

```
# importing car object detection dataset from Kaggle Library

import kagglehub
sshikamaru_car_object_detection_path = kagglehub.dataset_download('sshikamaru/car-object-detection')

print('Data source import complete.')

→ Downloading from https://www.kaggle.com/api/v1/datasets/download/sshikamaru/car-object-detection?dataset\_version\_number=2...
100%|██████████| 112M/112M [00:06<00:00, 19.5MB/s]Extracting files...

Data source import complete.

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# using OS level directory traversing to display the list of images from training and test directories of the dataset
import os
for dirname, _, filenames in os.walk('/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

→ /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/sample_submission.csv
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/train_solution_bounding_boxes (1).csv
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_29560.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_28620.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_29440.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_30120.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_29540.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_28020.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_27260.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_28560.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_29600.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_27400.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_30160.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_26620.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_27760.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_29020.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_29460.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_27580.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_26840.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_27900.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_27840.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_27300.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_29040.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_26940.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_26880.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_27360.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_27420.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_27440.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_25200.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_29620.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_29820.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_29580.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_30740.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_27500.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_26660.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_31500.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_27520.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_31120.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_26960.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_31080.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_30840.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_26860.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_27920.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_28320.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_29740.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_27320.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_27560.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_29900.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_27480.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_28080.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_26580.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_28540.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_31180.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_30180.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_29520.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_27540.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_30000.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_26820.jpg
```

```
!pip install ultralytics
```

Show hidden output

```
# Install and import python libraries

!pip install -U ipywidgets
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt # pyplot for plotting
import seaborn as sns # seaborn for plotting
from sklearn.model_selection import train_test_split
from glob import glob
import cv2
from PIL import Image
from ultralytics import YOLO
import os
```

Show hidden output

```
# creating directories for working session and training model

!mkdir -p "/content/working/data"
!mkdir -p "/content/working/data/images"
!mkdir -p "/content/working/data/images/train"
!mkdir -p "/content/working/data/images/val"
!mkdir -p "/content/working/data/labels"
!mkdir -p "/content/working/data/labels/train"
!mkdir -p "/content/working/data/labels/val"

root_dir = "/content/working/data"
labels_dir = "/content/working/data/labels"
images_dir = "/content/working/data/images"
```

```
train_data = "/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/training_images"
csv_data = "/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/train_solution_bounding_boxes_(1).cs
test_data = "/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images"
```

```
# Read the input dataset and display first few records
```

```
df = pd.read_csv(csv_data)
df.head()
```

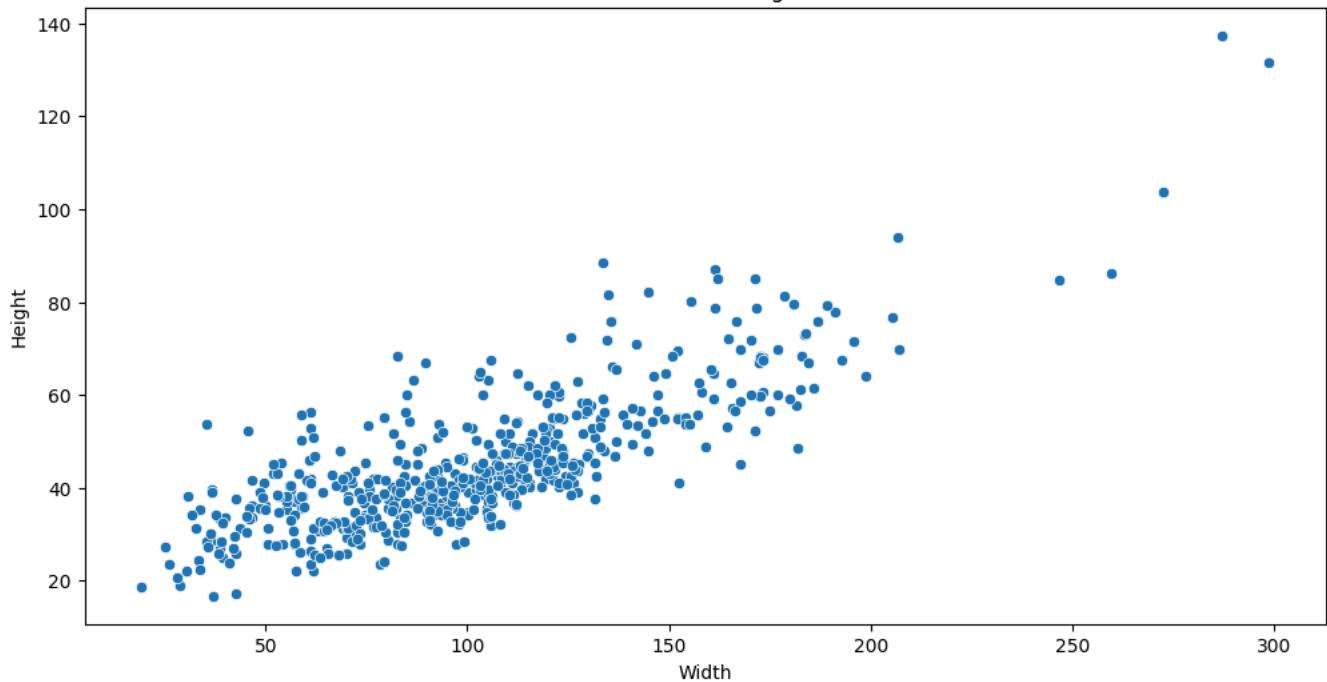
	image	xmin	ymin	xmax	ymax
0	vid_4_1000.jpg	281.259045	187.035071	327.727931	223.225547
1	vid_4_10000.jpg	15.163531	187.035071	120.329957	236.430180
2	vid_4_10040.jpg	239.192475	176.764801	361.968162	236.430180
3	vid_4_10020.jpg	496.483358	172.363256	630.020260	231.539575
4	vid_4_10060.jpg	16.630970	186.546010	132.558611	238.386422

```
# Plotting for distribution of height and width of cars in the input dataset
```

```
plt.figure(figsize=(12, 6))
df['width'] = df['xmax'] - df['xmin']
df['height'] = df['ymax'] - df['ymin']
sns.scatterplot(x='width', y='height', data=df)
plt.title('Distribution of bounding box sizes')
plt.xlabel('Width')
plt.ylabel('Height')
plt.show()
```



## Distribution of bounding box sizes



```
#Plotting for bounding box width and height of the cars

df['box_width'] = df['xmax'] - df['xmin']
df['box_height'] = df['ymax'] - df['ymin']

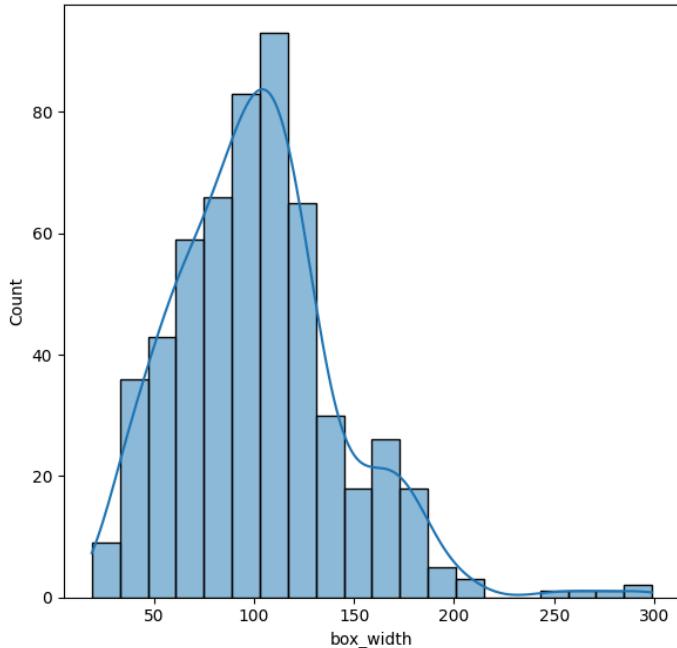
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
sns.histplot(df['box_width'], bins=20, kde=True)
plt.title('Distribution of Bounding Box Widths')

plt.subplot(1, 2, 2)
sns.histplot(df['box_height'], bins=20, kde=True)
plt.title('Distribution of Bounding Box Heights')

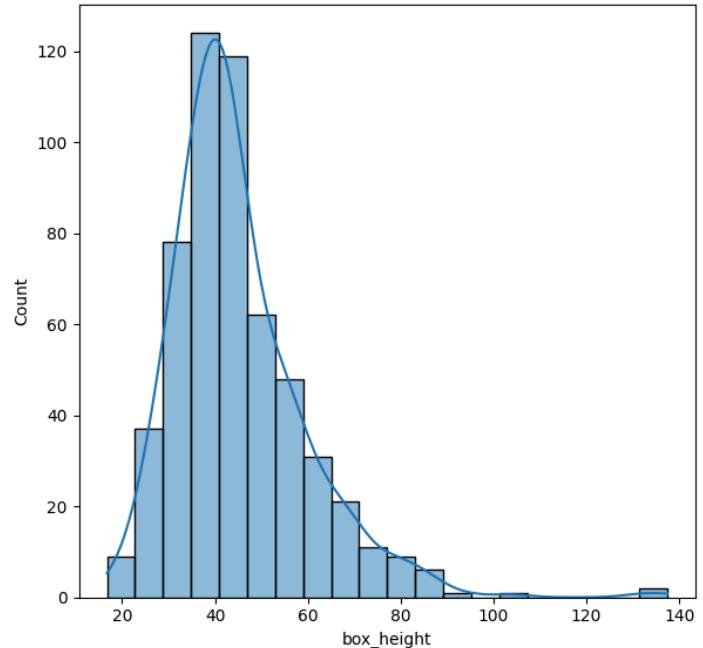
plt.tight_layout();
plt.show();
```



Distribution of Bounding Box Widths



Distribution of Bounding Box Heights

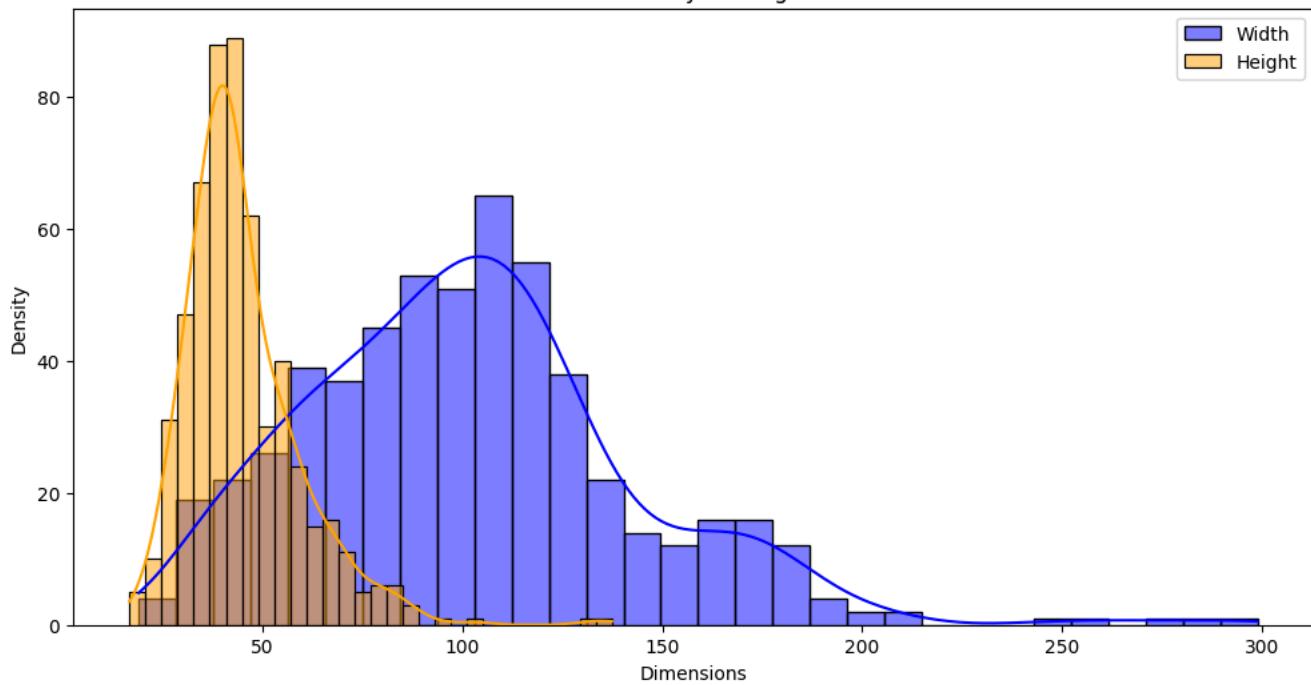


```
#Plotting density of cars found in the images
```

```
plt.figure(figsize=(12, 6))
sns.histplot(df['width'], bins=30, kde=True, color='blue', label='Width')
sns.histplot(df['height'], bins=30, kde=True, color='orange', label='Height')
plt.title('Car density in images')
plt.xlabel('Dimensions')
plt.ylabel('Density')
plt.legend()
plt.show()
```



Car density in images



```
#Identifying and displaying cars in the input dataset
#The images are in RGB format.
```

```
def plot_images_with_boxes(image_path, df, n=12):
    images = glob(f'{image_path}/*.jpg')[:n]
    for img_path in images:
        img = cv2.imread(img_path)
        img_name = os.path.basename(img_path)
        boxes = df[df['image'] == img_name]

        for _, box in boxes.iterrows():
            cv2.rectangle(img,
                          (int(box['xmin']), int(box['ymin'])),
                          (int(box['xmax']), int(box['ymax'])),
                          (0,0,255),2)

    plt.figure(figsize=(8, 8))
    plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
    plt.title(img_name)
    plt.axis('off')
    plt.show()
plot_images_with_boxes(train_data, df)
```



vid\_4\_22160.jpg



vid\_4\_12900.jpg



vid\_4\_9020.jpg



```
def create_yolo_annotation(row, img_width, img_height):
    x_center = ((row['xmin'] + row['xmax']) / 2) / img_width
    y_center = ((row['ymin'] + row['ymax']) / 2) / img_height
```

```

width = (row['xmax'] - row['xmin']) / img_width
height = (row['ymax'] - row['ymin']) / img_height
return f"0 {x_center} {y_center} {width} {height}"
```

---

```

for img_name in df['image'].unique():
    img_df = df[df['image'] == img_name]
    img_path = os.path.join(train_data, img_name)
    img = cv2.imread(img_path)
    if img is not None:
        img_height, img_width = img.shape[:2]

        if np.random.rand() < 0.8:
            subset = "train"
        else:
            subset = "val"

        dst_img_path = os.path.join(images_dir, subset, img_name)
        cv2.imwrite(dst_img_path, img)

        annotation_path = os.path.join(labels_dir, subset, f"{img_name.split('.')[0]}.txt")
        with open(annotation_path, 'w') as f:
            for _, row in img_df.iterrows():
                yolo_annotation = create_yolo_annotation(row, img_width, img_height)
                f.write(yolo_annotation + '\n')
```

#YAML configuration for YOLO model training

```

yaml_content = f"""
path: {root_dir}
train: images/train
val: images/val

nc: 1
names: ['car']
"""

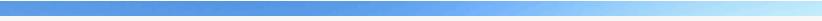
with open('car_detection.yaml', 'w') as f:
    f.write(yaml_content)
print("YAML configuration file created.")
```

→ YAML configuration file created.

#selecting the YOLO pre-training model for training against our dataset

```
model = YOLO('yolov8s.pt')
```

→ Downloading <https://github.com/ultralytics/assets/releases/download/v8.3.0/yolov8s.pt> to 'yolov8s.pt'...
100% [██████████] | 21.5M/21.5M [00:00<00:00, 364MB/s]



# Trainng the model with 100 epochs and storing the results

```
results = model.train(
    data='car_detection.yaml',
    epochs=100,
    imgsz=640,
    batch=16,
    name='car_detection_model'
)
```

→

12/10/24, 12:39 AM

Final_Mkaturi_Troy_AAI-521_Final_Project.ipynb - Colab									
	4G	69	116	0.982	0.983	Box(P R	mAP50 mAP50-95): 100% [██████████] 3/3 [00:01<00:00,	0.989	0.7
96/100	3.87G Class	box_loss Images	cls_loss Instances	dfl_loss Box(P	Instances 14	Size 640: 100% [██████████] 18/18 [00:06<00:00, 2.99it/	mAP50 mAP50-95): 100% [██████████] 3/3 [00:00<00:00,		
97/100	4G Class	box_loss Images	cls_loss Instances	dfl_loss Box(P	Instances 17	Size 640: 100% [██████████] 18/18 [00:05<00:00, 3.47it/	mAP50 mAP50-95): 100% [██████████] 3/3 [00:00<00:00,		
98/100	3.87G Class	box_loss Images	cls_loss Instances	dfl_loss Box(P	Instances 18	Size 640: 100% [██████████] 18/18 [00:07<00:00, 2.50it/	mAP50 mAP50-95): 100% [██████████] 3/3 [00:00<00:00,		
99/100	4G Class	box_loss Images	cls_loss Instances	dfl_loss Box(P	Instances 18	Size 640: 100% [██████████] 18/18 [00:05<00:00, 3.54it/	mAP50 mAP50-95): 100% [██████████] 3/3 [00:00<00:00,		
100/100	3.87G Class	box_loss Images	cls_loss Instances	dfl_loss Box(P	Instances 24	Size 640: 100% [██████████] 18/18 [00:06<00:00, 2.69it/	mAP50 mAP50-95): 100% [██████████] 3/3 [00:01<00:00,		

100 epochs completed in 0.224 hours.

Optimizer stripped from runs/detect/car\_detection\_model/weights/last.pt, 22.5MB

Optimizer stripped from runs/detect/car\_detection\_model/weights/best.pt, 22.5MB

Validating runs/detect/car\_detection\_model/weights/best.pt...

Ultralytics 8.3.48 🚀 Python-3.10.12 torch-2.5.1+cu121 CUDA:0 (Tesla T4, 15102MiB)

Model summary (fused): 168 layers, 11,125,971 parameters, 0 gradients, 28.4 GFLOPs

Class Images Instances Box(P R mAP50 mAP50-95): 100% [██████████] 3/3 [00:00<00:00,

all 69 116 0.974 0.983 0.984 0.71

Speed: 1.2ms preprocess, 3.3ms inference, 0.0ms loss, 1.6ms postprocess per image

Results saved to **runs/detect/car detection model**

```
# Printing the results from trained YOLO model
```

```
results = model.val()
print(results)
```



```

    ۰.۰۰۲۷/۴,    ۰.۰۰۲۷/۴,    ۰.۰۰۲۷/۴,    ۰.۰۰۲۷/۴,    ۰.۰۰۲۷/۴,    ۰.۰۰۲۷/۴,    ۰.۰۰۲۷/۴,
  ۰.۶۲۰۰۹,     ۰.۶۱۶۷۹,     ۰.۶۱۳۵,     ۰.۶۰۷۰۵,     ۰.۵۹۹۶۵,     ۰.۵۹۰۷۶,     ۰.۵۸۰۵۵,     ۰.۵۶۷۸۵,     ۰.۵۶۴۷۲,
  ۰.۴۸۷۰۶,     ۰.۴۷۴۴۲,     ۰.۴۷۱۴۵,     ۰.۴۶۸۷۲,     ۰.۴۶۵۹۹,     ۰.۴۵۳۴۷,     ۰.۴۴۴۶,     ۰.۴۳۹۷۷,     ۰.۴۱۶۰۸,
  ۰.۲۱۷۴,      ۰.۲۰۲۶۲,     ۰.۱۸۴۶۹,     ۰.۱۷۰۱۶,     ۰.۱۴۵۰۷,     ۰.۱۳۵۸۶,     ۰.۱۳۲۵۷,     ۰.۱۲۹۳۱,     ۰.۱۲۷۹,
  ۰.۰۴۱۴۹۹,    ۰.۰۴۰۸۲۹,    ۰.۰۴۰۱۶,    ۰.۰۳۹۴۹۱,    ۰.۰۳۸۸۲۱,    ۰.۰۳۸۱۵۲,    ۰.۰۳۷۴۸۳,    ۰.۰۳۶۸۱۳,    ۰.۰۳۶۱۴۴,
  ۰.۰۱۴۴۳۳,   ۰.۰۰۹۵۰۷۱,   ۰,          ۰,          ۰,          ۰,          ۰,          ۰,          ۰,          ۰,
  ۰,          ۰,          ۰,          ۰,          ۰,          ۰,          ۰,          ۰,          ۰,          ۰,
  ۰,          ۰,          ۰,          ۰,          ۰,          ۰,          ۰,          ۰,          ۰,          ۰,
  ۰,          ۰,          ۰,          ۰,          ۰,          ۰,          ۰,          ۰,          ۰,          ۰,
  ۰,          ۰,          ۰,          ۰,          ۰,          ۰,          ۰,          ۰,          ۰,          ۰,
fitness: 0.7360482138488454
keys: ['metrics/precision(B)', 'metrics/recall(B)', 'metrics/mAP50(B)', 'metrics/mAP50-95(B)']
maps: array([ 0.70849])
names: {0: 'car'}
plot: True
results_dict: {'metrics/precision(B)': 0.9741076874610393, 'metrics/recall(B)': 0.9827586206896551, 'metrics/mAP50(B)': 0.98
save_dir: PosixPath('runs/detect/car_detection_model2')
speed: {'preprocess': 0.3206488014995188, 'inference': 10.996777078379756, 'loss': 0.0010884326437245245, 'postprocess': 5.9
task: 'detect'

```

```
# mean of 50, 70, 95% precisions for the trained model
```

```

print(f"Mean Average Precision @.95 : {results.box.map}")
print(f"Mean Average Precision @ .50 : {results.box.map50}")
print(f"Mean Average Precision @ .70 : {results.box.map75}")

```

```
↳ Mean Average Precision @.95 : 0.7084891262125815
Mean Average Precision @ .50 : 0.9840800025752203
Mean Average Precision @ .70 : 0.9059462480927479
```

```
#Use of test images to predict the car objects using the trained YOLO model
```

```

image_dir = "/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images"

all_images = os.listdir(image_dir)
selected_images = all_images[:35]

for img_name in selected_images:
    img_path = os.path.join(image_dir, img_name)
    results = model.predict(img_path)
    img = cv2.imread(img_path)
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    for result in results:
        plotted_img = result.plot()
        plt.figure(figsize=(8, 6))
        plt.imshow(plotted_img)

        plt.axis('off')
        plt.show()

```



image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_29560.jpg: 38  
Speed: 3.2ms preprocess, 43.3ms inference, 1.4ms postprocess per image at shape (1, 3, 384, 640)



image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_28620.jpg: 38  
Speed: 3.0ms preprocess, 10.7ms inference, 0.8ms postprocess per image at shape (1, 3, 384, 640)



image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_29440.jpg: 38  
Speed: 3.5ms preprocess, 10.6ms inference, 2.1ms postprocess per image at shape (1, 3, 384, 640)



image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_30120.jpg: 38  
Speed: 2.6ms preprocess, 10.7ms inference, 0.7ms postprocess per image at shape (1, 3, 384, 640)





image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_29540.jpg: 38  
Speed: 2.6ms preprocess, 10.6ms inference, 1.7ms postprocess per image at shape (1, 3, 384, 640)



image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_28020.jpg: 38  
Speed: 4.8ms preprocess, 14.0ms inference, 0.8ms postprocess per image at shape (1, 3, 384, 640)

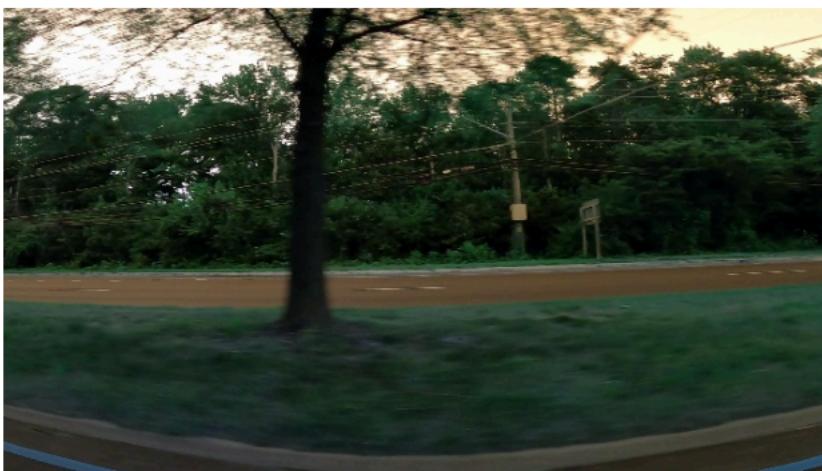


image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_27260.jpg: 38  
Speed: 3.4ms preprocess, 10.7ms inference, 1.7ms postprocess per image at shape (1, 3, 384, 640)

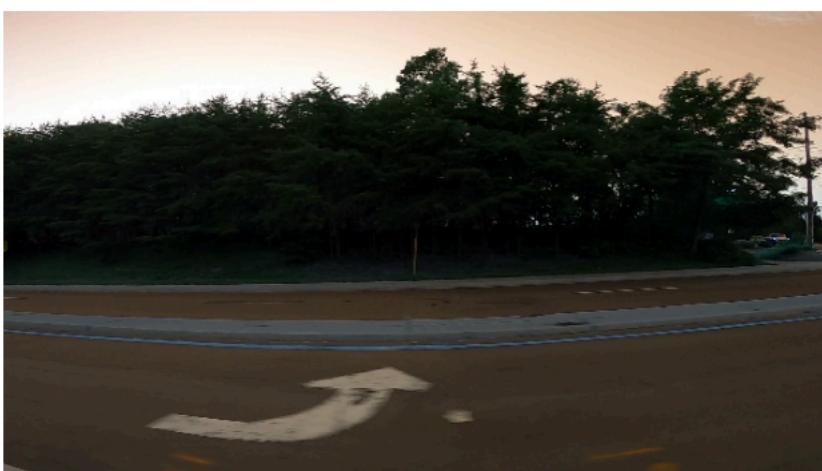


image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_28560.jpg: 38  
Speed: 2.8ms preprocess, 10.6ms inference, 0.7ms postprocess per image at shape (1, 3, 384, 640)

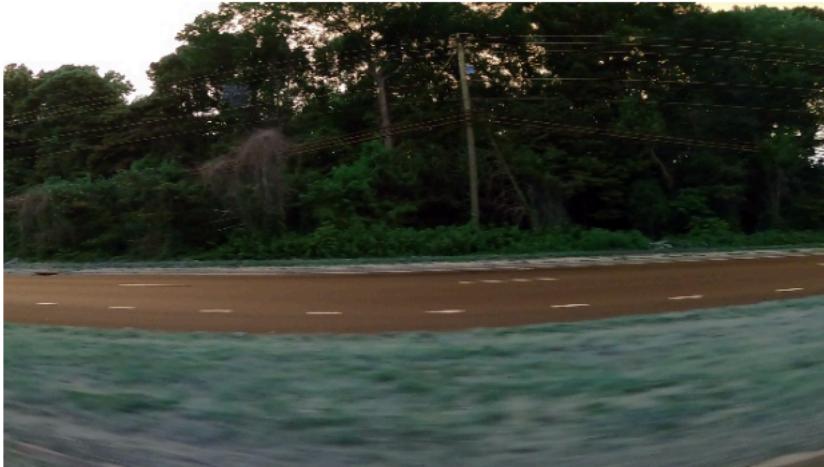


image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_29600.jpg: 38  
Speed: 2.7ms preprocess, 15.5ms inference, 0.9ms postprocess per image at shape (1, 3, 384, 640)



image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_27400.jpg: 38  
Speed: 2.7ms preprocess, 15.0ms inference, 1.7ms postprocess per image at shape (1, 3, 384, 640)



image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_30160.jpg: 38  
Speed: 2.7ms preprocess, 12.1ms inference, 0.8ms postprocess per image at shape (1, 3, 384, 640)





image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_26620.jpg: 38  
Speed: 2.8ms preprocess, 15.6ms inference, 1.7ms postprocess per image at shape (1, 3, 384, 640)

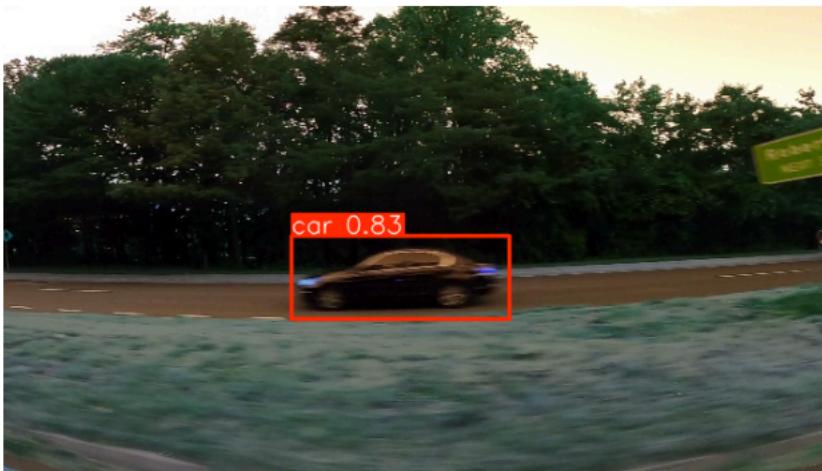


image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_27760.jpg: 38  
Speed: 2.6ms preprocess, 14.1ms inference, 0.7ms postprocess per image at shape (1, 3, 384, 640)

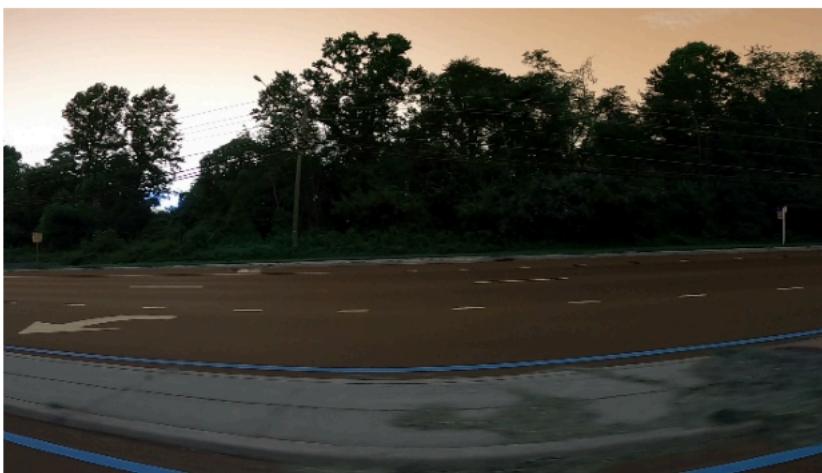


image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_29020.jpg: 38  
Speed: 2.7ms preprocess, 14.0ms inference, 1.7ms postprocess per image at shape (1, 3, 384, 640)



image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_29460.jpg: 38  
Speed: 2.9ms preprocess, 16.2ms inference, 1.6ms postprocess per image at shape (1, 3, 384, 640)



image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_27580.jpg: 38  
Speed: 3.2ms preprocess, 12.9ms inference, 0.9ms postprocess per image at shape (1, 3, 384, 640)

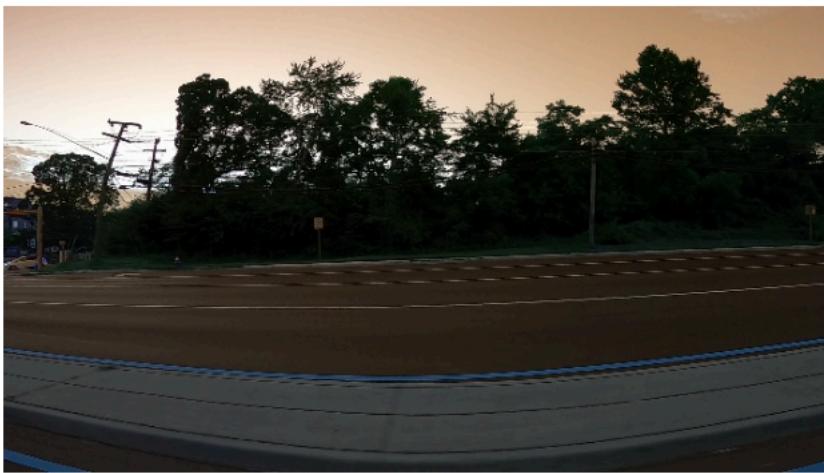


image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_26840.jpg: 38  
Speed: 2.7ms preprocess, 13.5ms inference, 1.9ms postprocess per image at shape (1, 3, 384, 640)

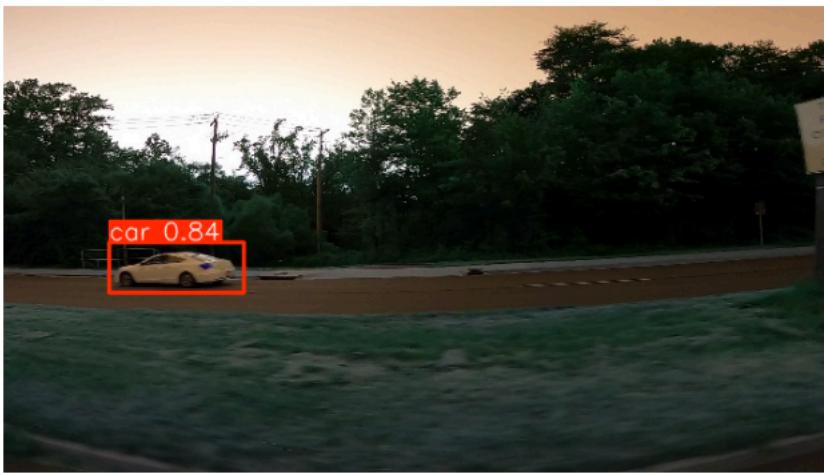


image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_27900.jpg: 38  
Speed: 2.7ms preprocess, 16.2ms inference, 1.7ms postprocess per image at shape (1, 3, 384, 640)

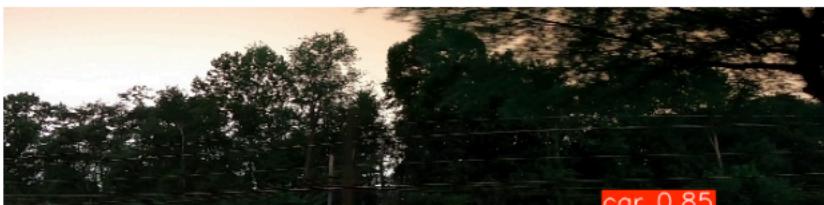




image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_27840.jpg: 38  
Speed: 2.9ms preprocess, 16.7ms inference, 2.0ms postprocess per image at shape (1, 3, 384, 640)



image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_27300.jpg: 38  
Speed: 2.7ms preprocess, 15.9ms inference, 1.8ms postprocess per image at shape (1, 3, 384, 640)

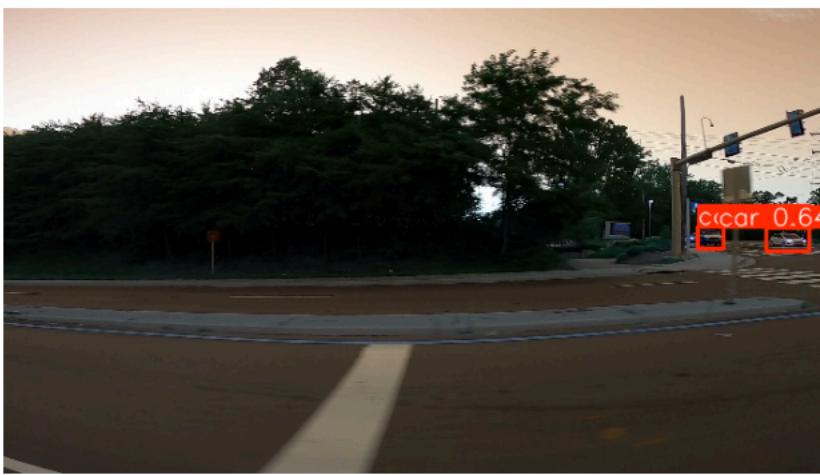


image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_29040.jpg: 38  
Speed: 2.7ms preprocess, 13.5ms inference, 5.9ms postprocess per image at shape (1, 3, 384, 640)



image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_26940.jpg: 38  
Speed: 2.6ms preprocess, 14.7ms inference, 1.9ms postprocess per image at shape (1, 3, 384, 640)

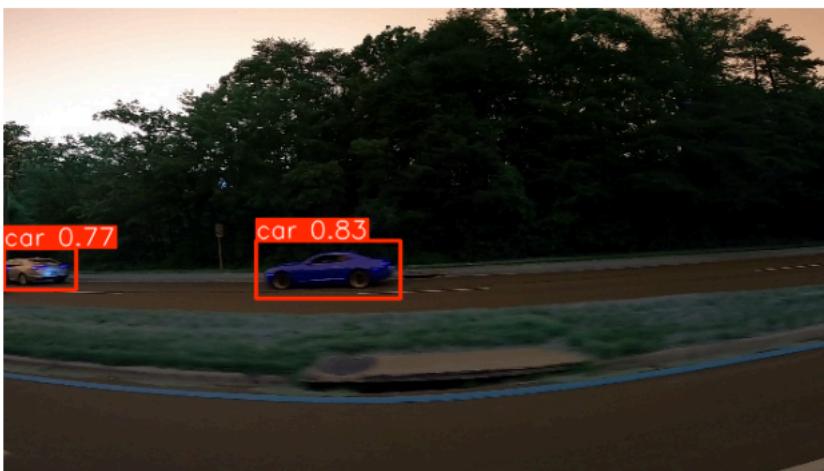


image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_26880.jpg: 38  
Speed: 2.7ms preprocess, 15.2ms inference, 2.3ms postprocess per image at shape (1, 3, 384, 640)

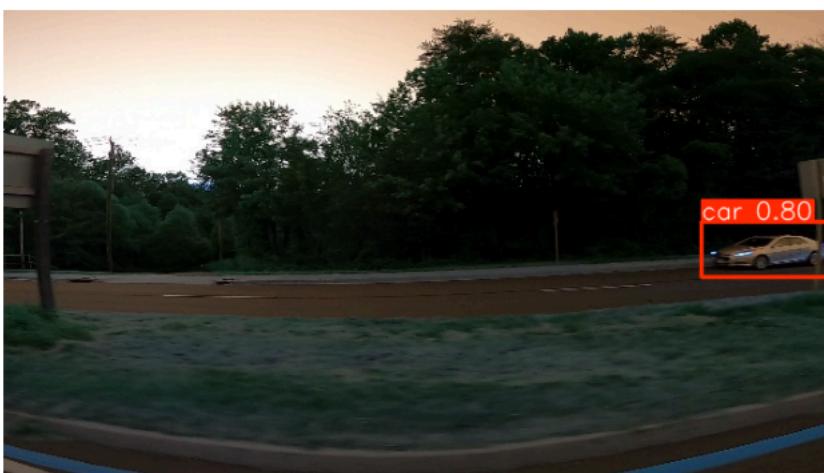


image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_27360.jpg: 38  
Speed: 2.5ms preprocess, 13.1ms inference, 5.8ms postprocess per image at shape (1, 3, 384, 640)

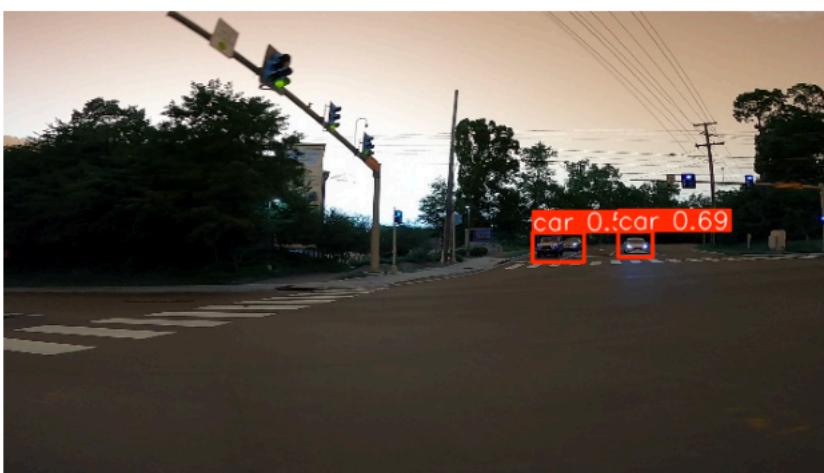


image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_27420.jpg: 38  
Speed: 2.7ms preprocess, 12.6ms inference, 1.8ms postprocess per image at shape (1, 3, 384, 640)





image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_27440.jpg: 38  
Speed: 2.7ms preprocess, 11.5ms inference, 2.3ms postprocess per image at shape (1, 3, 384, 640)



image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_25200.jpg: 38  
Speed: 2.7ms preprocess, 13.2ms inference, 0.8ms postprocess per image at shape (1, 3, 384, 640)

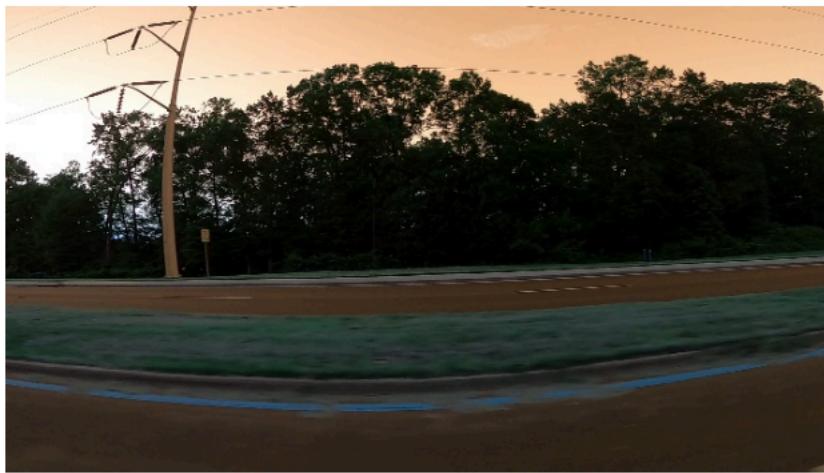


image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_29620.jpg: 38  
Speed: 2.7ms preprocess, 12.5ms inference, 0.9ms postprocess per image at shape (1, 3, 384, 640)



image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_29820.jpg: 38  
Speed: 2.7ms preprocess, 17.1ms inference, 1.7ms postprocess per image at shape (1, 3, 384, 640)

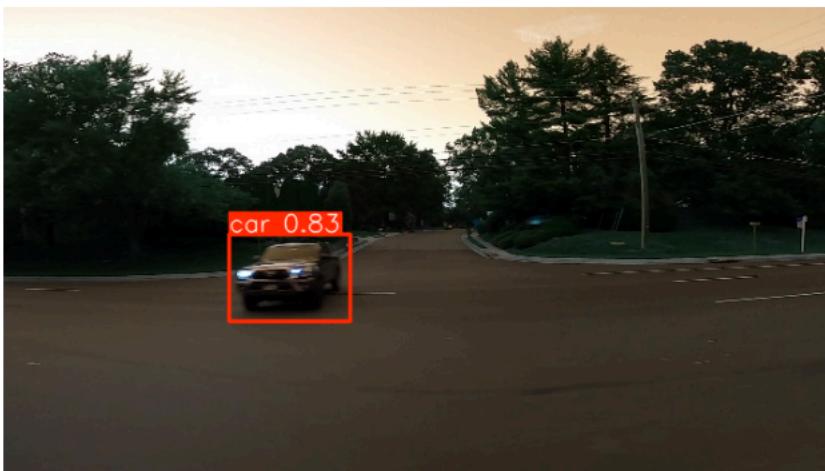


image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_29580.jpg: 38  
Speed: 2.6ms preprocess, 12.4ms inference, 1.9ms postprocess per image at shape (1, 3, 384, 640)



image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_30740.jpg: 38  
Speed: 2.7ms preprocess, 12.4ms inference, 4.2ms postprocess per image at shape (1, 3, 384, 640)

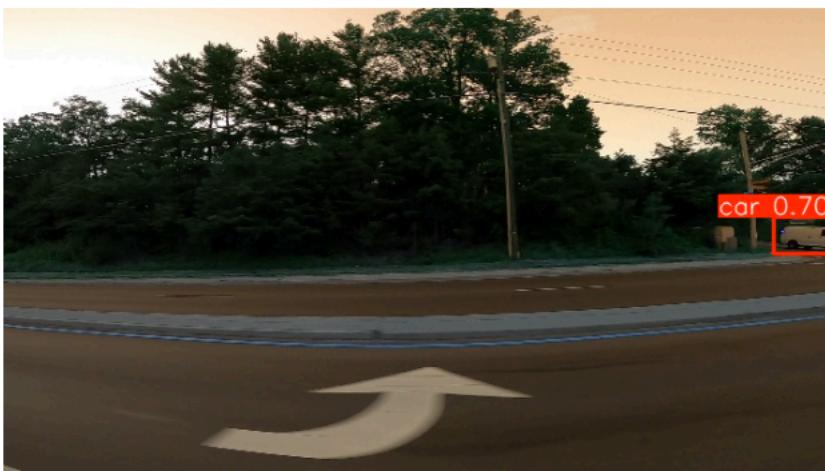


image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_27500.jpg: 38  
Speed: 2.6ms preprocess, 10.8ms inference, 1.8ms postprocess per image at shape (1, 3, 384, 640)





image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_26660.jpg: 38  
Speed: 2.7ms preprocess, 12.3ms inference, 1.8ms postprocess per image at shape (1, 3, 384, 640)

