Data Collection Infotech (India) Pvt Ltd (DCIL)

Problem Statement:

We want you to detect and classify the sign object on image. The related **dataset** and **image details** have been uploaded in below link. Also, I have attached the detected sample images for your reference.
Dataset and Annotations spreadsheet:
https://drive.google.com/drive/folders/1E5-ioNPynBWQBHWtKsWP9ygLrDgFcuvp?usp=sharing

Solution : https://github.com/ChetanB1997/Road-Sign-identification

Dataset folder view:

```
|---- Dataset
|       |--- Do Not Enter
|       |--- Direction Arrow Auxiliary
|       |--- Advisory Speed Mph
|
|---- Annotation
|       |--- Annotation for 3Classes.csv
|
|---- Data
|       |--- train (xxx.jpg & xxx.txt)
|       |--- test (xyz.jpg & xyz.txt)
|
|---- data.yamls
```

Data preprocessing:

- $center\_x = \frac{\frac{x_{min}+x_{max}}{2}}{width\ of\ the\ image}$
- $center\_y = \frac{\frac{y_{min}+y_{max}}{2}}{height\ of\ the\ image}$
- $w = \frac{x_{max}-x_{min}}{width\ of\ the\ image}$
- $h = \frac{y_{max}-y_{min}}{height\ of\ the\ image}$

The dataset was need to be convert in standard format which is an image and its annotation in txt file. I have used glob for operating the files from folder. Also, annotations are converted into Data frame. Extracted hight and width of frames using cv2, Calculated the bounding box coordinates based on annotated data and derived formulas as shown below.

Finally applied label encoding to the categorical column and given 0,1,2 as IDs for 3 different categories. **Ref file-** **01_data_preprocessing.ipynb**

Data splitting= The total dataset contains 6000 images each of 2000 X3 categories.

Unique count of images was=5610

Training dataset shape= 4488 ; Test dataset shape= 1122

data.yaml is file format required to yolo model contain train and test dataset path information and details about categories.

Model and Training:

As for this object detection task I majorly focused on YOLOv5s model as it is single stage detection model it can process faster and gives bit a good accuracy. Here I selected 's- small' version which is pretrained on coco dataset. I have trained the model on custom dataset.

Clones the yolov5 repo from git hub

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

Run train.py from yolov5 folder:

```
!python train.py --data data.yaml --cfg yolov5s.yaml --batch-size 8 --name Model --epochs 20
```

The model training is been done on google colab

Batch size	Data size	No. of batches	Avg. Time(min)
4	4488	1122	50
8	4488	561	9
16	4488	281	Ram exceeds

Selected batch Size 8 as optimum based on system memory and performance.

Training results:

```
      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
      18/19      2.98G     0.0134   0.003674   0.000422         10        640: 100% 561/561 [09:08<00:00,  1.02it/s]
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100% 71/71 [00:41<00:00,  1.71it/s]
                   all       1122       1181      0.947      0.959      0.966      0.822

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
      19/19      2.98G    0.01269   0.003629  0.0003693         16        640: 100% 561/561 [09:06<00:00,  1.03it/s]
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100% 71/71 [00:45<00:00,  1.55it/s]
                   all       1122       1181      0.943      0.957      0.962      0.826

20 epochs completed in 3.544 hours.
Optimizer stripped from runs/train/Model7/weights/last.pt, 14.4MB
Optimizer stripped from runs/train/Model7/weights/best.pt, 14.4MB

Validating runs/train/Model7/weights/best.pt...
Fusing layers...
YOLOv5s summary: 157 layers, 7018216 parameters, 0 gradients, 15.8 GFLOPs
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100% 71/71 [00:43<00:00,  1.62it/s]
                   all       1122       1181      0.943      0.957      0.962      0.824
    ADVISORY SPEED MPH       1122        410      0.993      0.976      0.994      0.888
DIRECTIONAL ARROW AUXILIARY      1122        367      0.865      0.929       0.91      0.704
         DO NOT ENTER       1122        404      0.972      0.965      0.982       0.88
Results saved to runs/train/Model7
```

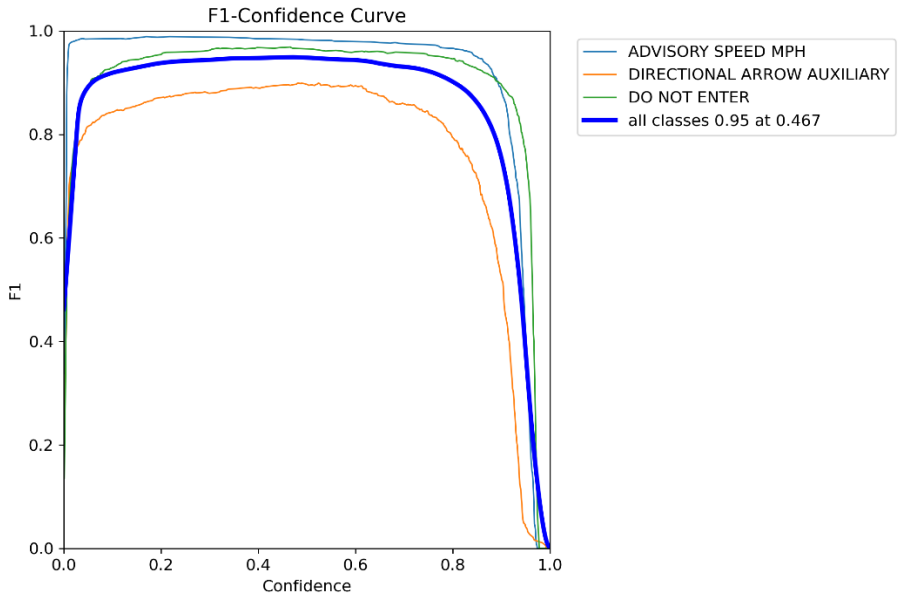
Based on Training

- Validation Metrics:

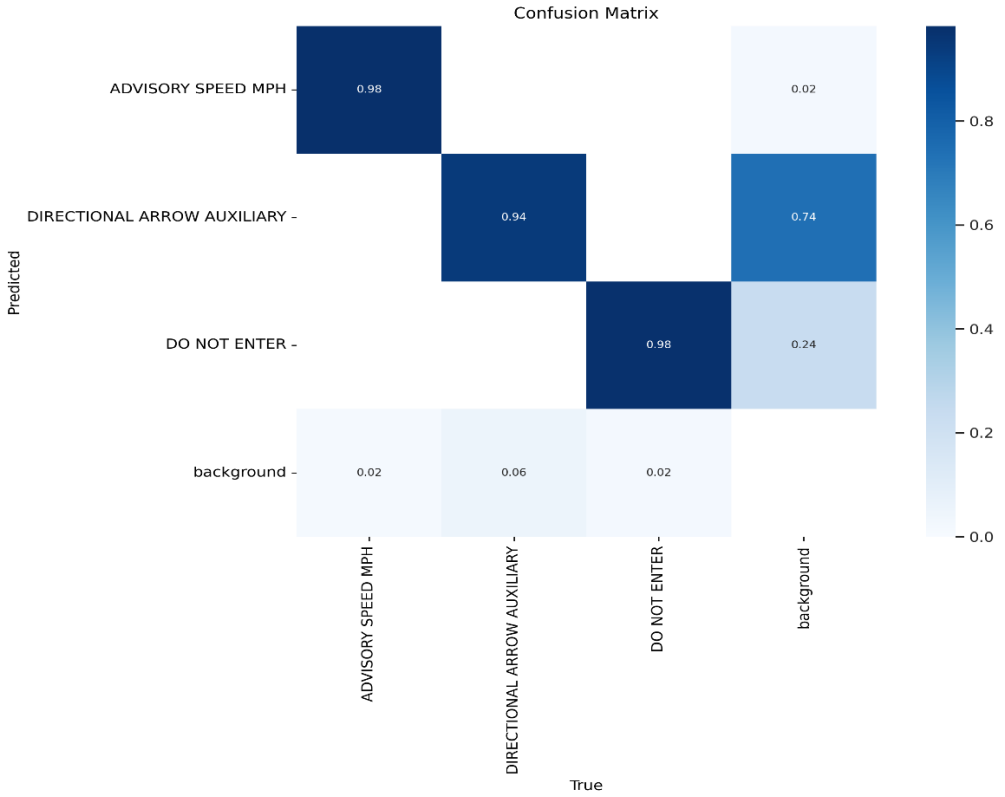
1. Overall mAP50 (Mean Average precision over 50% of IOU) = 96.2%
2. mAP50-95 (Mean Average precision over range 50- 95 % of IOU) = 82.4 %

category wise validation metrics:

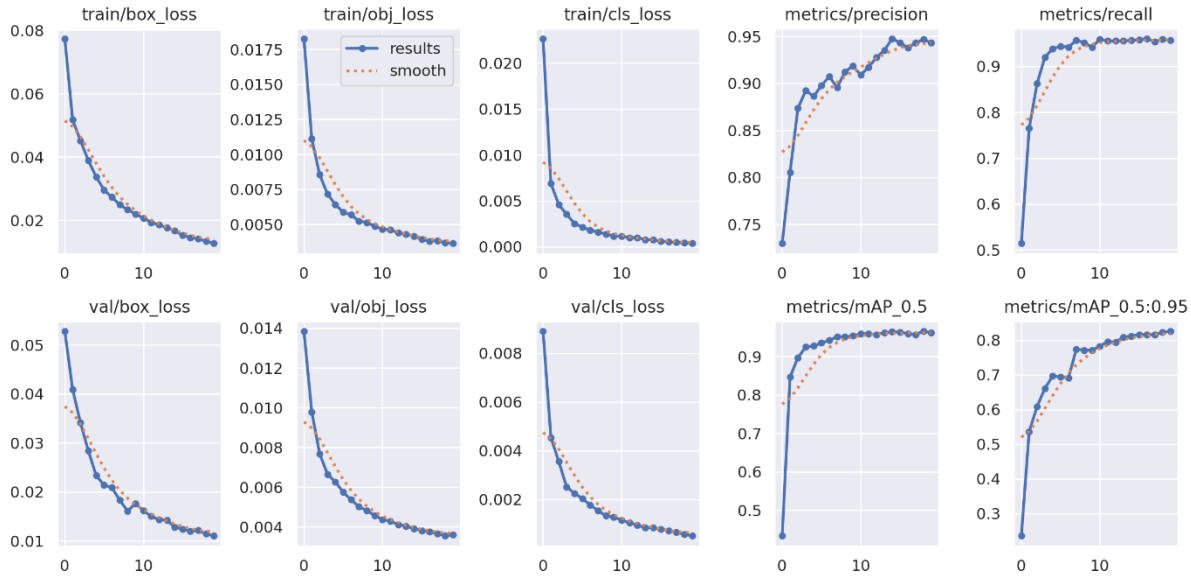
	Advisory Speed MPH	Directional Arrow Auxiliary	Do Not Enter
Precision	0.993	0.865	0.972
Recall	0.976	0.929	0.965
mAP50	0.994	0.91	0.982
mAP50-90	0.888	0.704	0.88



Confusion Matrix:



Training Plot:



As from graph we can visualize that the loss is still decreasing so its indicating that model is not converged yet there is still scope available for us to train the model further to get better results.

Prediction on test data:

Run yolo_prediction.py file (Set path for checkpoint and input image/ video)