

24.07- People Detection using VisDrone Dataset:

Gitlab repo: <https://gitlab.bewelltech.com.tr/Intern4/visdrone-object-detection>

Paddle Detection using VisDrone Dataset:

Page is on Chinese so its hard to read

<https://github.com/PaddlePaddle/PaddleDetection/tree/release/2.7>

YOLOE+

VisDrone Dataset:

Vision Meets Drones: A Challenge

<https://arxiv.org/abs/1804.07437>

```
# Classes
names:
0: pedestrian
1: people
2: bicycle
3: car
4: van
5: truck
6: tricycle
7: awning-tricycle
8: bus
9: motor
```

Dataset originally consist of followings:

Train: 6471 images

Val: 548 images

Test-dev: 1610 images

Test(challenge): 1580 images

Subset:

Train: 500 images

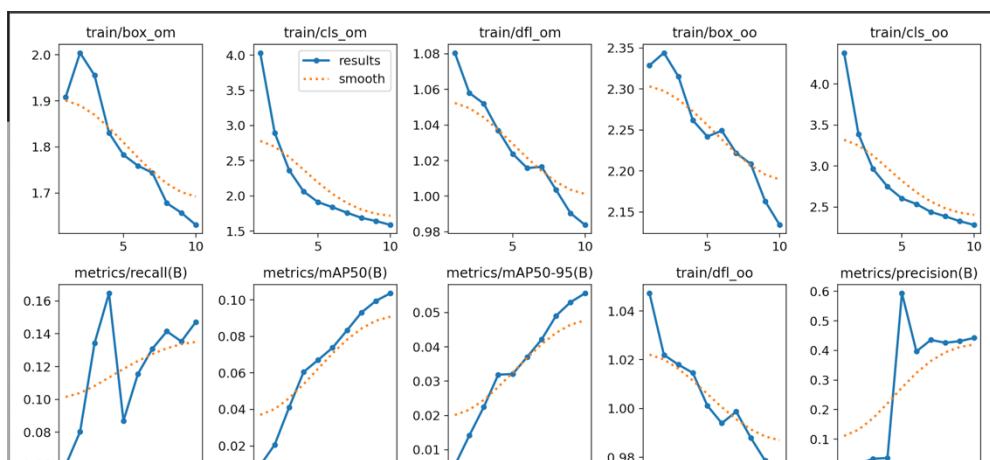
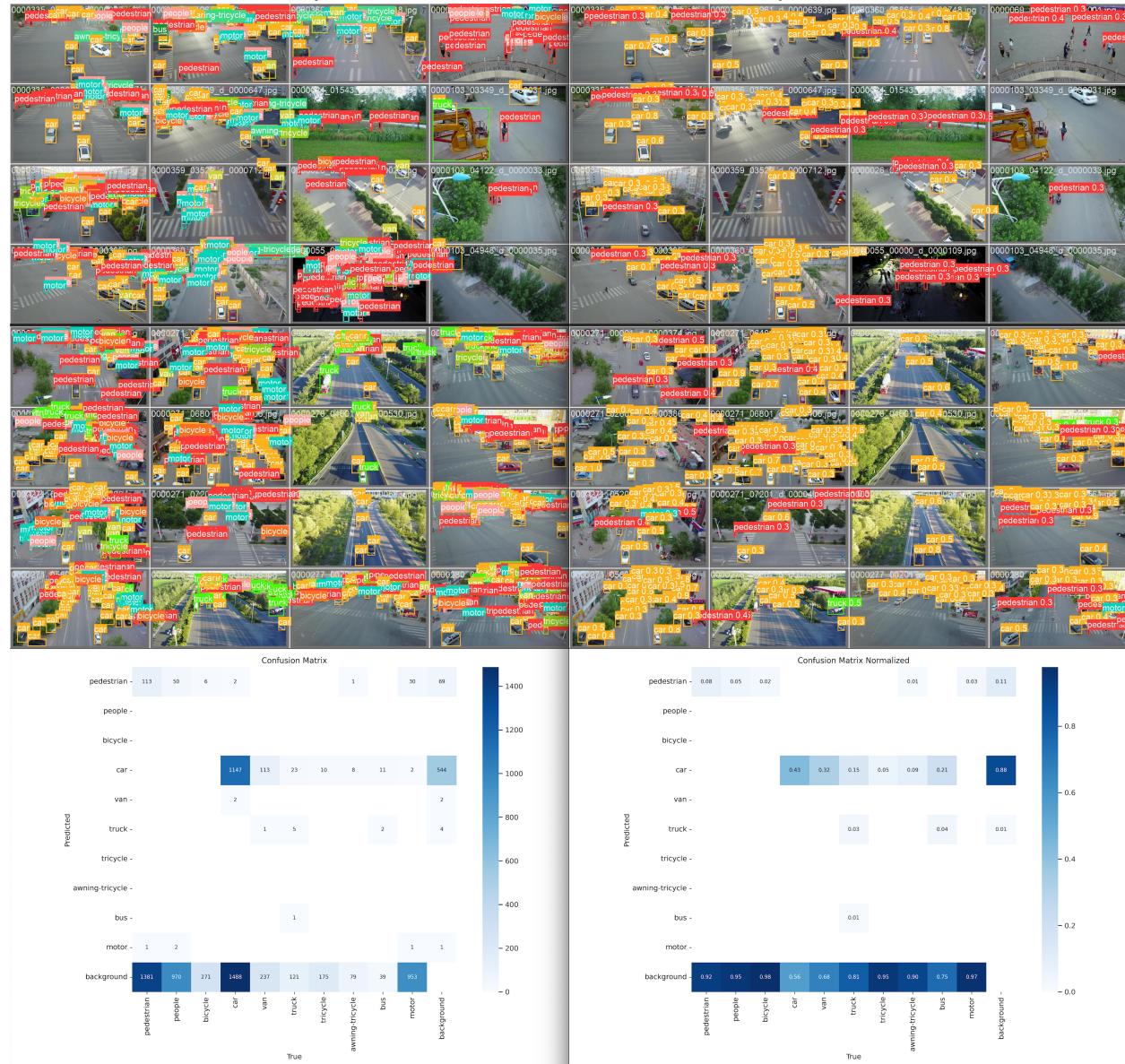
Val: 100 images

Test-dev: 100 images

- 1- Creating random subset with Python script. See “`data_reducer.py`” on [GitHub](#)
- 2- Data annotations converted to YOLO labels with Python Script. See “`annotation_handler.py`” on [GitHub](#)

Check [Colab notebook](#):

YOLOV10n multiclass object detection model results with 10 epochs batch size = 16



```

Validating runs/detect/train/weights/best.pt...
Ultralytics YOLOv8.1.34 🚀 Python-3.10.12 torch-2.3.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
YOLOv10n summary (fused): 285 layers, 2698316 parameters, 0 gradients, 8.2 GFLOPs
    Class     Images Instances      Box(P)        R      mAP50  mAP50-95): 100% 4/4 [00:12<00:00,  3.10s/it]
    all       100    7245    0.445    0.146    0.103    0.0556
pedestrian   100    1495    0.171    0.256    0.127    0.0508
  people    100    1022    0.245    0.0486   0.0926    0.0322
  bicycle   100     277      1        0        0        0
    car      100    2639    0.275    0.653    0.48     0.291
    van      100     351    0.108    0.162    0.0613   0.0375
    truck    100     150    0.147    0.0933   0.0655   0.0435
  tricycle   100     185      1        0        0.00235  0.00141
awning-tricycle 100      88      1        0        0        0
    bus      100      52    0.243    0.115    0.0824   0.0582
    motor    100    986     0.26    0.136    0.123    0.0407
Speed: 0.5ms preprocess, 5.1ms inference, 0.0ms loss, 0.3ms postprocess per image

```

Comments and inferences on my perspective:

10 epochs is not enough for training phase and also there are too many classes.

TODO 1: decrease class size(eliminate unnecessary classes and merge some classes into one)

TODO 2: increase train epochs

25.07- “Data Wrangling”

TODO 1: decrease class size (eliminate unnecessary classes and merge some classes into one)

- Combine class 0 (pedestrian) and 1 (people) to 0 (PERSON).
- Combine class 3 (car) 4 (van) 5(truck) 8(bus) and 9(motorcycle) to 1 (VEHICLE).
- Ignore class 2 (bicycle), 6 (tricycle), and 7 (awning-tricycle).

See updated python script “[annotation_handler.py](#)”