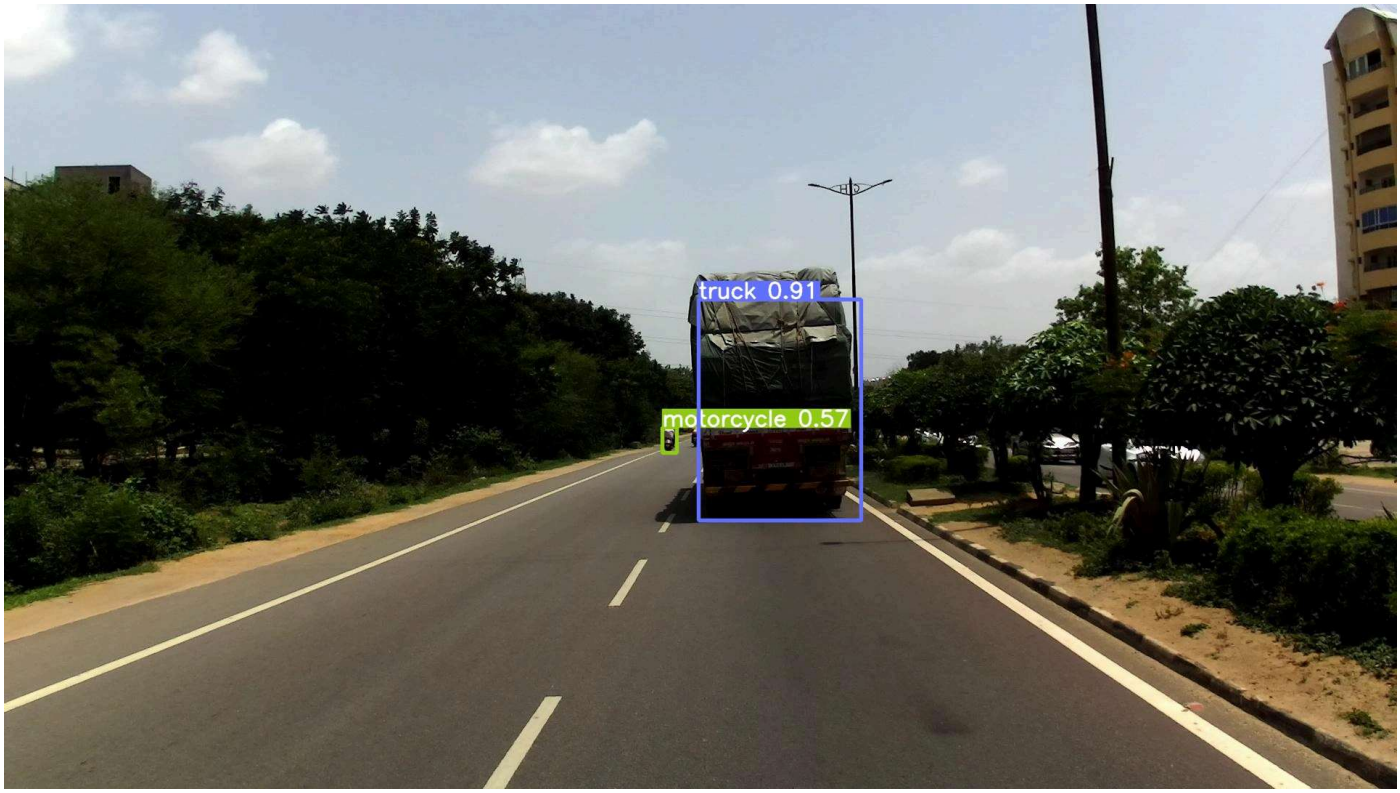
car 0.52

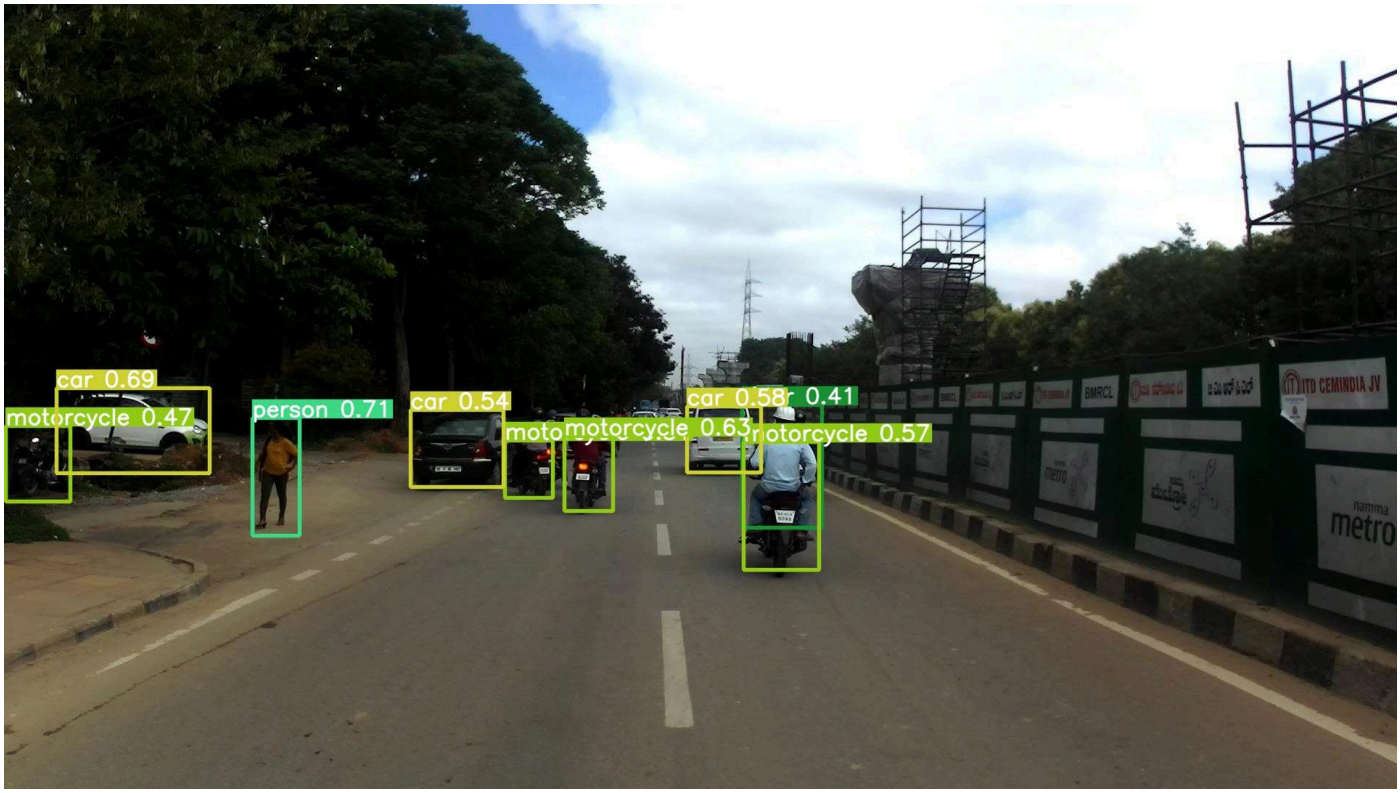

bus 0.76
truck 0.64
EXPRESS


rider 0.49
BMRCL
BMRCL

motorcycle 0.63

autorickshaw 0.42
autorickshaw 0.62
motorcycle 0.52

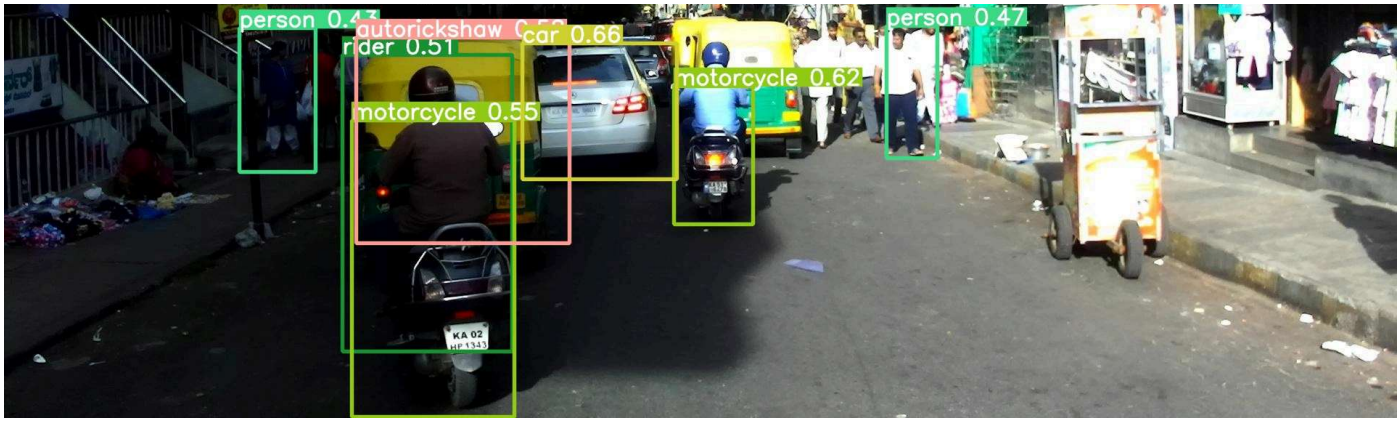

truck 0.91
motorcycle 0.57


car 0.69
motorcycle 0.47
person 0.71
car 0.54
moto
motorcycle 0.63
car 0.58
r 0.41
motorcycle 0.57
BMRCL
IITD CEMINDIA JV
namma
metro

FAQAR AMEENA GULSHAN
rider 0.43
motorcycle 0.66
autorickshaw 0.53
motorcycle 0.58
car 0.60

truck 0.57
autorickshaw 0.80
motorcycle 0.61

traffic sign 0.54
BRIGHT
NOVELTY HOUSE
RAVI GARMENT


person 0.47
autorickshaw
car 0.66
rider 0.51
motorcycle 0.62
motorcycle 0.55

bus 0.77
BMTC
www.mybmtc.com
Helpline No-1800 425 1663
VOLVO
NO HAND SIGNAL
8400
KA-57-F-1785
ITD CEMINDIA JV
BMRCL
car 0.83

CAUTION
NAMMA METRO
WORK IN
PROGRESS
GO SLOW
ITD CEMINDIA JV
car 0.47
car 0.90

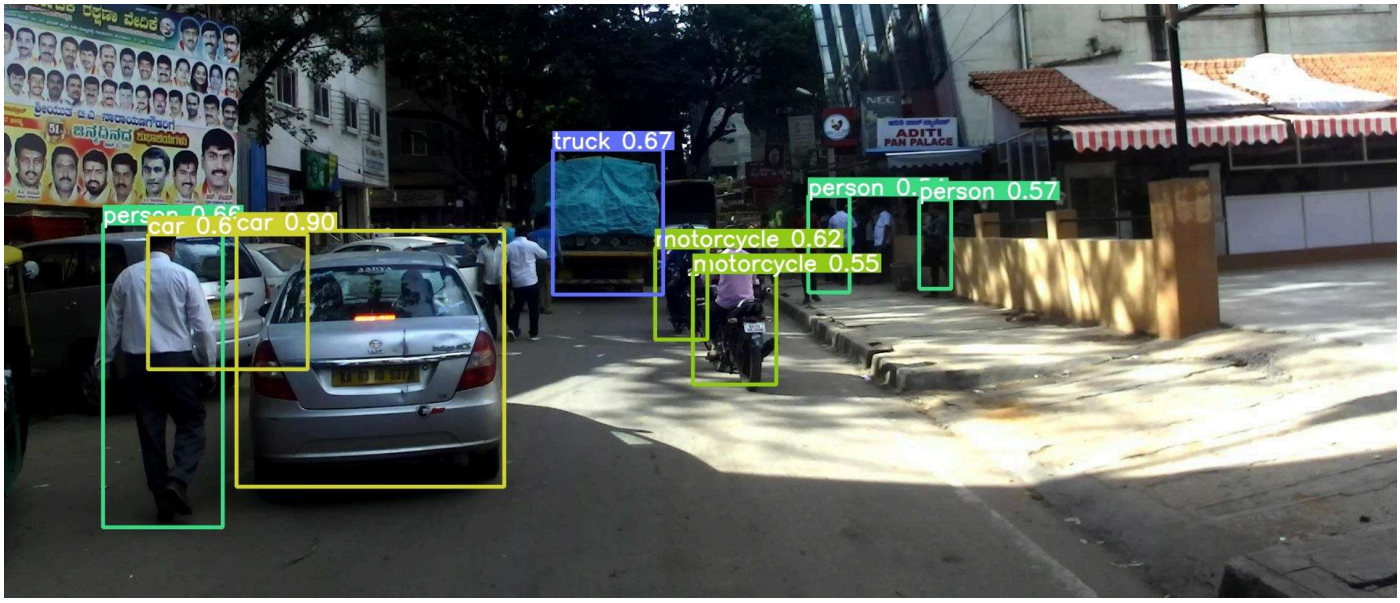
bus 0.78
bus 0.80
car 0.94
car 0.89
motorcycle 0.40
BMRCL
ITD CEMINDIA JV
namma
metro

motorcycle 0.40

bus 0.92
truck 0.40
motorcycle 0.59
motorcycle 0.62

motorcycle 0.47

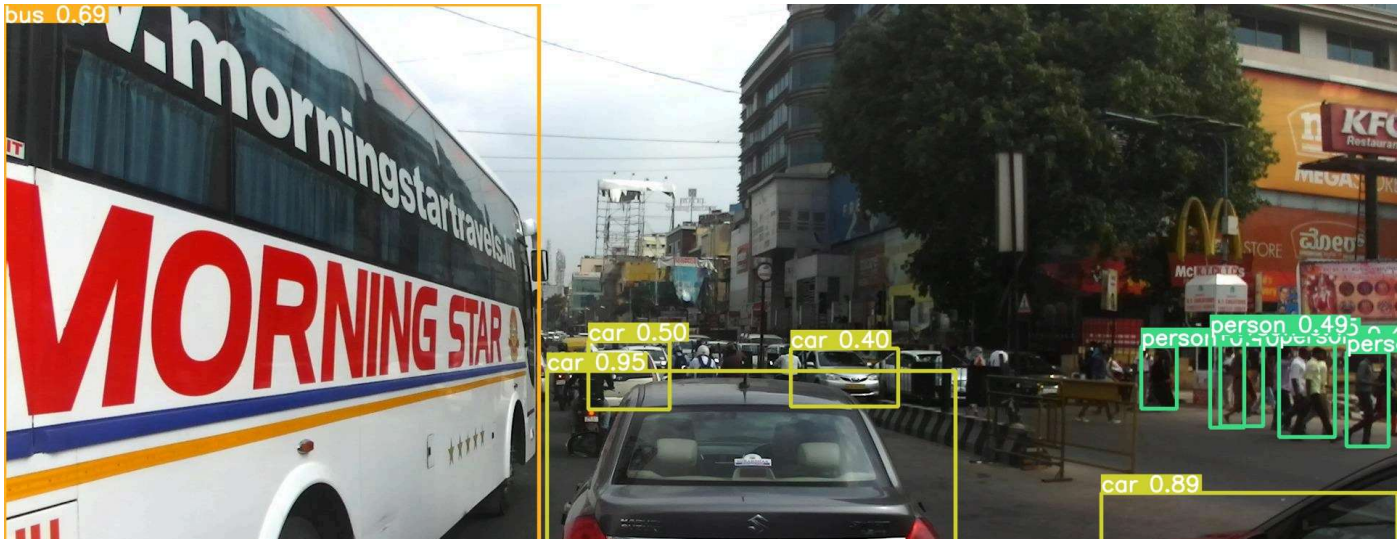
naati
KARNATAKA & ANDHRA


truck 0.67
car 0.90
motorcycle 0.62
motorcycle 0.55
person 0.57
ADITI
PAN PALACE

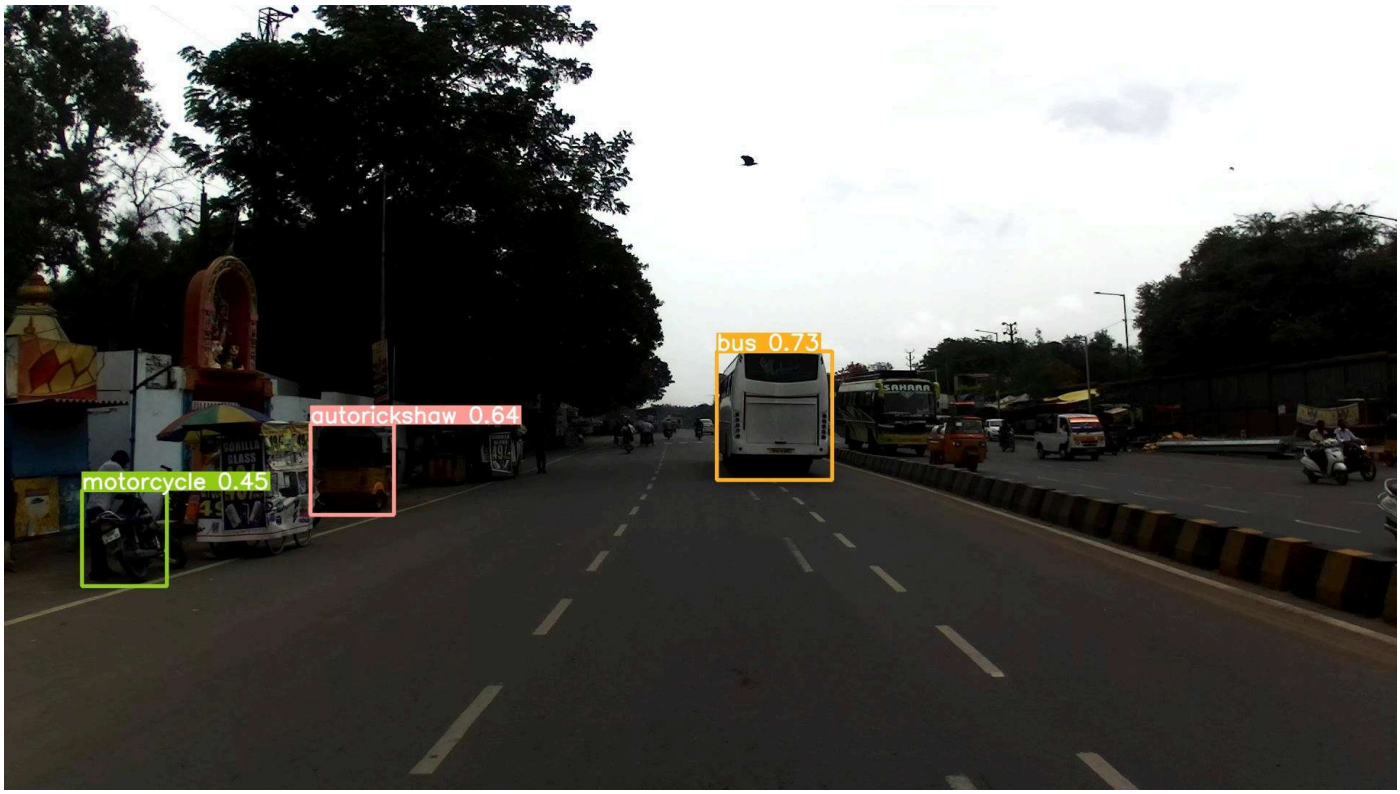
bus 0.71

car 0.56
car 0.88

person 0.64
person 0.7
car 0.86
motorcycle 0.59
motorcycle 0.52
motorcycle 0.66
TS08FR9297

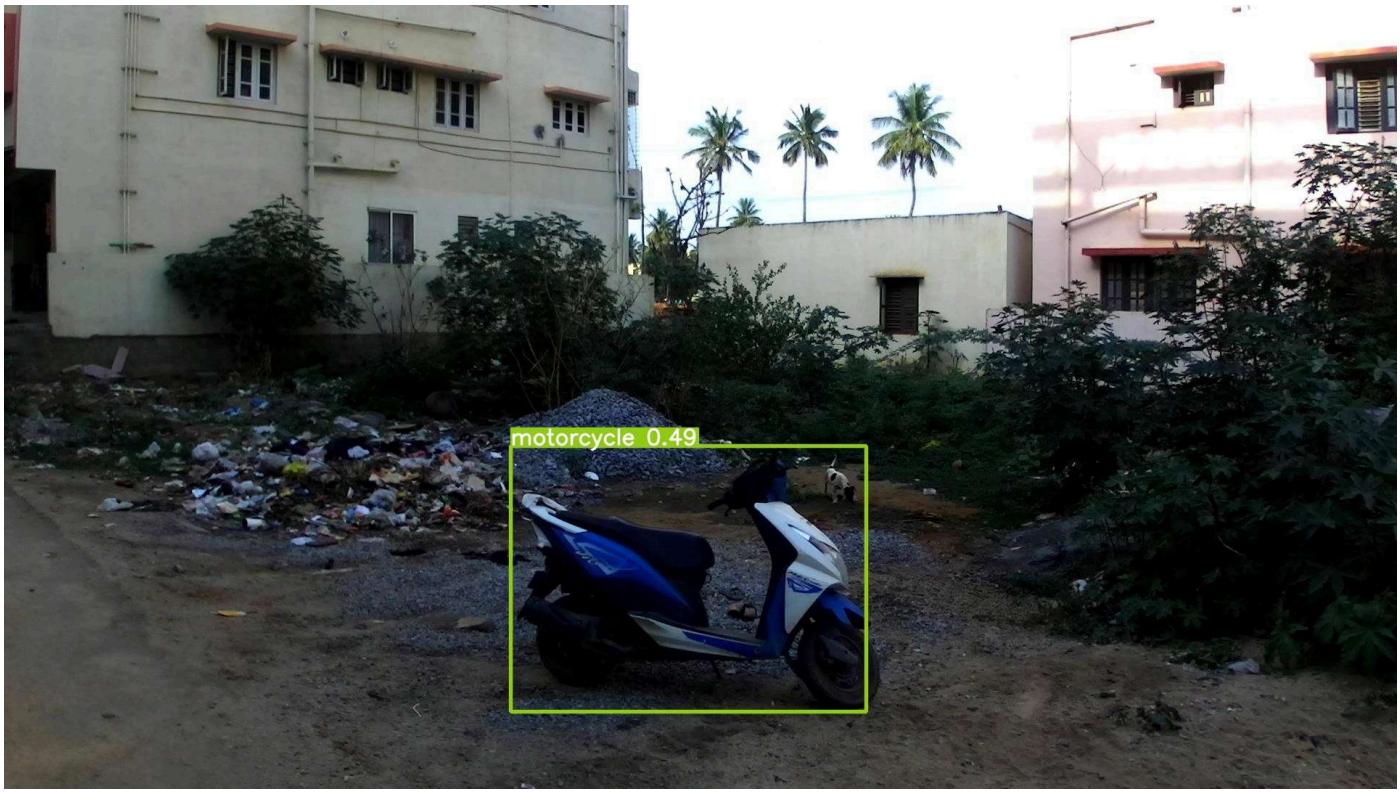

bus 0.69
MORNING STAR
car 0.50
car 0.95
car 0.40
person 0.49
car 0.89

KA 51 MF 6521


bus 0.73
autorickshaw 0.64
motorcycle 0.45

car 0.78
rider 0.41
motorcycle 0.59
motorcycle 0.61

bus 0.80
car 0.89


motorcycle 0.49

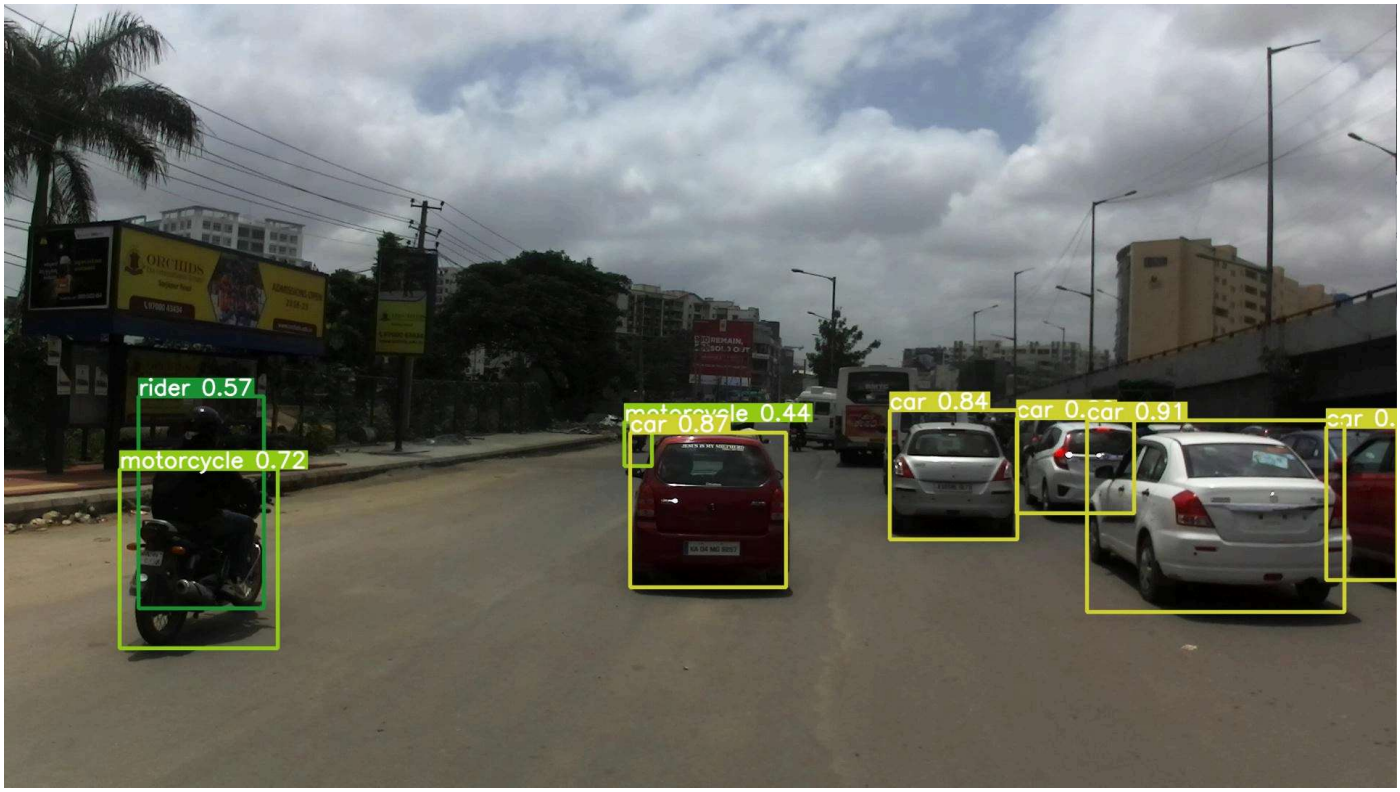
pers

GO SLOW
DIVERSION
metro
ITD CEMINDIA JV
WORK IN PROGRESS

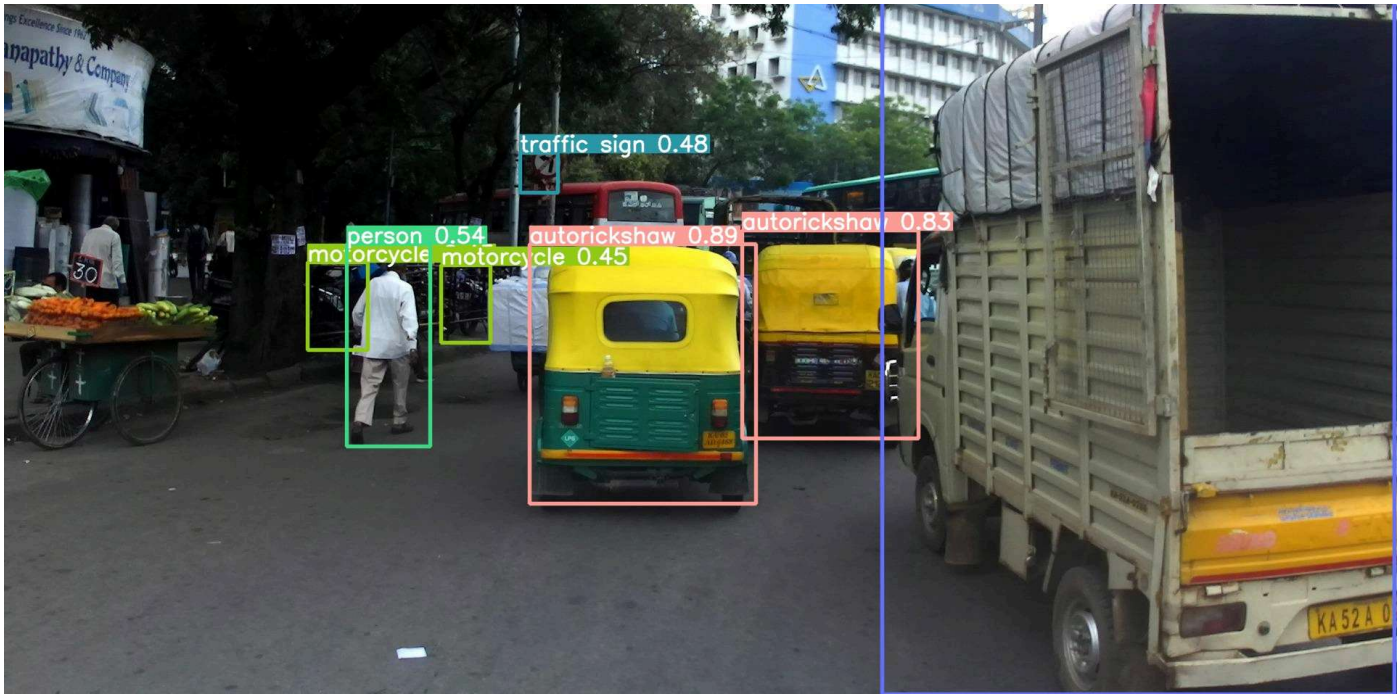
SHRI DURGA BIKE & SCOTER SERVICES
Specialist in : All Types of Two Wellers Service
autorickshaw 0.87
truck 0.76

traffic sign 0.77

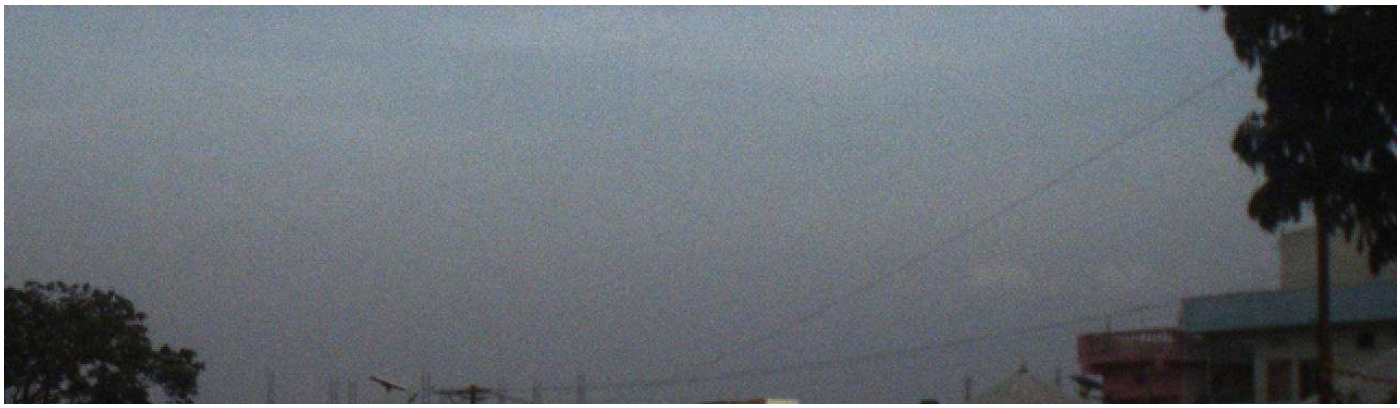
car 0.69

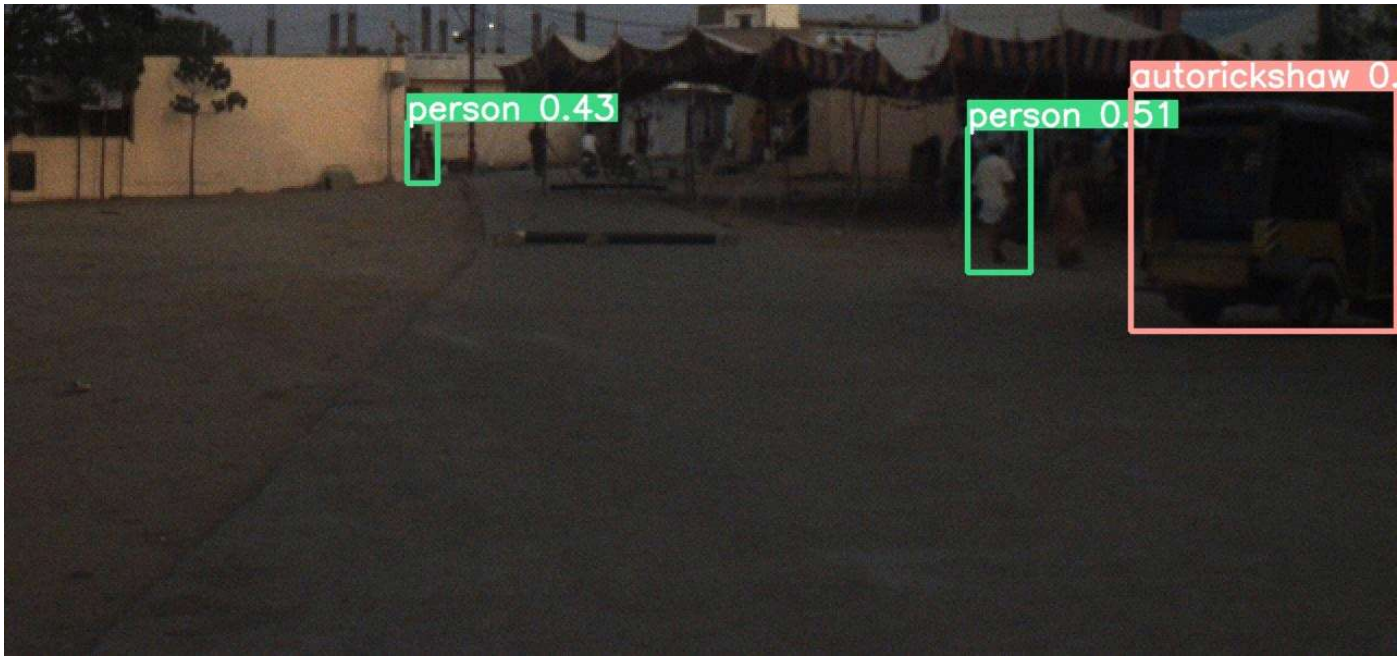

rider 0.57
motorcycle 0.72
car 0.87
car 0.84
car 0.91

truck 0.87


traffic sign 0.48
person 0.54
motorcycle
motorcycle 0.45
autorickshaw 0.89
autorickshaw 0.83

car 0.86
person 0.42
car 0.82




person 0.43
person 0.51
autorickshaw 0.

truck 0.73
bus 0.68
bus 0.43
bus 0.57
car 0.68
car 0.72
BMTC
KA 07 M 8098

Buffered data was truncated after reaching the output size limit.

```
%cp /content/yolov5/runs/detect/exp13/
```

```
cp: cannot stat '/content/yolov5/runs/detect/exp13/newroad.mp4': No such file or directory
```

```
!pip install -qr requirements.txt
import torch
from IPython.display import Image, clear_output  # to display images
print('Setup complete. Using torch %s %s' % (torch.__version__, torch.cuda.get_device_properties(0) if torch.cuda.is_available() else 'CPU'))

# Load necessary libraries
import cv2
import numpy as np

# Load YOLO model
model = torch.hub.load('ultralytics/yolov5', 'yolov5s', pretrained=True).to(device).eval()

# Define function to perform object detection on video frames
def detect_objects_in_video(video_path, output_path):
    cap = cv2.VideoCapture(video_path)
    frame_width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
    frame_height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
    fps = cap.get(cv2.CAP_PROP_FPS)

    fourcc = cv2.VideoWriter_fourcc(*'XVID')
    out = cv2.VideoWriter(output_path, fourcc, fps, (frame_width, frame_height))

    while cap.isOpened():
        ret, frame = cap.read()
        if not ret:
            break

        # Perform object detection
        results = model(frame)

        # Draw bounding boxes on the frame
        annotated_frame = results.render()

        # Convert the frame to BGR format (OpenCV uses BGR)
        annotated_frame = cv2.cvtColor(np.array(annotated_frame), cv2.COLOR_RGB2BGR)

        # Write the annotated frame to the output video
        out.write(annotated_frame)

    cap.release()
    out.release()
    cv2.destroyAllWindows()

# Run object detection on the input video
video_path = '/content/yolov5/data/input/WhatsApp Video 2024-04-18 at 7.51.47 PM.mp4'
output_path = '/content/yolov5/output'
detect_objects_in_video(video_path, output_path)
```

```
ERROR: Could not open requirements file: [Errno 2] No such file or directory: 'requirements.txt'
```

```
!pip install -qr requirements.txt
import torch
from IPython.display import Image, clear_output  # to display images
print('Setup complete. Using torch %s %s' % (torch.__version__, torch.cuda.get_device_properties(0) if torch.cuda.is_available() else 'CPU'))
```