

▮ Traffic Sign Detection using YOLOv8

▮ Dataset Overview

The dataset used for training and evaluation was downloaded from Kaggle:

- **Source:** [pkdarabi/cardetection](#)
 - Contains annotated car and traffic sign images.
 - Data organized into:
 - /train/images
 - /train/labels
 - /val/images
 - /val/labels
 - Data is referenced using a `data.yaml` configuration file.
-

▮ Model Architecture

The model used is:

- **YOLOv8n:** The nano version of the YOLOv8 object detection architecture from Ultralytics.
- Lightweight and optimized for speed.

Code Snippets:

```
from ultralytics import YOLO
Final_model = YOLO('yolov8n.pt')
▮ Training Configuration
Epochs: 25

Batch Size: Automatically selected (batch = -1)

Image Size: 640x640

Optimizer: Auto

Framework: Ultralytics YOLOv8 with PyTorch backend

Hardware: GPU (when available)
```

▮ Training Configuration

- Epochs: 25
 - Batch Size: Automatically selected (batch = -1)
 - Image Size: 640x640
 - Optimizer: Auto
 - Framework: Ultralytics YOLOv8 with PyTorch backend
 - Hardware: GPU (when available)
-

Training Command :

```
Result_Final_model = Final_model.train(  
    data="/kaggle/input/cardetection/car/data.yaml",  
    epochs=25,  
    batch=-1,  
    optimizer='auto'  
)
```

▮ Training Curves and Metrics

Metrics Visualized:

- train/box_loss, train/cls_loss, train/dfl_loss
- val/box_loss, val/cls_loss

```
sns.lineplot(x='epoch', y='train/box_loss', data=Result_Final_model)  
sns.lineplot(x='epoch', y='val/cls_loss', data=Result_Final_model)
```

Loss graphs were plotted using Seaborn and Matplotlib.

▮ Detection Results (Visuals)

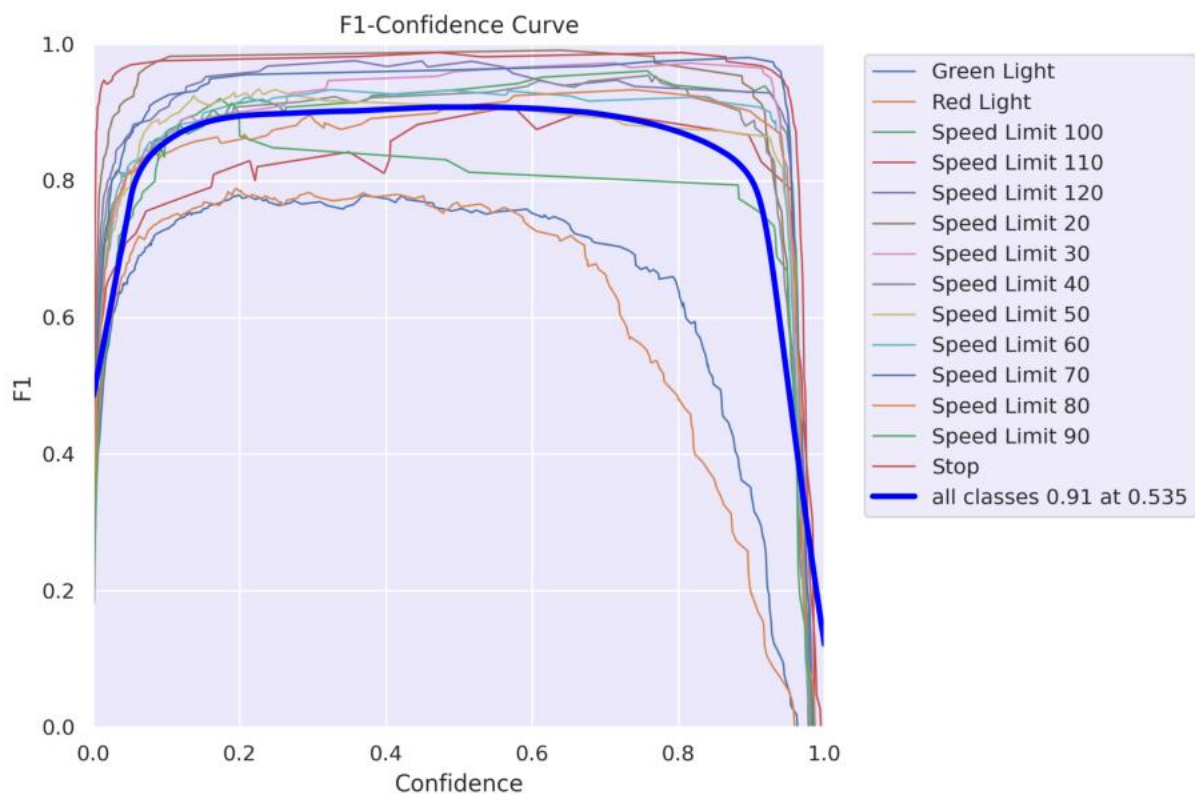
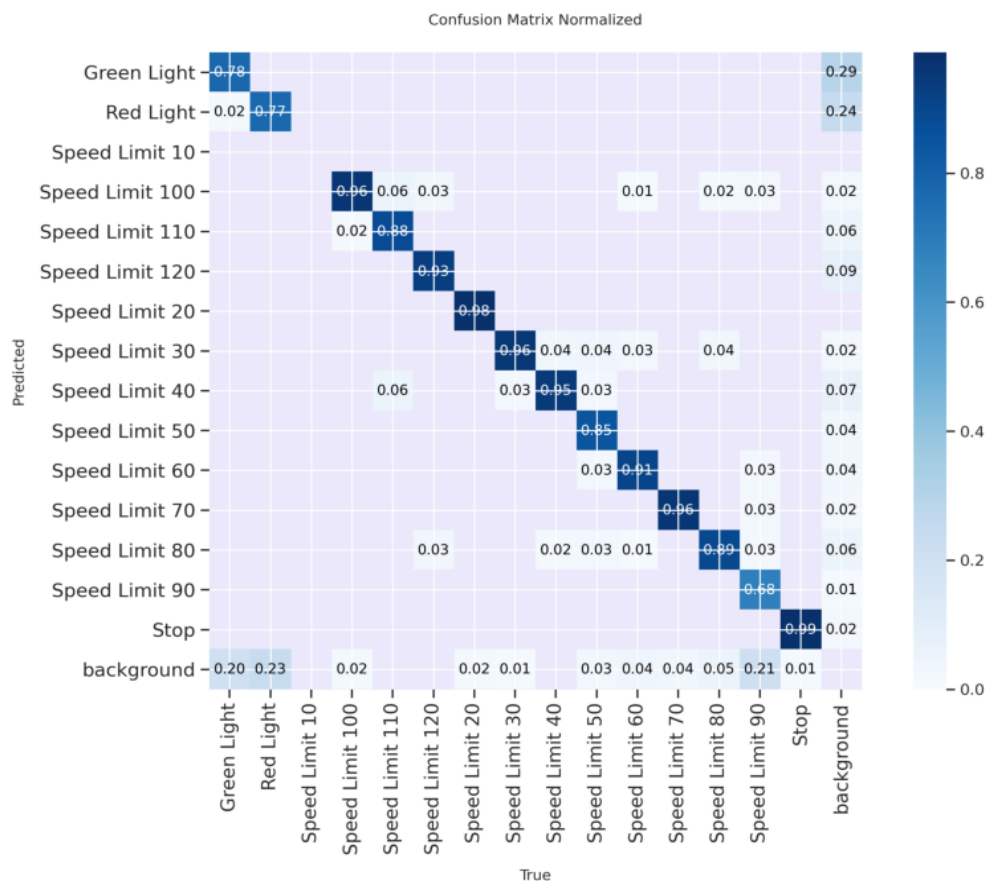
- Images from validation set used for testing:

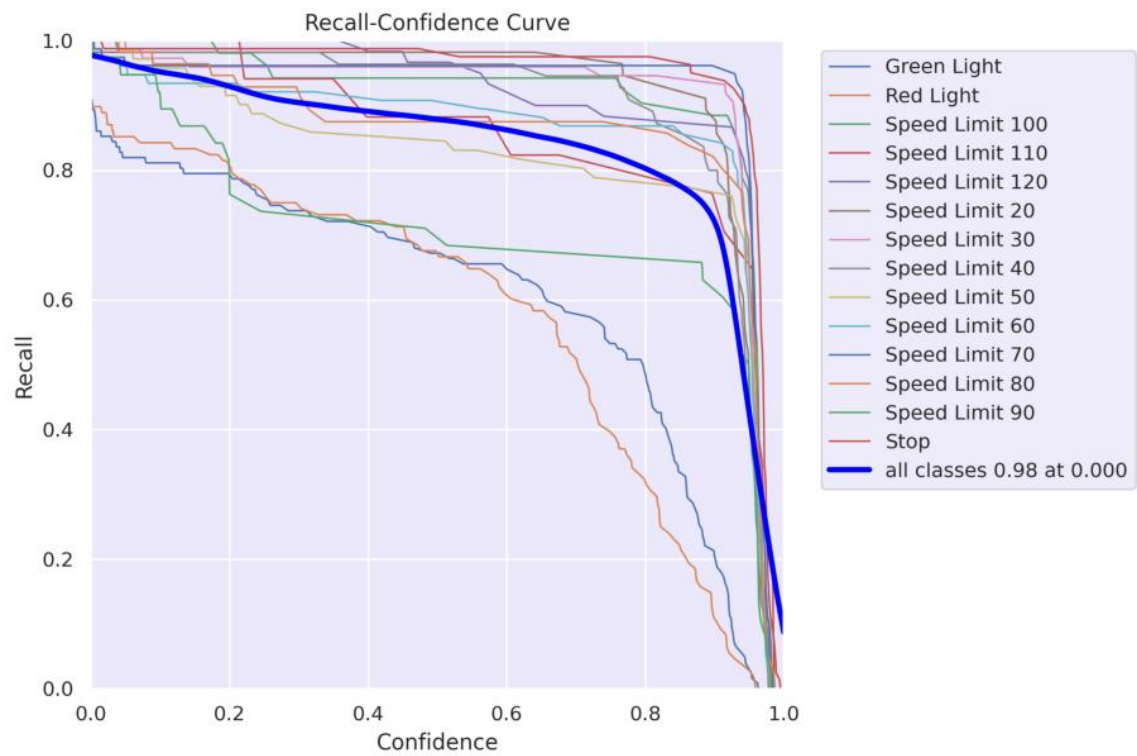
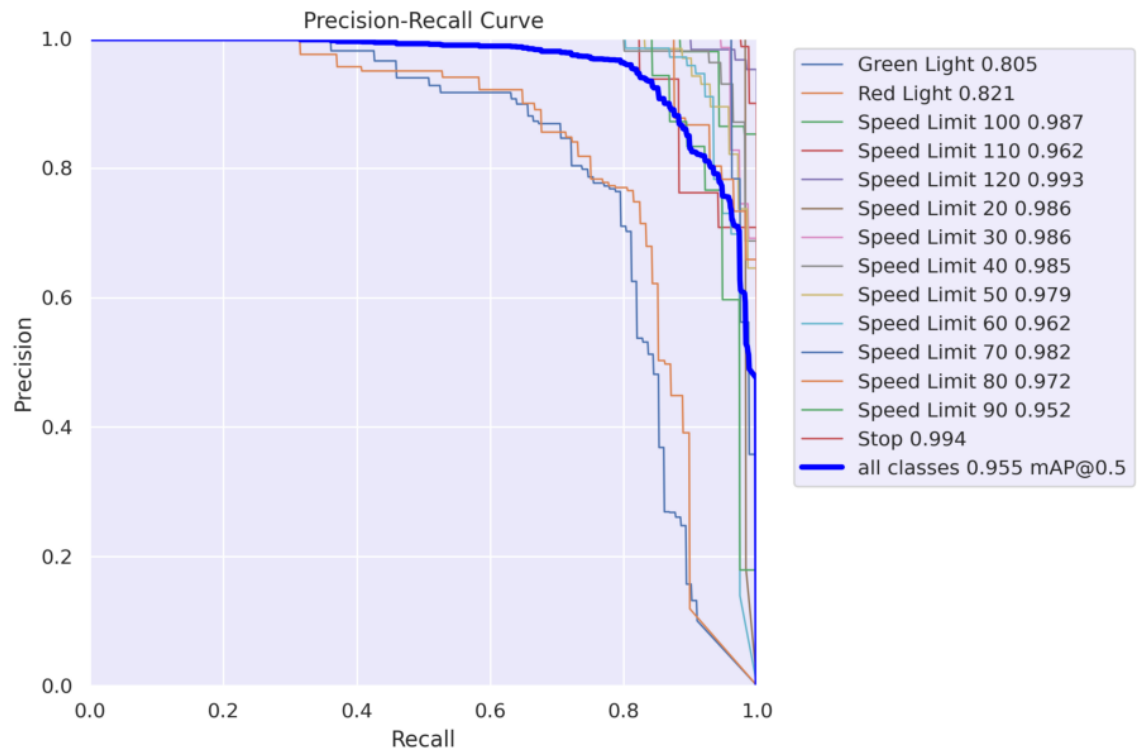
```
image = "/kaggle/input/cardetection/car/train/images/example.jpg"  
result_predict = model.predict(source=image, imgsz=(640))  
plot = result_predict[0].plot()
```

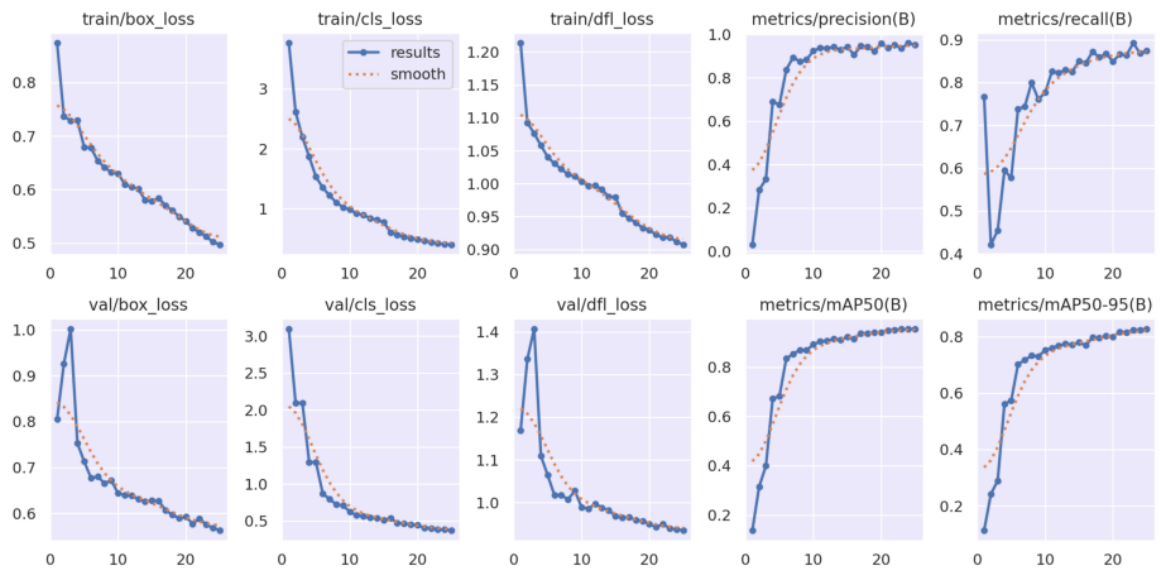
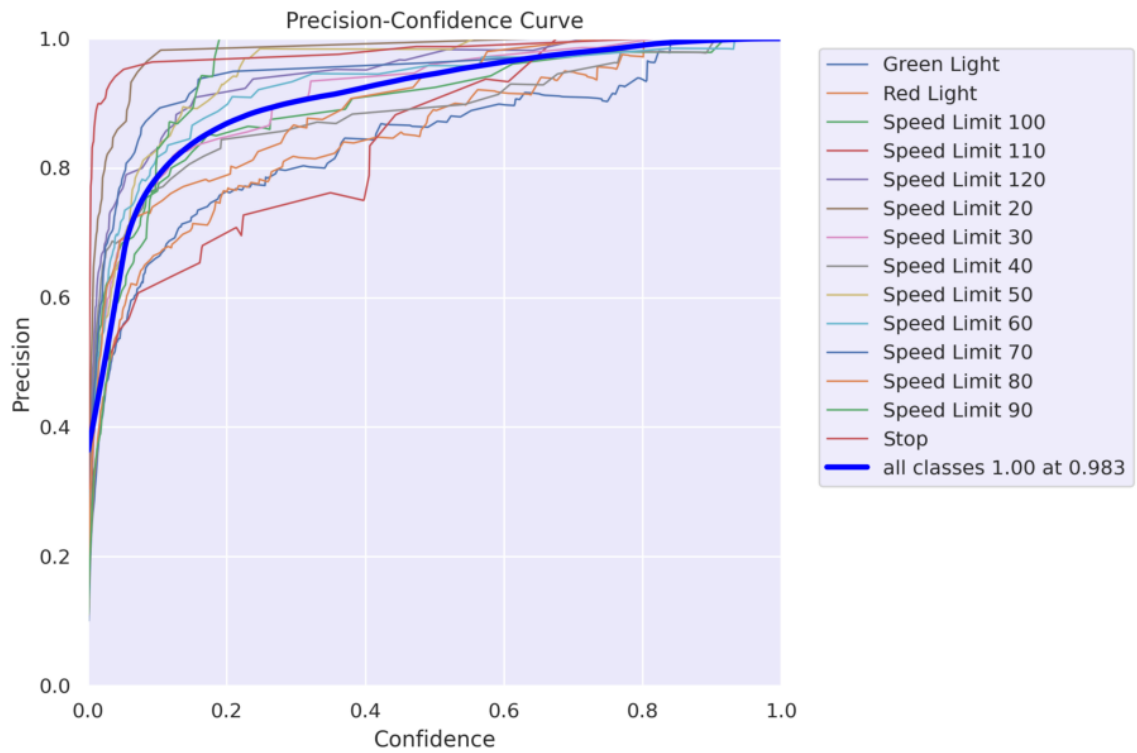
- Detected objects are overlaid with bounding boxes and class labels.

▮ Hyperparameter Tuning Summary

- Model Variant: yolov8n.pt (smallest, fastest)
 - Image Size: Default 640, can be tuned to 416 for speed
 - Cache: Enabled for faster training (cache=True)
 - AMP: Mixed-precision training supported (amp=True)
 - No additional grid search or learning rate tuning was performed. Further tuning can include:
 - Adjusting image size
 - Manual batch size control
 - Epoch adjustment
 - Using a learning rate scheduler
-







Training Metrics and Loss

