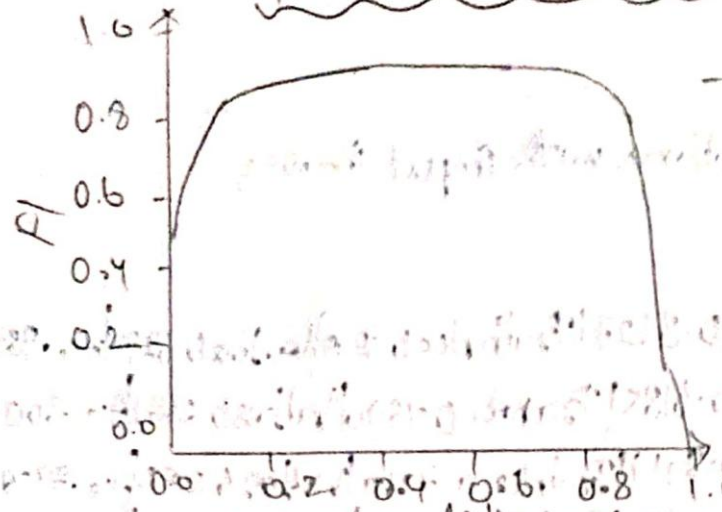
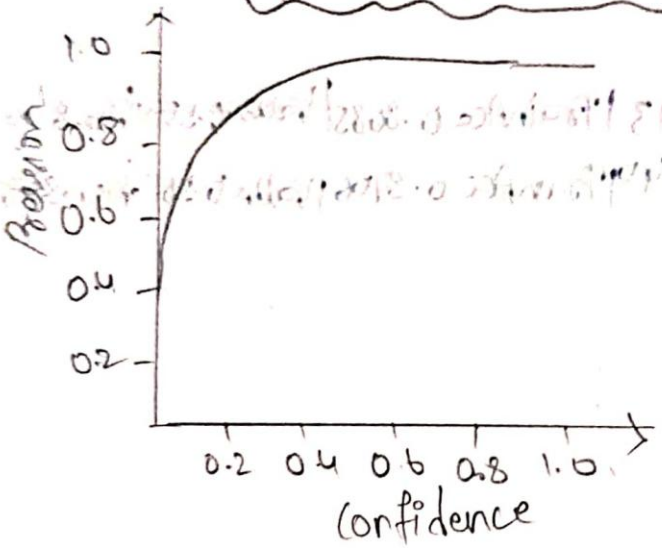
F1-Confidence Curve



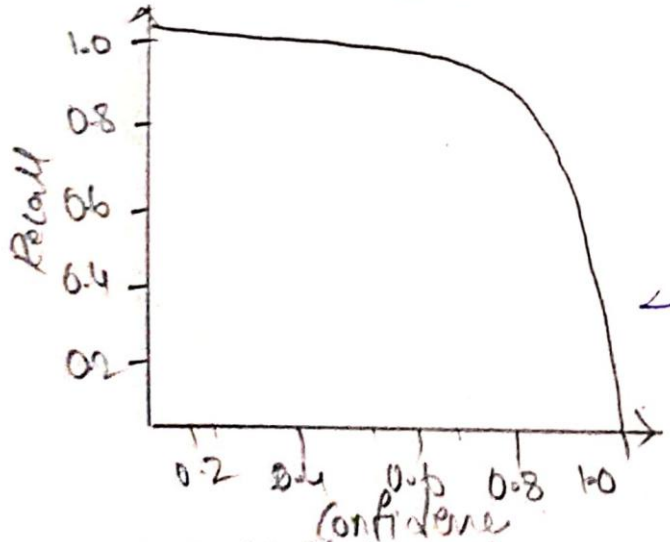
- all classes 0.92 at 0.339

Precision Confidence Curve



- all classes 1.00 at 0.906

Recall Confidence Curve



- all classes 0.98 at 0.000

Exp-15 IMPLEMENT A YOLO MODEL TO DETECT OBJECT

AIM

To develop a robust traffic sign detection system using YOLOv8 that can accurately identify and localize multiple traffic signs in images/videos.

OBJECTIVES:

↳ Data Acquisition and Preparation
↳ Model Selection and setup
↳ Training the YOLOv8 model
↳ Evaluation Validation.
↳ Prediction and Inference
↳ Model Export and Deployment.

PSEUDOCODE:

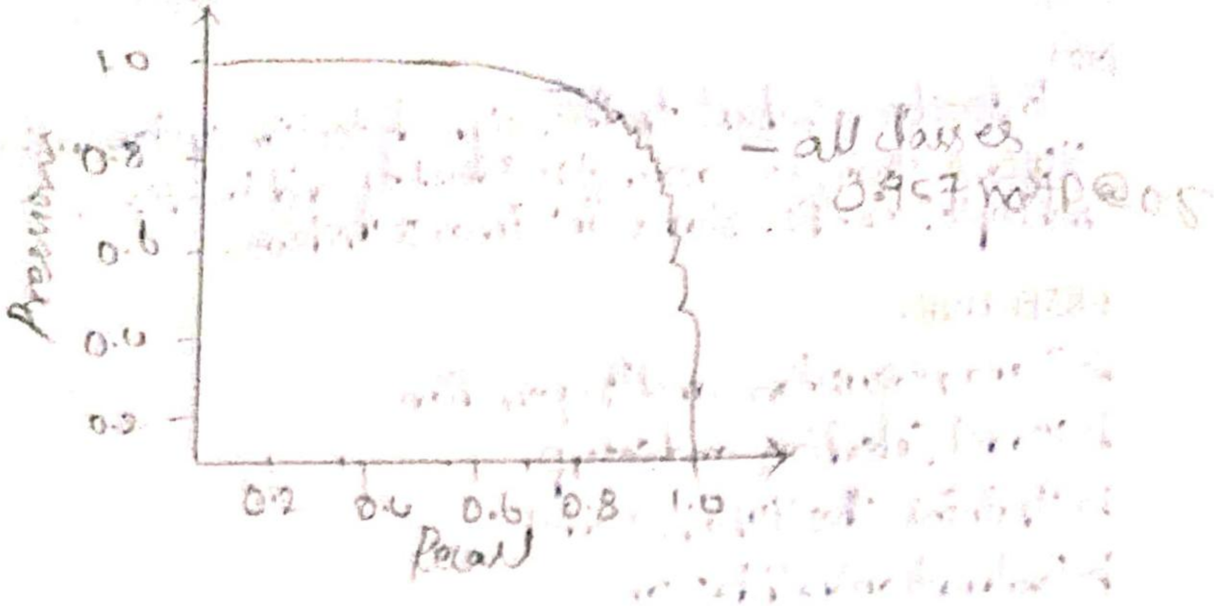
↳ Import necessary libraries
↳ Download and prepare the dataset.
↳ Initialize YOLOv5 model
↳ Train YOLOv5 model
↳ Evaluate model performance
↳ Perform inference on test images
    for each image in testset:
↳ Perform inference on videos.
↳ Export final model.
↳ Deployment (optional)

OBSERVATIONS:

epoch

Epoch	box_loss	cls_loss	dfl_loss	Instances	Size
1/30	0.9063	3.816	1.231		640
2/30	0.7314	2.68	106		640
3/30	0.7322	1.914	121		
4/30	0.7114	1.914	121		640
5/30	0.6686	1.391	114		640

Precision - Recall Curve



:

29/30	0.4945	0.3838	0.9014	65	640
30/30	0.4902	0.3806	0.909	65	640

Precission :0.9492598112977769 3

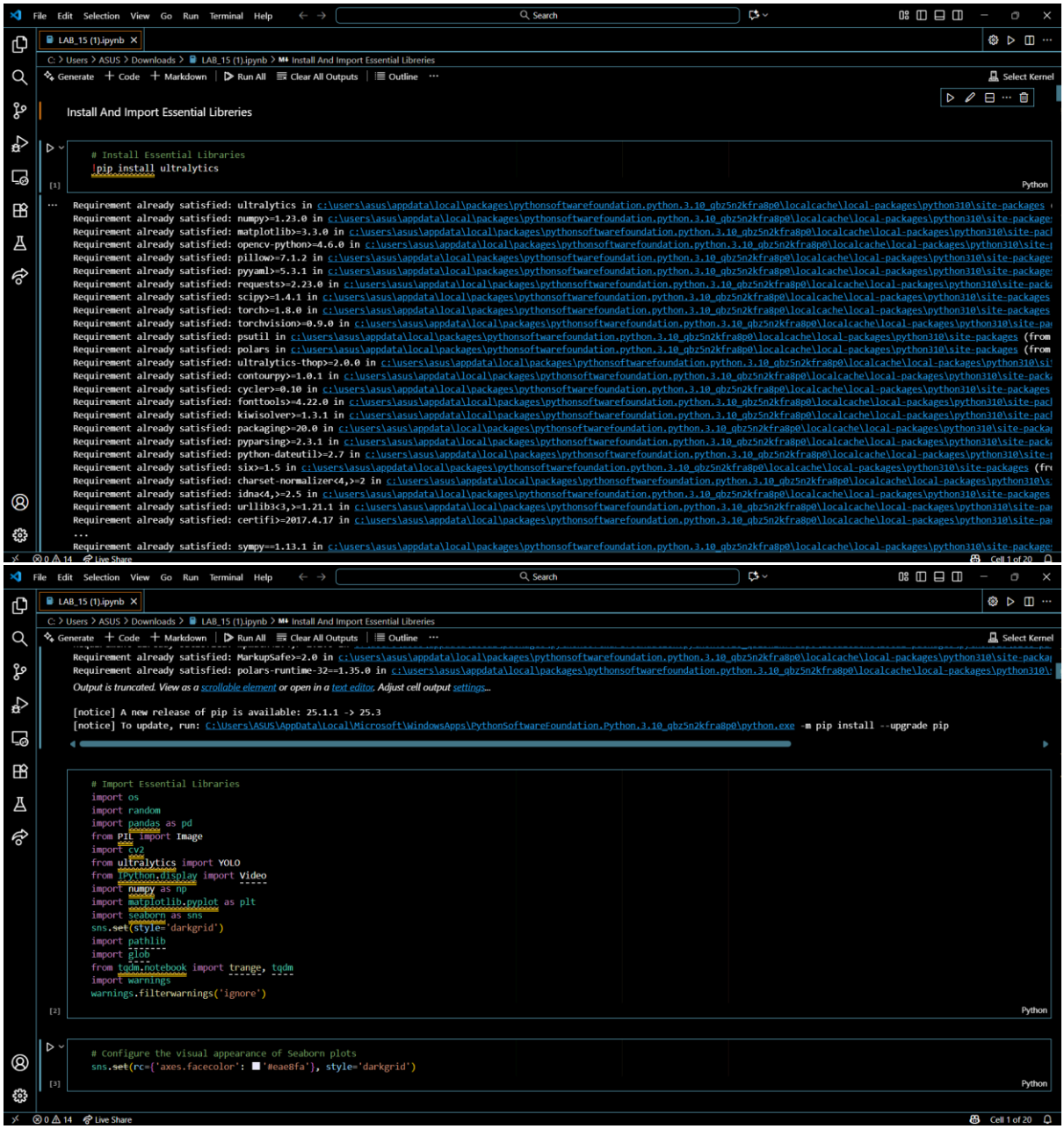
metrics/recall : 0.90602816301361 68

metrics/mAP50(B) 0.9571950392546826

metrics/mAP50-95(B): 0.8296650258385937.

Result

Successfully Implemented a YOLOv8 model to detect object.


File Edit Selection View Go Run Terminal Help
Search
LAB_15 (1).ipynb
C: > Users > ASUS > Downloads > LAB_15 (1).ipynb > Install And Import Essential Libreries
Generate + Code + Markdown | Run All | Clear All Outputs | Outline ...
Select Kernel
Install And Import Essential Libreries
Install Essential Libraries
!pip install ultralytics
[1]
Python
Requirement already satisfied: ultralytics in c:\users\asus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages
Requirement already satisfied: numpy>=1.23.0 in c:\users\asus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages
Requirement already satisfied: matplotlib>=3.3.0 in c:\users\asus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages
Requirement already satisfied: opencv-python>=4.6.0 in c:\users\asus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages
Requirement already satisfied: pillow>=7.1.2 in c:\users\asus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages
Requirement already satisfied: pyyaml>=5.3.1 in c:\users\asus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages
Requirement already satisfied: requests>=2.23.0 in c:\users\asus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages
Requirement already satisfied: scipy>=1.4.1 in c:\users\asus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages
Requirement already satisfied: torch>=1.8.0 in c:\users\asus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages
Requirement already satisfied: torchvision>=0.9.0 in c:\users\asus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages
Requirement already satisfied: psutil in c:\users\asus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages (from
Requirement already satisfied: polars in c:\users\asus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages (from
Requirement already satisfied: ultralytics-thop>=2.0.0 in c:\users\asus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages
Requirement already satisfied: contourpy>=1.0.1 in c:\users\asus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages
Requirement already satisfied: cycler>=0.10 in c:\users\asus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages
Requirement already satisfied: fonttools>=4.22.0 in c:\users\asus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\asus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages
Requirement already satisfied: packaging>=20.0 in c:\users\asus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\asus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages
Requirement already satisfied: python-dateutil>=2.7 in c:\users\asus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages
Requirement already satisfied: six>=1.5 in c:\users\asus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages (fr
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\asus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\s
Requirement already satisfied: idna<4,>=2.5 in c:\users\asus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\asus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages
Requirement already satisfied: certifi>=2017.4.17 in c:\users\asus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages
...
Requirement already satisfied: sympy==1.13.1 in c:\users\asus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages
0 14 Live Share
Cell 1 of 20

File Edit Selection View Go Run Terminal Help
Search
LAB_15 (1).ipynb
C: > Users > ASUS > Downloads > LAB_15 (1).ipynb > Install And Import Essential Libreries
Generate + Code + Markdown | Run All | Clear All Outputs | Outline ...
Select Kernel
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\asus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages
Requirement already satisfied: polars-runtime-32==1.35.0 in c:\users\asus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\s
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
[notice] A new release of pip is available: 25.1.1 -> 25.3
[notice] To update, run: C:\Users\ASUS\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation.Python.3.10_qbz5n2kfra8p0\python.exe -m pip install --upgrade pip
# Import Essential Libraries
import os
import random
import pandas as pd
from PIL import Image
import cv2
from ultralytics import YOLO
from IPython.display import Video
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(style='darkgrid')
import pathlib
import glob
from tqdm.notebook import trange, tqdm
import warnings
warnings.filterwarnings('ignore')
[2]
Python
# Configure the visual appearance of Seaborn plots
sns.set(rc={'axes.facecolor': '#eae8fa'}, style='darkgrid')
[3]
Python
0 14 Live Share
Cell 1 of 20

```
# Configure the visual appearance of Seaborn plots
sns.set(rc={'axes.facecolor': '#eae8fa'}, style='darkgrid')
```

[3] Python

Dataset

```
import kagglehub

# Download latest version
path = kagglehub.dataset_download("pkdarabi/cardetection")

print("Path to dataset files:", path)
```

[4] Python

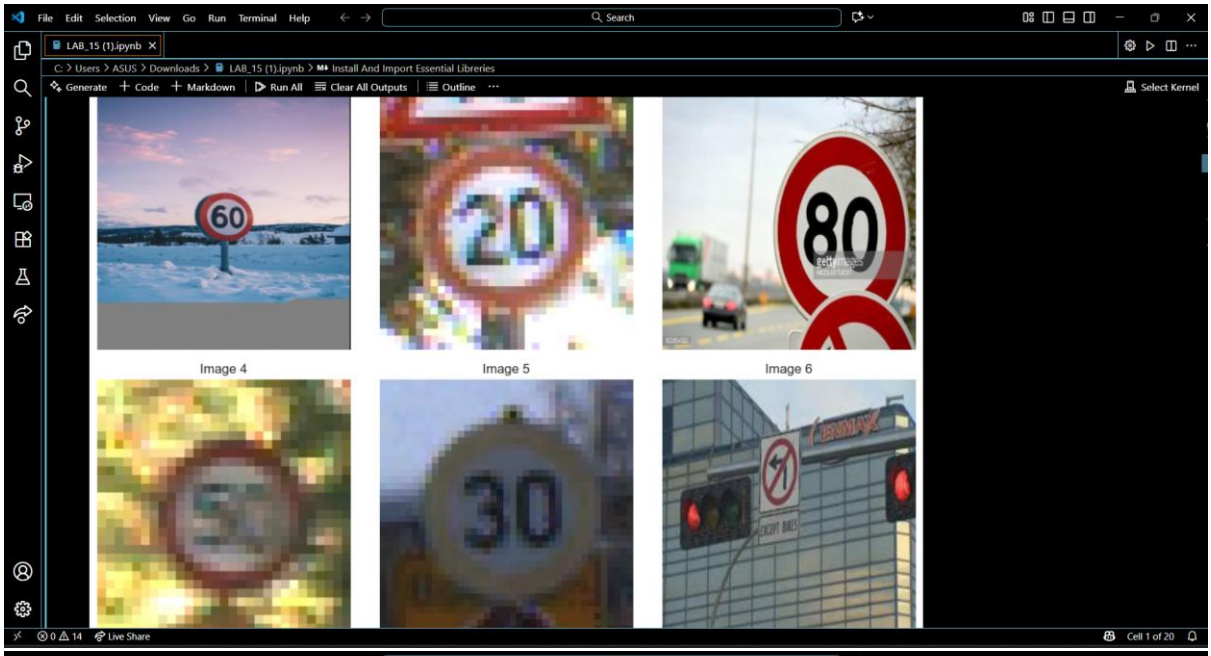
```
Path to dataset files: C:\Users\ASUS\.cache\kagglehub\datasets\pkdarabi\cardetection\versions\5
```

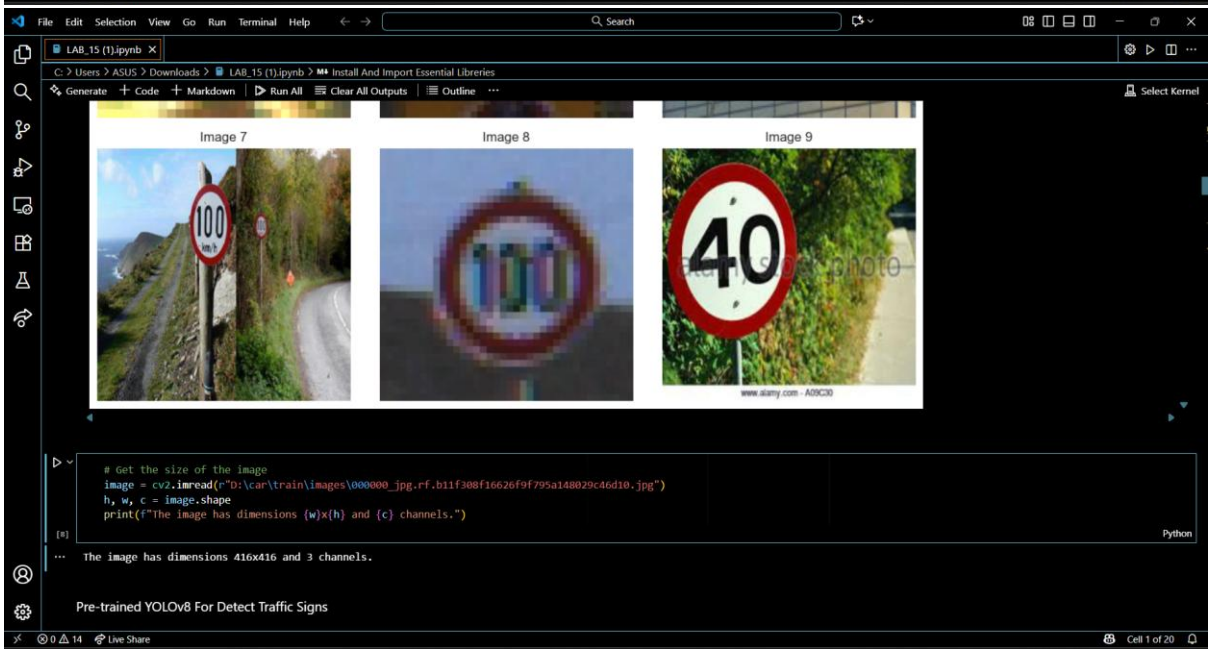
```
Image_dir = r'D:\car\train\images'

num_samples = 9
image_files = os.listdir(Image_dir)

# Randomly select num_samples images
rand_images = random.sample(image_files, num_samples)

fig, axes = plt.subplots(3, 3, figsize=(11, 11))

for i in range(num_samples):
    image = rand_images[i]
    ax = axes[i // 3, i % 3]
    ax.imshow(plt.imread(os.path.join(Image_dir, image)))
```

[6] Python


```
Image_dir = r'D:\car\train\images'

num_samples = 9
image_files = os.listdir(Image_dir)

# Randomly select num_samples images
rand_images = random.sample(image_files, num_samples)

fig, axes = plt.subplots(3, 3, figsize=(11, 11))

for i in range(num_samples):
    image = rand_images[i]
    ax = axes[i // 3, i % 3]
    ax.imshow(plt.imread(os.path.join(Image_dir, image)))
    ax.set_title(f'Image {i+1}')
    ax.axis('off')

plt.tight_layout()
plt.show()
```


[6] Python


File Edit Selection View Go Run Terminal Help
Search
LAB_15 (1).ipynb
C: > Users > ASUS > Downloads > LAB_15 (1).ipynb > Install And Import Essential Libreries
Generate + Code + Markdown Run All Clear All Outputs Outline ...
Select Kernel
Image 4
Image 5
Image 6
Cell 1 of 20


File Edit Selection View Go Run Terminal Help
Search
LAB_15 (1).ipynb
C: > Users > ASUS > Downloads > LAB_15 (1).ipynb > Install And Import Essential Libreries
Generate + Code + Markdown Run All Clear All Outputs Outline ...
Select Kernel
Image 7
Image 8
Image 9
# Get the size of the image
image = cv2.imread(r"D:\car\train\images\000000_jpg.rf.b11f308f16626f9f795a148029c46d10.jpg")
h, w, c = image.shape
print(f"The image has dimensions {w}x{h} and {c} channels.")
Python
The image has dimensions 416x416 and 3 channels.
Pre-trained YOLOv8 For Detect Traffic Signs
Cell 1 of 20

File Edit Selection View Go Run Terminal Help

Search

LAB_15 (1).ipynb

C: > Users > ASUS > Downloads > LAB_15 (1).ipynb > Install And Import Essential Libreries

Generate + Code + Markdown | Run All | Clear All Outputs | Outline ... | Select Kernel

```python
# Get the size of the image
image = cv2.imread(r"D:\car\train\images\000000_jpg.rf.b11f308f16626f9f795a148029c46d10.jpg")
h, w, c = image.shape
print(f"The image has dimensions {w}x{h} and {c} channels.")
```

[8] Python

```
The image has dimensions 416x416 and 3 channels.
```

Pre-trained YOLOv8 For Detect Traffic Signs

```python
# Use a pretrained YOLOv8n model
model = YOLO("yolov8n.pt")

# Use the model to detect object
image = r"D:\car\train\images\000000_jpg.rf.b11f308f16626f9f795a148029c46d10.jpg"
result_predict = model.predict(source = image, imgsz=(640))

# show results
plot = result_predict[0].plot()
plot = cv2.cvtColor(plot, cv2.COLOR_BGR2RGB)
display(Image.fromarray(plot))
```

[9] Python

```
Downloading https://github.com/ultralytics/assets/releases/download/v8.3.0/yolov8n.pt to 'yolov8n.pt': 100% ━━━━━━━━━━━━ 6.2MB 56.2KB/s 1:54 1:54<0.2s2:06

image 1/1 D:\car\train\images\000000_jpg.rf.b11f308f16626f9f795a148029c46d10.jpg: 640x640 (no detections), 18.7ms
Speed: 7.8ms preprocess, 18.7ms inference, 2.5ms postprocess per image at shape (1, 3, 640, 640)
```

0 ⚠ 14 Live Share Cell 1 of 20

File Edit Selection View Go Run Terminal Help

Search

LAB_15 (1).ipynb

C: > Users > ASUS > Downloads > LAB_15 (1).ipynb > Install And Import Essential Libreries

Generate + Code + Markdown | Run All | Clear All Outputs | Outline ... | Select Kernel

[9] Python

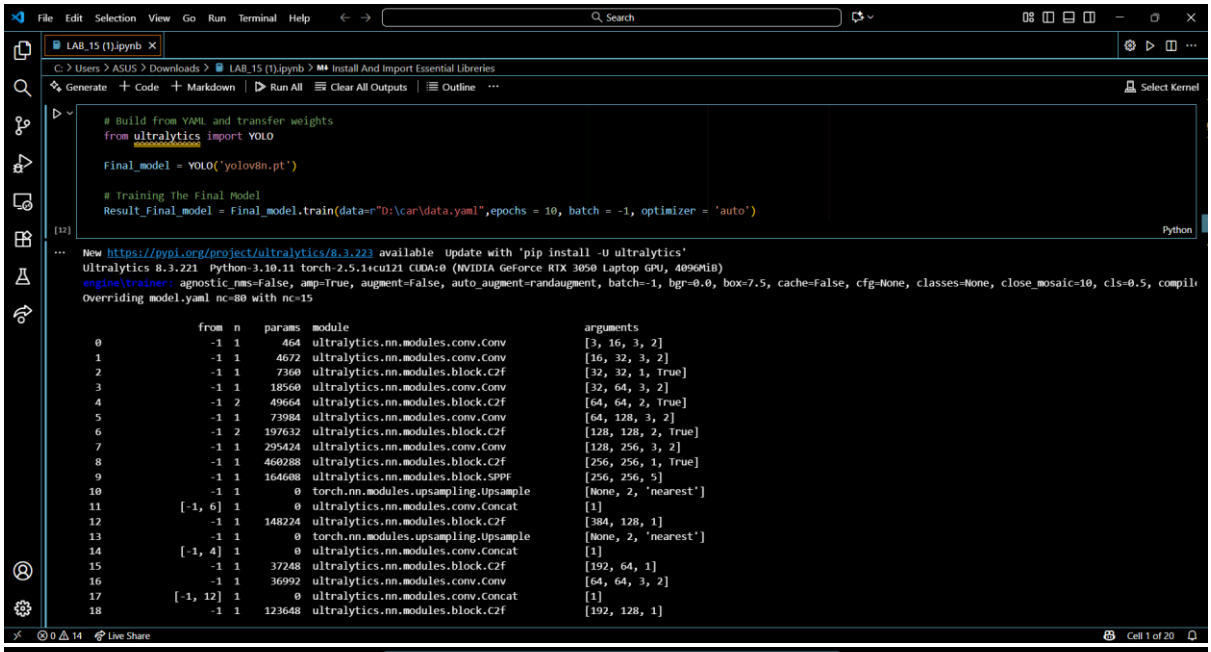
Downloading https://github.com/ultralytics/assets/releases/download/v8.3.0/yolov8n.pt to 'yolov8n.pt': 100% ━━━━━━━━━━━━ 6.2MB 56.2KB/s 1:54 1:54<0.2s2:06

image 1/1 D:\car\train\images\000000_jpg.rf.b11f308f16626f9f795a148029c46d10.jpg: 640x640 (no detections), 18.7ms
Speed: 7.8ms preprocess, 18.7ms inference, 2.5ms postprocess per image at shape (1, 3, 640, 640)


```python
# Build from YAML and transfer weights
from ultralytics import YOLO
```

[12]

0 ⚠ 14 Live Share Cell 1 of 20


File Edit Selection View Go Run Terminal Help
Search
LAB_15 (1).ipynb
C: > Users > ASUS > Downloads > LAB_15 (1).ipynb > Install And Import Essential Libreries
Generate + Code + Markdown | Run All | Clear All Outputs | Outline ...
Select Kernel
# Build from YAML and transfer weights
from ultralytics import YOLO
Final_model = YOLO('yolov8n.pt')
# Training The Final Model
Result_Final_model = Final_model.train(data=r"D:\car\data.yaml",epochs = 10, batch = -1, optimizer = 'auto')
[12]
Python
New https://pypi.org/project/ultralytics/8.3.223 available Update with 'pip install -U ultralytics'
Ultralytics 8.3.221 Python-3.10.11 torch-2.5.1+cu121 CUDA:0 (NVIDIA GeForce RTX 3050 Laptop GPU, 4096MiB)
engine\trainer: agnostic_nms=False, amp=True, augment=False, auto_augment=randaugment, batch=-1, bgr=0.0, box=7.5, cache=False, cfg=None, classes=None, close_mosaic=10, cls=0.5, compil
Overriding model.yaml nc=80 with nc=15
from n params module arguments
0 -1 1 464 ultralytics.nn.modules.conv.Conv [3, 16, 3, 2]
1 -1 1 4672 ultralytics.nn.modules.conv.Conv [16, 32, 3, 2]
2 -1 1 7360 ultralytics.nn.modules.block.C2f [32, 32, 1, True]
3 -1 1 18560 ultralytics.nn.modules.conv.Conv [32, 64, 3, 2]
4 -1 2 49664 ultralytics.nn.modules.block.C2f [64, 64, 2, True]
5 -1 1 73984 ultralytics.nn.modules.conv.Conv [64, 128, 3, 2]
6 -1 2 197632 ultralytics.nn.modules.block.C2f [128, 128, 2, True]
7 -1 1 295424 ultralytics.nn.modules.conv.Conv [128, 256, 3, 2]
8 -1 1 460288 ultralytics.nn.modules.block.C2f [256, 256, 1, True]
9 -1 1 164608 ultralytics.nn.modules.block.SPPF [256, 256, 5]
10 -1 1 0 torch.nn.modules.upsampling.Upsample [None, 2, 'nearest']
11 [-1, 6] 1 0 ultralytics.nn.modules.conv.Concat [1]
12 -1 1 148224 ultralytics.nn.modules.block.C2f [384, 128, 1]
13 -1 1 0 torch.nn.modules.upsampling.Upsample [None, 2, 'nearest']
14 [-1, 4] 1 0 ultralytics.nn.modules.conv.Concat [1]
15 -1 1 37248 ultralytics.nn.modules.block.C2f [192, 64, 1]
16 -1 1 36992 ultralytics.nn.modules.conv.Conv [64, 64, 3, 2]
17 [-1, 12] 1 0 ultralytics.nn.modules.conv.Concat [1]
18 -1 1 123648 ultralytics.nn.modules.block.C2f [192, 128, 1]
0 14 Live Share
Cell 1 of 20


File Edit Selection View Go Run Terminal Help
Search
LAB_15 (1).ipynb
C: > Users > ASUS > Downloads > LAB_15 (1).ipynb > Install And Import Essential Libreries
Generate + Code + Markdown | Run All | Clear All Outputs | Outline ...
Select Kernel
17 [-1, 12] 1 0 ultralytics.nn.modules.conv.Concat [1]
18 -1 1 123648 ultralytics.nn.modules.block.C2f [192, 128, 1]
...
Speed Limit 90 38 38 0.275 0.632 0.358 0.307
Stop 81 81 0.945 0.84 0.924 0.826
Speed: 0.4ms preprocess, 5.0ms inference, 0.0ms loss, 1.2ms postprocess per image
Results saved to C:\Users\ASUS\Downloads\runs\detect\train3
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
Validation Step
import os
import cv2
import matplotlib.pyplot as plt
def display_images(post_training_files_path, image_files):
for image_file in image_files:
image_path = os.path.join(post_training_files_path, image_file)
img = cv2.imread(image_path)
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
plt.figure(figsize=(10, 10), dpi=120)
plt.imshow(img)
plt.axis('off')
plt.show()
# List of image files to display
image_files = [
'confusion_matrix_normalized.png',
'BoxF1_curve.png',
[16]
0 14 Live Share
Cell 1 of 20

File Edit Selection View Go Run Terminal Help

Search

LAB_15 (1).ipynb

C: > Users > ASUS > Downloads > LAB_15 (1).ipynb > Install And Import Essential Libreries

Generate + Code + Markdown | Run All | Clear All Outputs | Outline ··· Select Kernel

Validation Step

```
import os
import cv2
import matplotlib.pyplot as plt

def display_images(post_training_files_path, image_files):

    for image_file in image_files:
        image_path = os.path.join(post_training_files_path, image_file)
        img = cv2.imread(image_path)
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

        plt.figure(figsize=(10, 10), dpi=120)
        plt.imshow(img)
        plt.axis('off')
        plt.show()

# List of image files to display
image_files = [
    'confusion_matrix_normalized.png',
    'BoxF1_curve.png',
    'BoxP_curve.png',
    'BoxR_curve.png',
    'BoxPR_curve.png',
    'results.png'
]

# Path to the directory containing the images
post_training_files_path = r"C:\Users\ASUS\Downloads\runs\detect\train3"

# Display the images
display_images(post_training_files_path, image_files)
```

[16]

0 14 Live Share Cell 1 of 20

File Edit Selection View Go Run Terminal Help

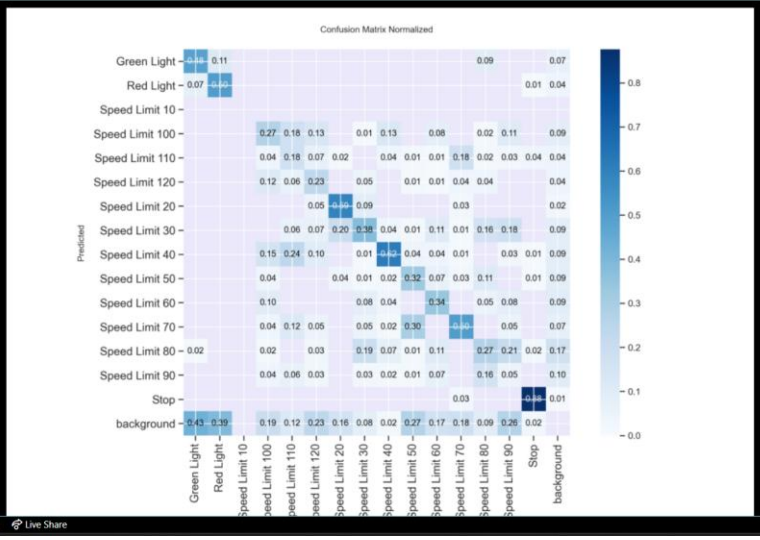
Search

LAB_15 (1).ipynb

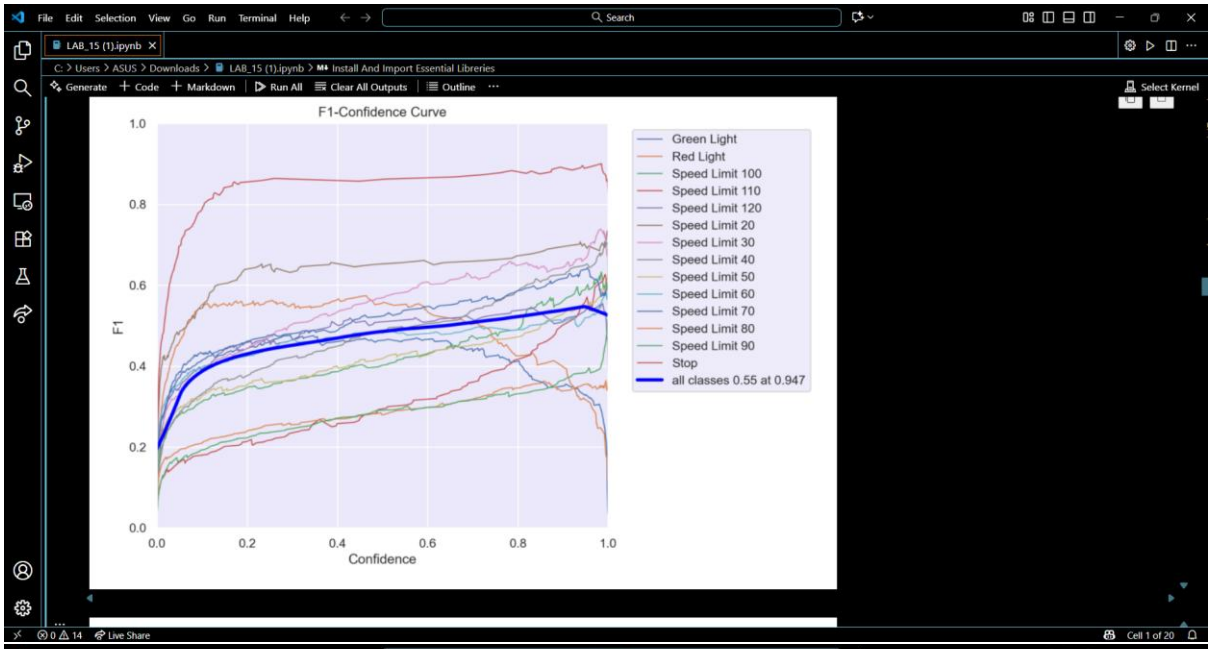
C: > Users > ASUS > Downloads > LAB_15 (1).ipynb > Install And Import Essential Libreries

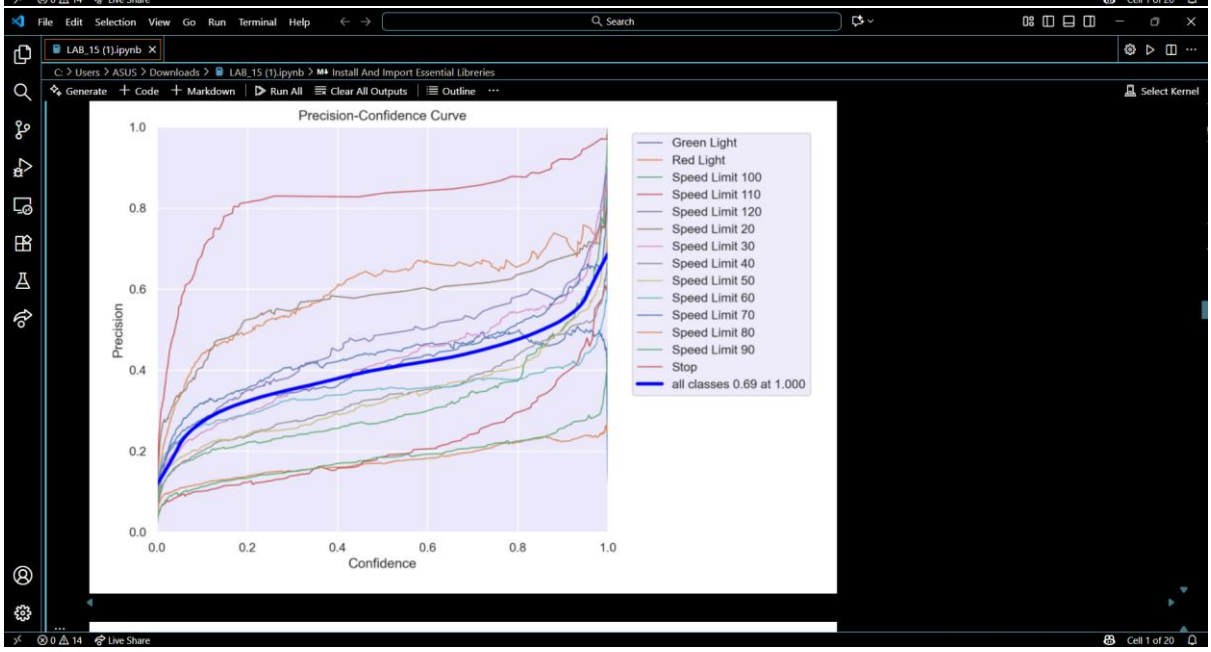
Generate + Code + Markdown | Run All | Clear All Outputs | Outline ··· Select Kernel

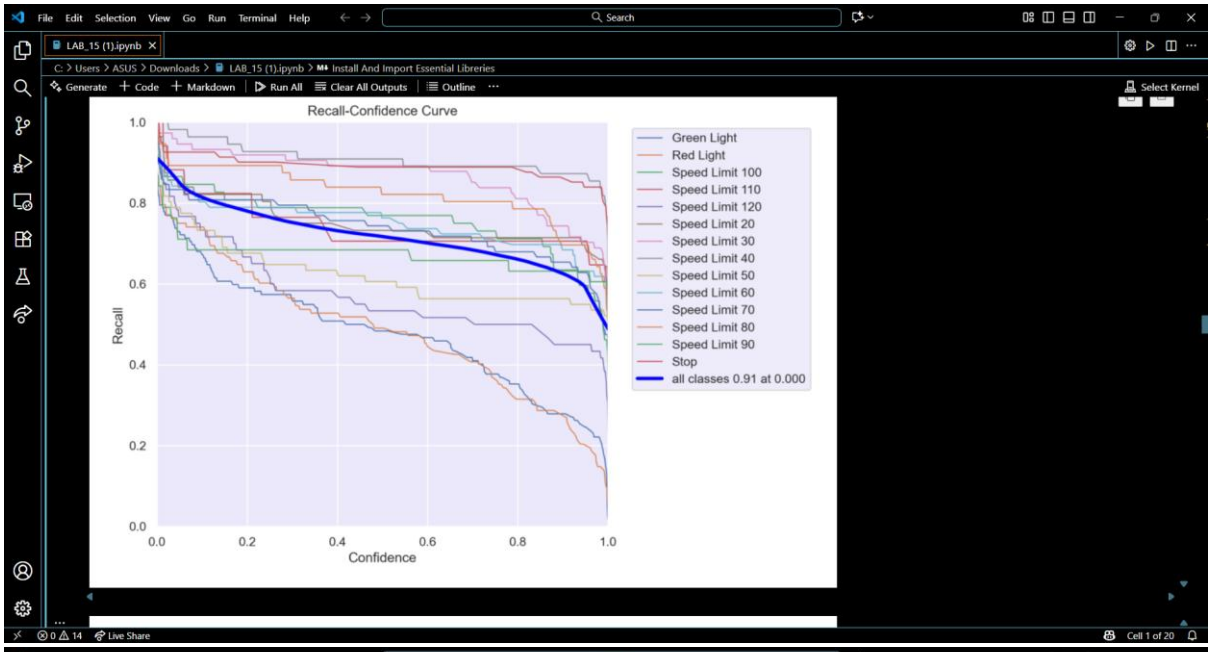
[16] Python

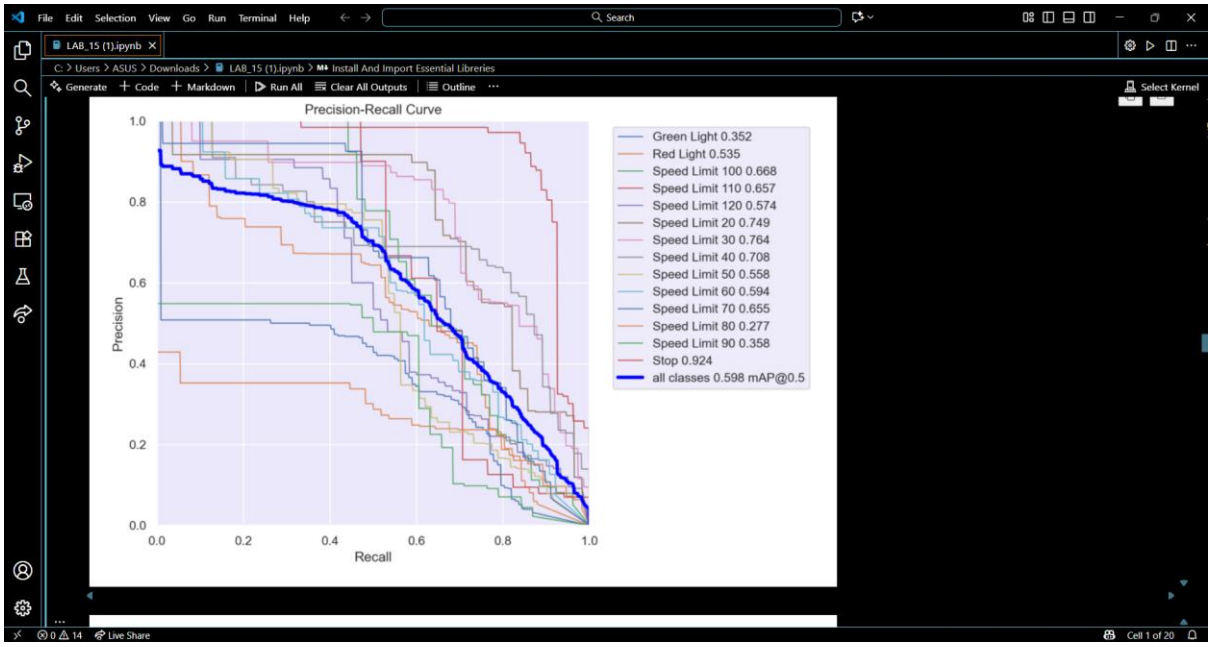


0 14 Live Share Cell 1 of 20


File Edit Selection View Go Run Terminal Help
Search
LAB_15 (1).ipynb
C: > Users > ASUS > Downloads > LAB_15 (1).ipynb > Install And Import Essential Libreries
Generate Code Markdown Run All Clear All Outputs Outline
Select Kernel
F1-Confidence Curve
F1
Confidence
Green Light
Red Light
Speed Limit 100
Speed Limit 110
Speed Limit 120
Speed Limit 20
Speed Limit 30
Speed Limit 40
Speed Limit 50
Speed Limit 60
Speed Limit 70
Speed Limit 80
Speed Limit 90
Stop
all classes 0.55 at 0.947
Live Share
Cell 1 of 20


File Edit Selection View Go Run Terminal Help
Search
LAB_15 (1).ipynb
C: > Users > ASUS > Downloads > LAB_15 (1).ipynb > Install And Import Essential Libreries
Generate Code Markdown Run All Clear All Outputs Outline
Select Kernel
Precision-Confidence Curve
Precision
Confidence
Green Light
Red Light
Speed Limit 100
Speed Limit 110
Speed Limit 120
Speed Limit 20
Speed Limit 30
Speed Limit 40
Speed Limit 50
Speed Limit 60
Speed Limit 70
Speed Limit 80
Speed Limit 90
Stop
all classes 0.69 at 1.000
Live Share
Cell 1 of 20


File Edit Selection View Go Run Terminal Help
Search
LAB_15 (1).ipynb
C: > Users > ASUS > Downloads > LAB_15 (1).ipynb > Install And Import Essential Libreries
Generate Code Markdown Run All Clear All Outputs Outline
Select Kernel
Recall-Confidence Curve
Recall
Confidence
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
Green Light
Red Light
Speed Limit 100
Speed Limit 110
Speed Limit 120
Speed Limit 20
Speed Limit 30
Speed Limit 40
Speed Limit 50
Speed Limit 60
Speed Limit 70
Speed Limit 80
Speed Limit 90
Stop
all classes 0.91 at 0.000
0 14 Live Share
Cell 1 of 20


File Edit Selection View Go Run Terminal Help
Search
LAB_15 (1).ipynb
C: > Users > ASUS > Downloads > LAB_15 (1).ipynb > Install And Import Essential Libreries
Generate Code Markdown Run All Clear All Outputs Outline
Select Kernel
Precision-Recall Curve
Precision
Recall
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
Green Light 0.352
Red Light 0.535
Speed Limit 100 0.668
Speed Limit 110 0.657
Speed Limit 120 0.574
Speed Limit 20 0.749
Speed Limit 30 0.764
Speed Limit 40 0.708
Speed Limit 50 0.558
Speed Limit 60 0.594
Speed Limit 70 0.655
Speed Limit 80 0.277
Speed Limit 90 0.358
Stop 0.924
all classes 0.598 mAP@0.5
0 14 Live Share
Cell 1 of 20


```
Result_Final_model = pd.read_csv(r"C:\Users\ASUS\Downloads\runs\detect\train3\results.csv")
Result_Final_model.tail(10)
```

[17] Python

0 ⚠ 14 Live Share Cell 1 of 20

```
Result_Final_model = pd.read_csv(r"C:\Users\ASUS\Downloads\runs\detect\train3\results.csv")
Result_Final_model.tail(10)
```

[17] Python

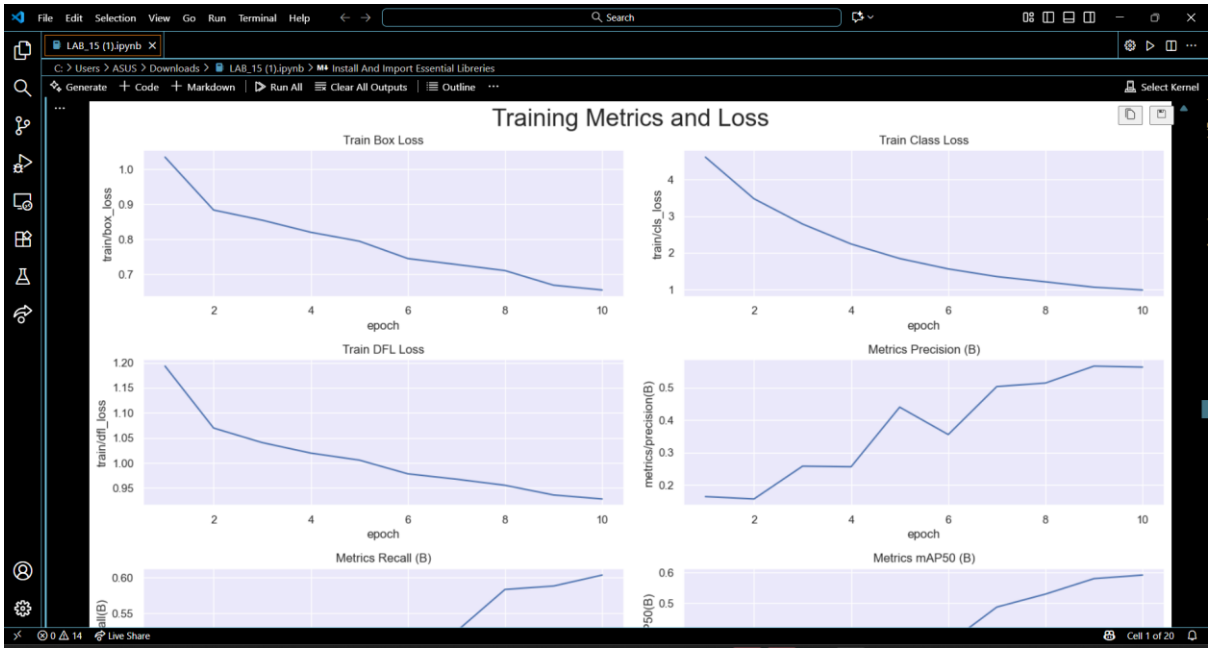
	epoch	time	train/box_loss	train/cls_loss	train/dfl_loss	metrics/precision(B)	metrics/recall(B)	metrics/mAP50(B)	metrics/mAP50-95(B)	val/box_loss	val/cls_loss	val/dfl_loss	lr/pg0	lr/pg1
0	1	351.803	1.03454	4.61360	1.19330	0.16486	0.46959	0.15700	0.12389	0.93672	4.75627	1.22327	0.000175	0.000175
1	2	659.193	0.88350	3.47847	1.06969	0.15753	0.47209	0.17730	0.13779	0.90675	8.59826	1.22401	0.000316	0.000316
2	3	2539.730	0.85455	2.78900	1.04070	0.25848	0.45206	0.26834	0.21459	0.89381	4.02141	1.25550	0.000422	0.000422
3	4	2844.240	0.81998	2.24390	1.01972	0.25667	0.41797	0.28088	0.23024	0.88684	7.08076	1.21504	0.000370	0.000370
4	5	3150.750	0.79453	1.84733	1.00574	0.44015	0.44958	0.39317	0.32277	0.84307	3.69593	1.14722	0.000318	0.000318
5	6	3455.730	0.74490	1.56438	0.97834	0.35576	0.46463	0.36853	0.30036	0.95079	6.14174	1.33585	0.000266	0.000266
6	7	3869.110	0.72795	1.35442	0.96766	0.50356	0.52471	0.48712	0.40636	0.82530	5.51463	1.17455	0.000214	0.000214
7	8	4173.570	0.71054	1.21170	0.95550	0.51431	0.58328	0.52956	0.44566	0.79880	6.20576	1.15376	0.000161	0.000161
8	9	4477.310	0.66878	1.06478	0.93612	0.56645	0.58804	0.58024	0.49109	0.76923	5.48869	1.13798	0.000109	0.000109
9	10	4782.320	0.65508	0.98858	0.92817	0.56331	0.60331	0.59186	0.49904	0.76462	5.01017	1.12541	0.000057	0.000057

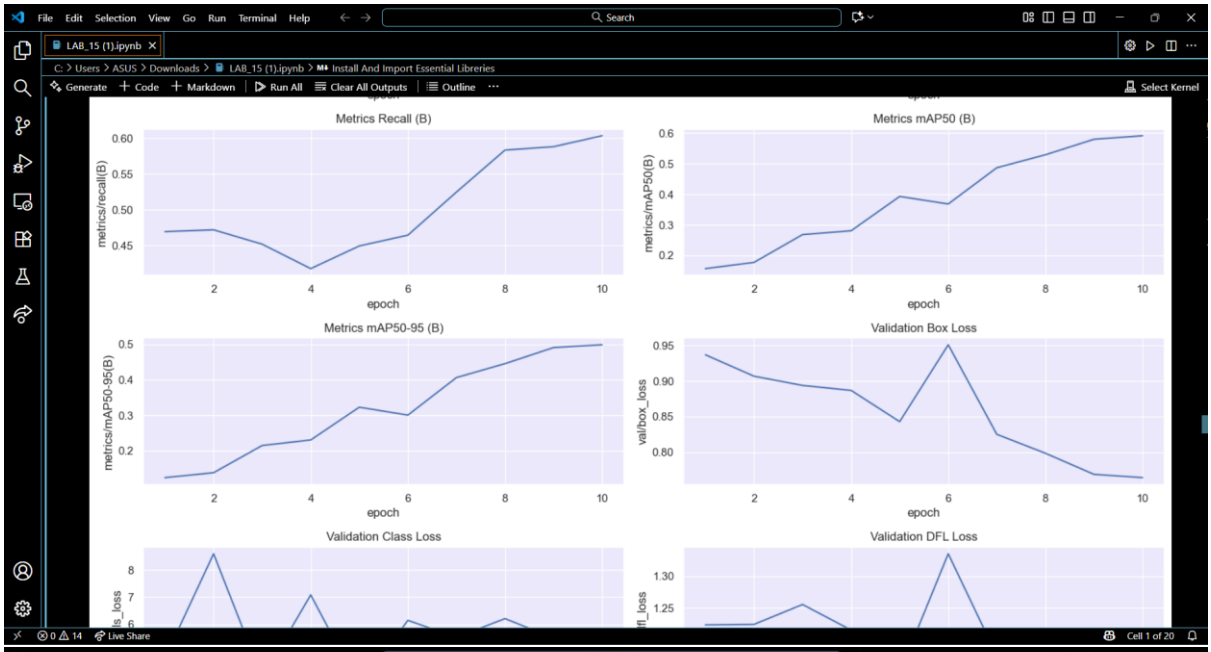
```
# Read the results.csv file as a pandas dataframe
Result_Final_model.columns = Result_Final_model.columns.str.strip()

# Create subplots
fig, axs = plt.subplots(nrows=5, ncols=2, figsize=(15, 15))

# Plot the columns using seaborn
```

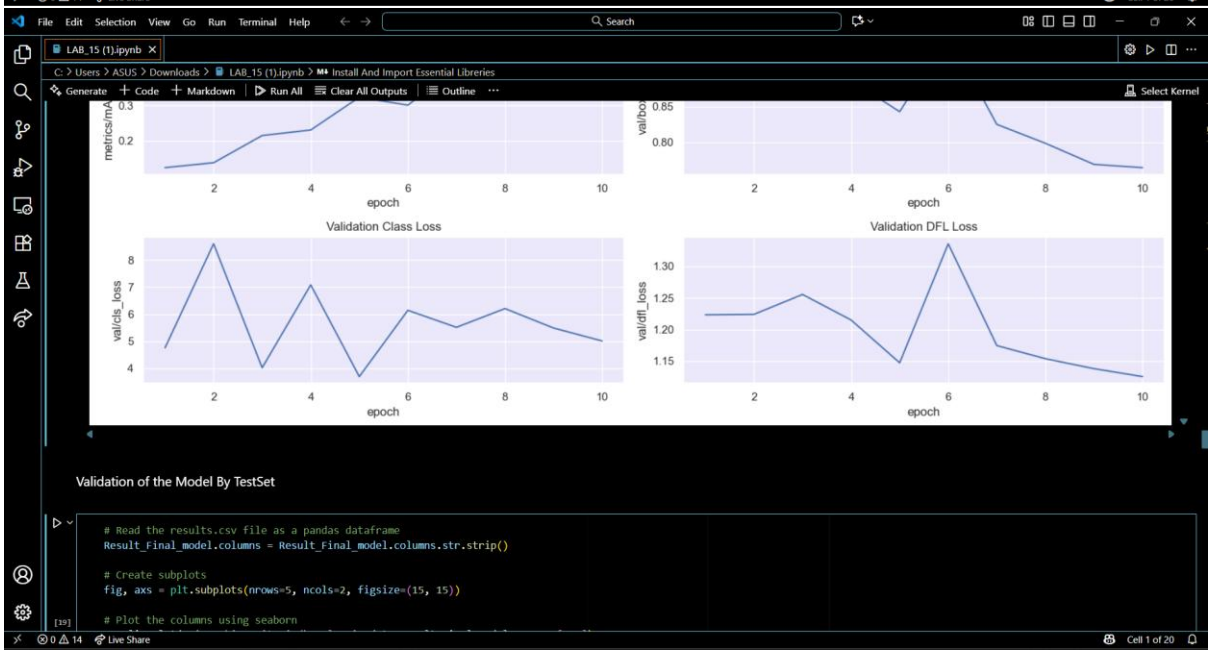
[18]

0 ⚠ 14 Live Share Cell 1 of 20


File Edit Selection View Go Run Terminal Help
Search
LAB_15 (1).ipynb
C: > Users > ASUS > Downloads > LAB_15 (1).ipynb > Install And Import Essential Libreries
Generate + Code + Markdown Run All Clear All Outputs Outline
Select Kernel
9 10 4782.320 0.65508 0.98858 0.92817 0.56331 0.60331 0.59186 0.49904 0.76462 5.01017 1.12541 0.000057 0.000057
# Read the results.csv file as a pandas dataframe
Result_Final_model.columns = Result_Final_model.columns.str.strip()
# Create subplots
fig, axs = plt.subplots(nrows=5, ncols=2, figsize=(15, 15))
# Plot the columns using seaborn
sns.lineplot(x='epoch', y='train/box_loss', data=Result_Final_model, ax=axs[0,0])
sns.lineplot(x='epoch', y='train/cls_loss', data=Result_Final_model, ax=axs[0,1])
sns.lineplot(x='epoch', y='train/dfl_loss', data=Result_Final_model, ax=axs[1,0])
sns.lineplot(x='epoch', y='metrics/precision(B)', data=Result_Final_model, ax=axs[1,1])
sns.lineplot(x='epoch', y='metrics/recall(B)', data=Result_Final_model, ax=axs[2,0])
sns.lineplot(x='epoch', y='metrics/mAP50(B)', data=Result_Final_model, ax=axs[2,1])
sns.lineplot(x='epoch', y='metrics/mAP50-95(B)', data=Result_Final_model, ax=axs[3,0])
sns.lineplot(x='epoch', y='val/box_loss', data=Result_Final_model, ax=axs[3,1])
sns.lineplot(x='epoch', y='val/cls_loss', data=Result_Final_model, ax=axs[4,0])
sns.lineplot(x='epoch', y='val/dfl_loss', data=Result_Final_model, ax=axs[4,1])
# Set titles and axis labels for each subplot
axs[0,0].set(title='Train Box Loss')
axs[0,1].set(title='Train Class Loss')
axs[1,0].set(title='Train DFL Loss')
axs[1,1].set(title='Metrics Precision (B)')
axs[2,0].set(title='Metrics Recall (B)')
axs[2,1].set(title='Metrics mAP50 (B)')
axs[3,0].set(title='Metrics mAP50-95 (B)')
axs[3,1].set(title='Validation Box Loss')
axs[4,0].set(title='Validation Class Loss')
axs[4,1].set(title='Validation DFL Loss')
[18]
0 14 Live Share
Cell 1 of 20


File Edit Selection View Go Run Terminal Help
Search
LAB_15 (1).ipynb
C: > Users > ASUS > Downloads > LAB_15 (1).ipynb > Install And Import Essential Libreries
Generate + Code + Markdown Run All Clear All Outputs Outline
Select Kernel
Training Metrics and Loss
Train Box Loss
train/box_loss
epoch
Train Class Loss
train/cls_loss
epoch
Train DFL Loss
train/dfl_loss
epoch
Metrics Precision (B)
metrics/precision(B)
epoch
Metrics Recall (B)
Metrics mAP50 (B)
0 14 Live Share
Cell 1 of 20





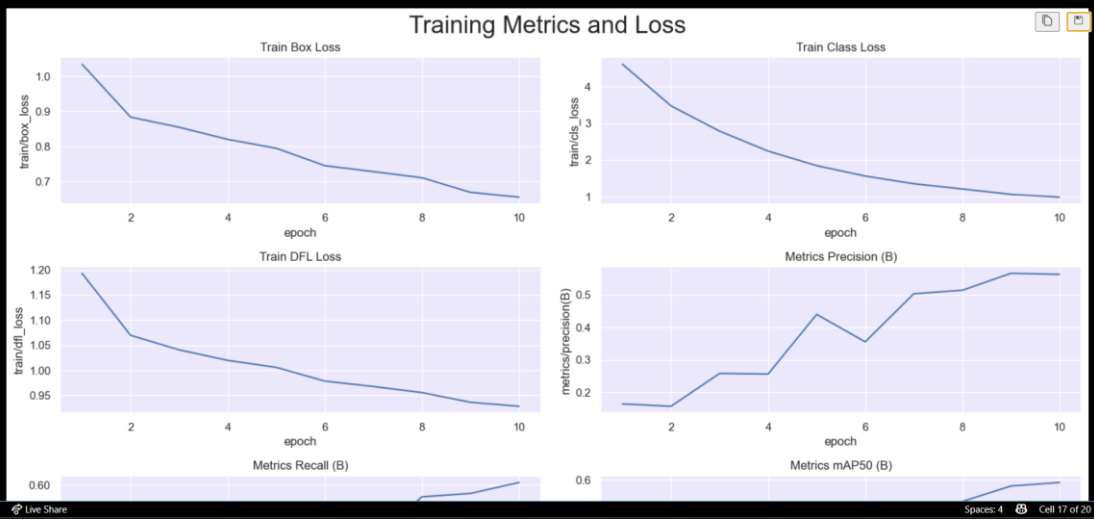
Validation of the Model By TestSet

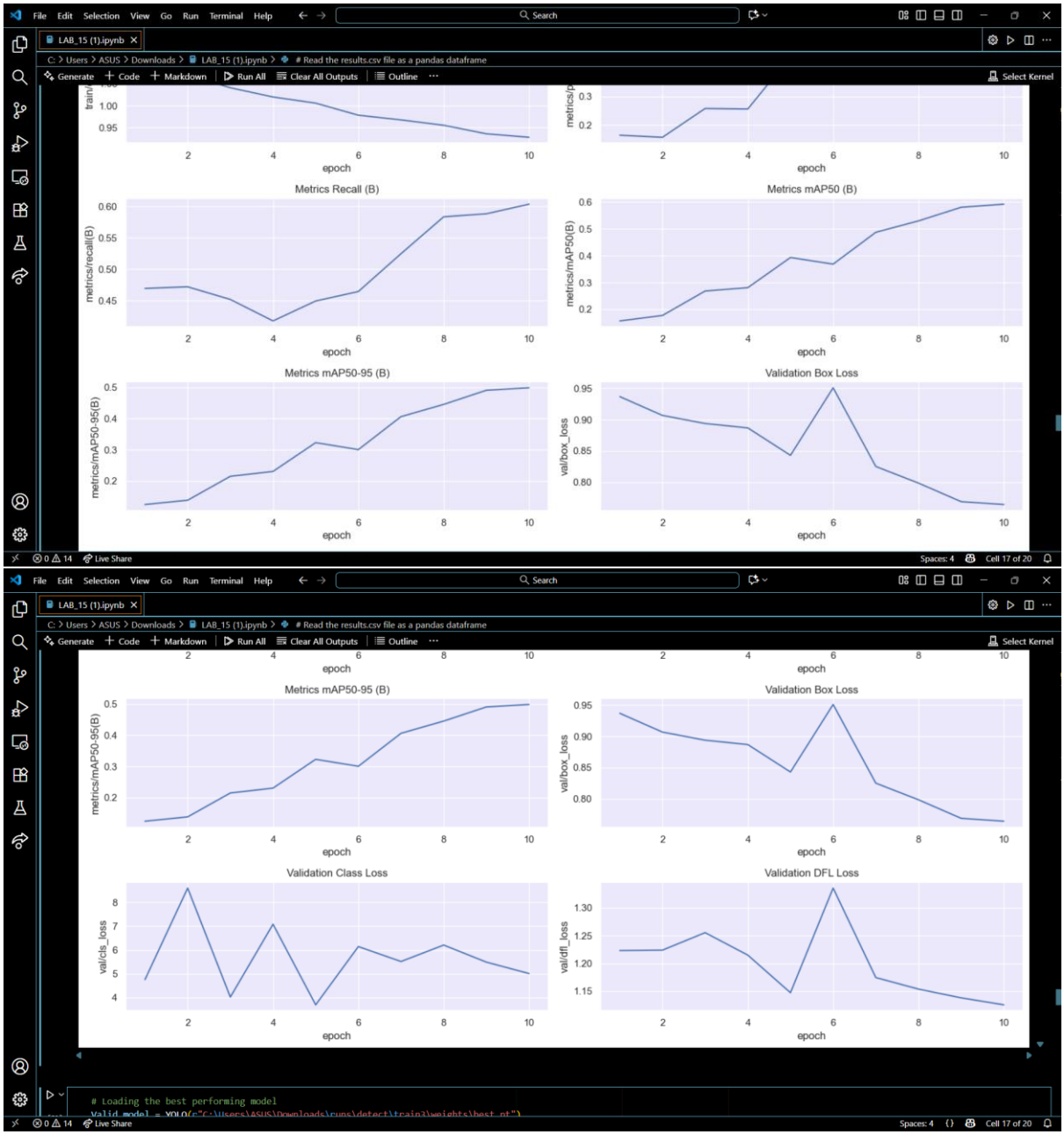
```
# Read the results.csv file as a pandas dataframe
Result_Final_model.columns = Result_Final_model.columns.str.strip()

# Create subplots
fig, axs = plt.subplots(nrows=5, ncols=2, figsize=(15, 15))

# Plot the columns using seaborn
```

Validation of the Model By TestSet

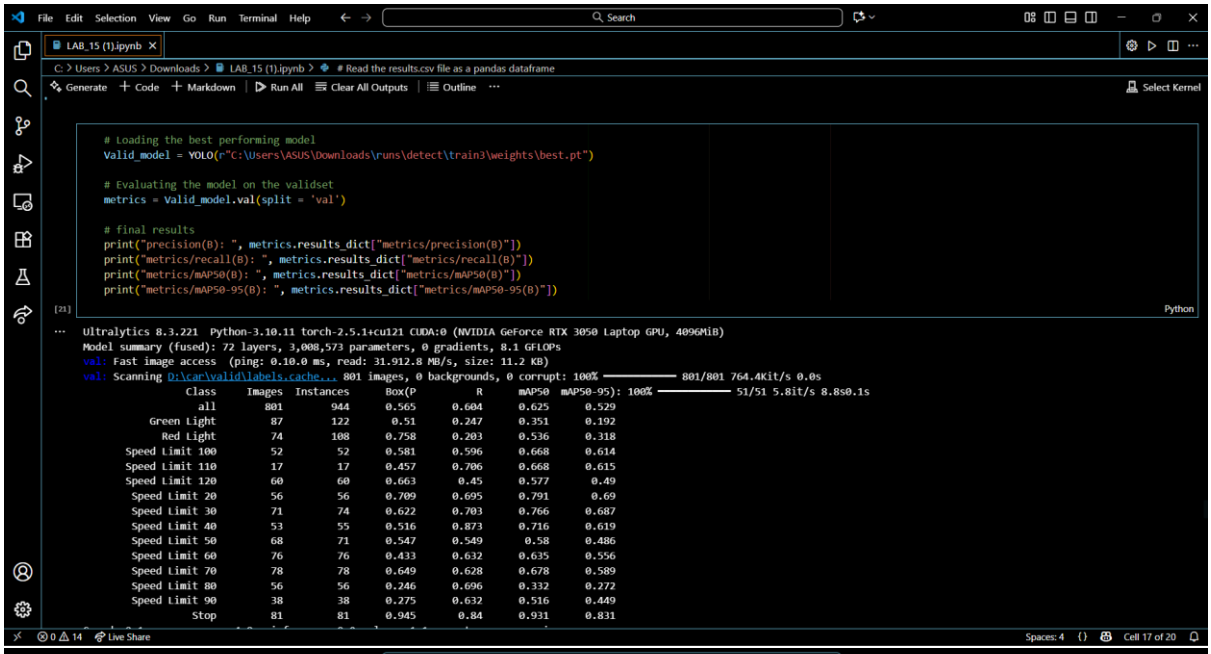
```
# Read the results.csv file as a pandas dataframe
Result_Final_model.columns = Result_Final_model.columns.str.strip()

# Create subplots
fig, axs = plt.subplots(nrows=5, ncols=2, figsize=(15, 15))

# Plot the columns using seaborn
sns.lineplot(x='epoch', y='train/box_loss', data=Result_Final_model, ax=axs[0,0])
sns.lineplot(x='epoch', y='train/cls_loss', data=Result_Final_model, ax=axs[0,1])
sns.lineplot(x='epoch', y='train/dfl_loss', data=Result_Final_model, ax=axs[1,0])
sns.lineplot(x='epoch', y='metrics/precision(B)', data=Result_Final_model, ax=axs[1,1])
sns.lineplot(x='epoch', y='metrics/recall(B)', data=Result_Final_model, ax=axs[2,0])
sns.lineplot(x='epoch', y='metrics/mAP50(B)', data=Result_Final_model, ax=axs[2,1])
sns.lineplot(x='epoch', y='metrics/mAP50-95(B)', data=Result_Final_model, ax=axs[3,0])
sns.lineplot(x='epoch', y='val/box_loss', data=Result_Final_model, ax=axs[3,1])
sns.lineplot(x='epoch', y='val/cls_loss', data=Result_Final_model, ax=axs[4,0])
sns.lineplot(x='epoch', y='val/dfl_loss', data=Result_Final_model, ax=axs[4,1])

# Set titles and axis labels for each subplot
axs[0,0].set(title='Train Box Loss')
axs[0,1].set(title='Train Class Loss')
axs[1,0].set(title='Train DFL Loss')
axs[1,1].set(title='Metrics Precision (B)')
axs[2,0].set(title='Metrics Recall (B)')
axs[2,1].set(title='Metrics mAP50 (B)')
axs[3,0].set(title='Metrics mAP50-95 (B)')
axs[3,1].set(title='Validation Box Loss')
axs[4,0].set(title='Validation Class Loss')
axs[4,1].set(title='Validation DFL Loss')
```

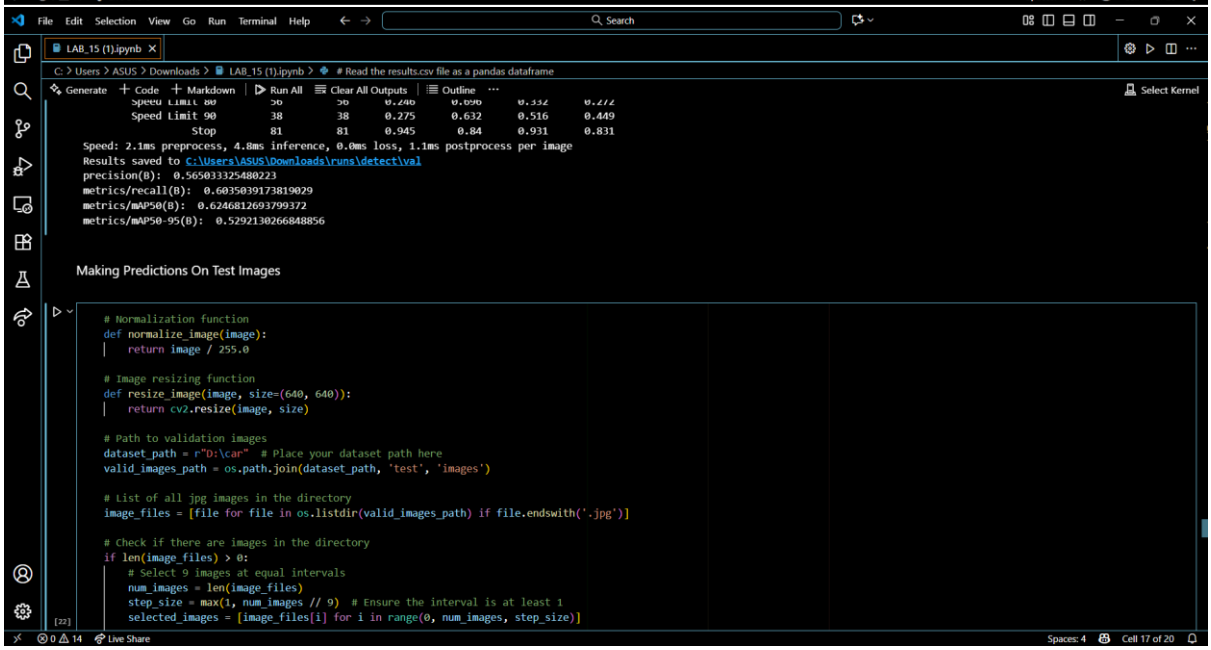
```
pit.show()
```

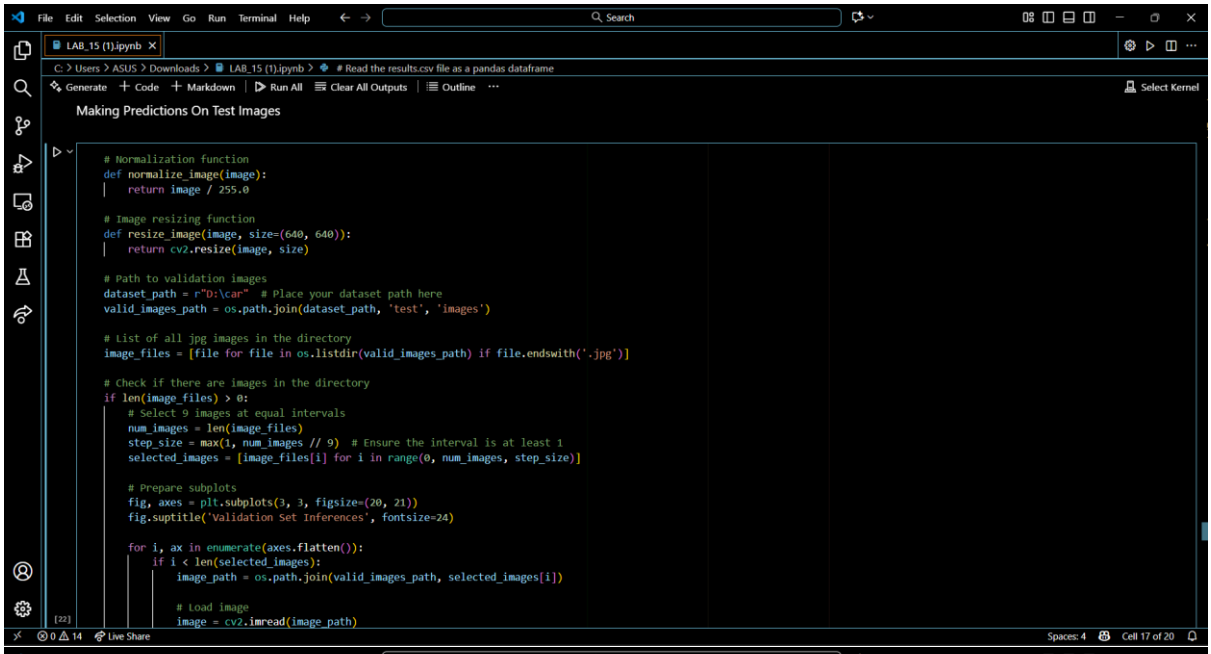


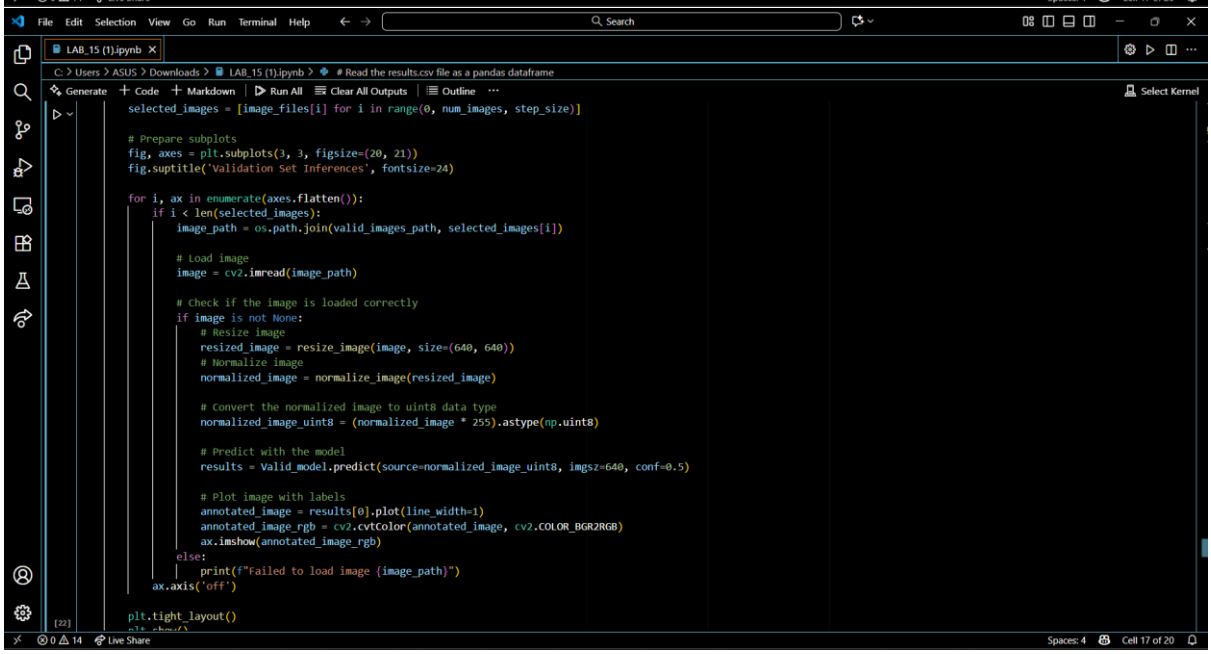

File Edit Selection View Go Run Terminal Help
Search
LAB_15 (1).ipynb
C: > Users > ASUS > Downloads > LAB_15 (1).ipynb > # Read the results.csv file as a pandas dataframe
Generate Code Markdown Run All Clear All Outputs Outline
Select Kernel
metrics/recall(B)
Metrics Recall (B)
metrics/mAP50(B)
Metrics mAP50 (B)
metrics/mAP50-95(B)
Metrics mAP50-95 (B)
val/box_loss
Validation Box Loss
epoch
Live Share
Spaces: 4
Cell 17 of 20

File Edit Selection View Go Run Terminal Help
Search
LAB_15 (1).ipynb
C: > Users > ASUS > Downloads > LAB_15 (1).ipynb > # Read the results.csv file as a pandas dataframe
Generate Code Markdown Run All Clear All Outputs Outline
Select Kernel
metrics/mAP50-95(B)
Metrics mAP50-95 (B)
val/box_loss
Validation Box Loss
val/cls_loss
Validation Class Loss
val/dfl_loss
Validation DFL Loss
epoch
Loading the best performing model
Live Share
Spaces: 4
Cell 17 of 20


File Edit Selection View Go Run Terminal Help
Search
LAB_15 (1).ipynb
C: > Users > ASUS > Downloads > LAB_15 (1).ipynb > # Read the results.csv file as a pandas dataframe
Generate + Code + Markdown | Run All Clear All Outputs | Outline ···
Select Kernel
# Loading the best performing model
Valid_model = YOLO(r"C:\Users\ASUS\Downloads\runs\detect\train3\weights\best.pt")
# Evaluating the model on the validset
metrics = Valid_model.val(split = 'val')
# final results
print("precision(B): ", metrics.results_dict["metrics/precision(B)"])
print("metrics/recall(B): ", metrics.results_dict["metrics/recall(B)"])
print("metrics/mAP50(B): ", metrics.results_dict["metrics/mAP50(B)"])
print("metrics/mAP50-95(B): ", metrics.results_dict["metrics/mAP50-95(B)"])
[21]
Python
Ultralytics 8.3.221 Python-3.10.11 torch-2.5.1+cu121 CUDA:0 (NVIDIA GeForce RTX 3050 Laptop GPU, 4096MiB)
Model summary (fused): 72 layers, 3,008,573 parameters, 0 gradients, 8.1 GFLOPs
val: Fast image access (ping: 0.10.0 ms, read: 31.912.8 MB/s, size: 11.2 KB)
val: Scanning D:\car\valid\labels.cache... 801 images, 0 backgrounds, 0 corrupt: 100% ━━━━━━━━━━━━ 801/801 764.4Kit/s 0.0s
Class Images Instances Box(P R mAP50 mAP50-95): 100% ━━━━━━━━━━━━ 51/51 5.8it/s 8.8s0.1s
all 801 944 0.565 0.604 0.625 0.529
Green Light 87 122 0.51 0.247 0.351 0.192
Red Light 74 108 0.758 0.203 0.536 0.318
Speed Limit 100 52 52 0.581 0.596 0.668 0.614
Speed Limit 110 17 17 0.457 0.706 0.668 0.615
Speed Limit 120 60 60 0.663 0.45 0.577 0.49
Speed Limit 20 56 56 0.709 0.695 0.791 0.69
Speed Limit 30 71 74 0.622 0.703 0.766 0.687
Speed Limit 40 53 55 0.516 0.873 0.716 0.619
Speed Limit 50 68 71 0.547 0.549 0.58 0.486
Speed Limit 60 76 76 0.433 0.632 0.635 0.556
Speed Limit 70 78 78 0.649 0.628 0.678 0.589
Speed Limit 80 56 56 0.246 0.696 0.332 0.272
Speed Limit 90 38 38 0.275 0.632 0.516 0.449
Stop 81 81 0.945 0.84 0.931 0.831
0 14 Live Share
Spaces: 4 Cell 17 of 20


File Edit Selection View Go Run Terminal Help
Search
LAB_15 (1).ipynb
C: > Users > ASUS > Downloads > LAB_15 (1).ipynb > # Read the results.csv file as a pandas dataframe
Generate + Code + Markdown | Run All Clear All Outputs | Outline ···
Select Kernel
Speed Limit 80 56 56 0.246 0.696 0.332 0.272
Speed Limit 90 38 38 0.275 0.632 0.516 0.449
Stop 81 81 0.945 0.84 0.931 0.831
Speed: 2.1ms preprocess, 4.8ms inference, 0.0ms loss, 1.1ms postprocess per image
Results saved to C:\Users\ASUS\Downloads\runs\detect\val
precision(B): 0.565033325480223
metrics/recall(B): 0.6035039173819029
metrics/mAP50(B): 0.6246812693799372
metrics/mAP50-95(B): 0.5292130266848856
Making Predictions On Test Images
# Normalization function
def normalize_image(image):
return image / 255.0
# Image resizing function
def resize_image(image, size=(640, 640)):
return cv2.resize(image, size)
# Path to validation images
dataset_path = r"D:\car" # Place your dataset path here
valid_images_path = os.path.join(dataset_path, 'test', 'images')
# List of all jpg images in the directory
image_files = [file for file in os.listdir(valid_images_path) if file.endswith('.jpg')]
# Check if there are images in the directory
if len(image_files) > 0:
# Select 9 images at equal intervals
num_images = len(image_files)
step_size = max(1, num_images // 9) # Ensure the interval is at least 1
selected_images = [image_files[i] for i in range(0, num_images, step_size)]
[22]
0 14 Live Share
Spaces: 4 Cell 17 of 20


File Edit Selection View Go Run Terminal Help
Search
LAB_15 (1).ipynb
C: > Users > ASUS > Downloads > LAB_15 (1).ipynb > # Read the results.csv file as a pandas dataframe
Generate + Code + Markdown Run All Clear All Outputs Outline ...
Select Kernel
Making Predictions On Test Images
# Normalization function
def normalize_image(image):
return image / 255.0
# Image resizing function
def resize_image(image, size=(640, 640)):
return cv2.resize(image, size)
# Path to validation images
dataset_path = r"D:\car" # Place your dataset path here
valid_images_path = os.path.join(dataset_path, 'test', 'images')
# List of all jpg images in the directory
image_files = [file for file in os.listdir(valid_images_path) if file.endswith('.jpg')]
# Check if there are images in the directory
if len(image_files) > 0:
# Select 9 images at equal intervals
num_images = len(image_files)
step_size = max(1, num_images // 9) # Ensure the interval is at least 1
selected_images = [image_files[i] for i in range(0, num_images, step_size)]
# Prepare subplots
fig, axes = plt.subplots(3, 3, figsize=(20, 21))
fig.suptitle('Validation Set Inferences', fontsize=24)
for i, ax in enumerate(axes.flatten()):
if i < len(selected_images):
image_path = os.path.join(valid_images_path, selected_images[i])
# Load image
image = cv2.imread(image_path)
[22]
0 14 Live Share
Spaces: 4 Cell 17 of 20


File Edit Selection View Go Run Terminal Help
Search
LAB_15 (1).ipynb
C: > Users > ASUS > Downloads > LAB_15 (1).ipynb > # Read the results.csv file as a pandas dataframe
Generate + Code + Markdown Run All Clear All Outputs Outline ...
Select Kernel
selected_images = [image_files[i] for i in range(0, num_images, step_size)]
# Prepare subplots
fig, axes = plt.subplots(3, 3, figsize=(20, 21))
fig.suptitle('Validation Set Inferences', fontsize=24)
for i, ax in enumerate(axes.flatten()):
if i < len(selected_images):
image_path = os.path.join(valid_images_path, selected_images[i])
# Load image
image = cv2.imread(image_path)
# Check if the image is loaded correctly
if image is not None:
# Resize image
resized_image = resize_image(image, size=(640, 640))
# Normalize image
normalized_image = normalize_image(resized_image)
# Convert the normalized image to uint8 data type
normalized_image_uint8 = (normalized_image * 255).astype(np.uint8)
# Predict with the model
results = Valid_model.predict(source=normalized_image_uint8, imgsz=640, conf=0.5)
# Plot image with labels
annotated_image = results[0].plot(line_width=1)
annotated_image_rgb = cv2.cvtColor(annotated_image, cv2.COLOR_BGR2RGB)
ax.imshow(annotated_image_rgb)
else:
print(f"Failed to load image {image_path}")
ax.axis('off')
plt.tight_layout()
[22]
0 14 Live Share
Spaces: 4 Cell 17 of 20


File Edit Selection View Go Run Terminal Help
Search
LAB_15 (1).ipynb
C: > Users > ASUS > Downloads > LAB_15 (1).ipynb > # Read the results.csv file as a pandas dataframe
Generate Code Markdown Run All Clear All Outputs Outline
Select Kernel
ax.axis('off')
plt.tight_layout()
plt.show()
[22]
Python
0: 640x640 1 Speed Limit 30, 1 Speed Limit 60, 58.5ms
Speed: 5.0ms preprocess, 58.5ms inference, 3.6ms postprocess per image at shape (1, 3, 640, 640)
0: 640x640 1 Speed Limit 30, 58.1ms
Speed: 4.2ms preprocess, 58.1ms inference, 7.2ms postprocess per image at shape (1, 3, 640, 640)
0: 640x640 1 Speed Limit 70, 8.1ms
Speed: 1.3ms preprocess, 8.1ms inference, 1.3ms postprocess per image at shape (1, 3, 640, 640)
0: 640x640 1 Stop, 8.1ms
Speed: 1.6ms preprocess, 8.1ms inference, 1.4ms postprocess per image at shape (1, 3, 640, 640)
0: 640x640 2 Speed Limit 70s, 8.3ms
Speed: 1.1ms preprocess, 8.3ms inference, 1.3ms postprocess per image at shape (1, 3, 640, 640)
0: 640x640 2 Green Lights, 9.8ms
Speed: 1.3ms preprocess, 9.8ms inference, 1.4ms postprocess per image at shape (1, 3, 640, 640)
0: 640x640 1 Red Light, 8.1ms
Speed: 1.1ms preprocess, 8.1ms inference, 1.4ms postprocess per image at shape (1, 3, 640, 640)
0: 640x640 (no detections), 8.1ms
Speed: 1.3ms preprocess, 8.1ms inference, 0.7ms postprocess per image at shape (1, 3, 640, 640)
0: 640x640 1 Speed Limit 60, 1 Speed Limit 80, 1 Speed Limit 90, 8.0ms
Speed: 1.2ms preprocess, 8.0ms inference, 1.6ms postprocess per image at shape (1, 3, 640, 640)
0 14 Live Share
Spaces: 4
Cell 17 of 20

File Edit Selection View Go Run Terminal Help
Search
LAB_15 (1).ipynb
C: > Users > ASUS > Downloads > LAB_15 (1).ipynb > # Read the results.csv file as a pandas dataframe
Generate Code Markdown Run All Clear All Outputs Outline
Select Kernel
0: 640x640 1 Stop, 8.1ms
Speed: 1.6ms preprocess, 8.1ms inference, 1.4ms postprocess per image at shape (1, 3, 640, 640)
0: 640x640 2 Speed Limit 70s, 8.3ms
Speed: 1.1ms preprocess, 8.3ms inference, 1.3ms postprocess per image at shape (1, 3, 640, 640)
0: 640x640 2 Green Lights, 9.8ms
Speed: 1.3ms preprocess, 9.8ms inference, 1.4ms postprocess per image at shape (1, 3, 640, 640)
0: 640x640 1 Red Light, 8.1ms
Speed: 1.1ms preprocess, 8.1ms inference, 1.4ms postprocess per image at shape (1, 3, 640, 640)
0: 640x640 (no detections), 8.1ms
Speed: 1.3ms preprocess, 8.1ms inference, 0.7ms postprocess per image at shape (1, 3, 640, 640)
0: 640x640 1 Speed Limit 60, 1 Speed Limit 80, 1 Speed Limit 90, 8.0ms
Speed: 1.2ms preprocess, 8.0ms inference, 1.6ms postprocess per image at shape (1, 3, 640, 640)
Validation Set Inferences
30
30
0 14 Live Share
Spaces: 4
Cell 17 of 20


File Edit Selection View Go Run Terminal Help
Search
LAB_15 (1).ipynb
C: > Users > ASUS > Downloads > LAB_15 (1).ipynb > # Read the results.csv file as a pandas dataframe
Generate Code Markdown Run All Clear All Outputs Outline
Select Kernel
STOP
70
70
© Can Stock Photo
Live Share
Spaces: 4
Cell 17 of 20

File Edit Selection View Go Run Terminal Help
Search
LAB_15 (1).ipynb
C: > Users > ASUS > Downloads > LAB_15 (1).ipynb > # Read the results.csv file as a pandas dataframe
Generate Code Markdown Run All Clear All Outputs Outline
Select Kernel
© Can Stock Photo
90
Live Share
Spaces: 4
Cell 17 of 20