

```
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 [00:00<00:00, 129MB/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)

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_29540.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_30040.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_26640.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_29060.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_28000.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_31620.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_27900.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_26980.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_28420.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_31680.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_30860.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_28380.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_31300.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_27380.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_28180.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_27620.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_26400.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_26700.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_31720.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_27940.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_31640.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_27600.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_31260.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_29760.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_26720.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_27660.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_29860.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_26680.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_27740.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_25100.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_28020.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_27640.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_27320.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_420.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_28660.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_26780.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_25240.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_28260.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_28680.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_26760.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_31140.jpg
/root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing_images/vid_5_30140.jpg
```

```
!pip install ultralytics
```

→ Show hidden output

```
!pip install -U ipywidgets
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
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

```
!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"
```

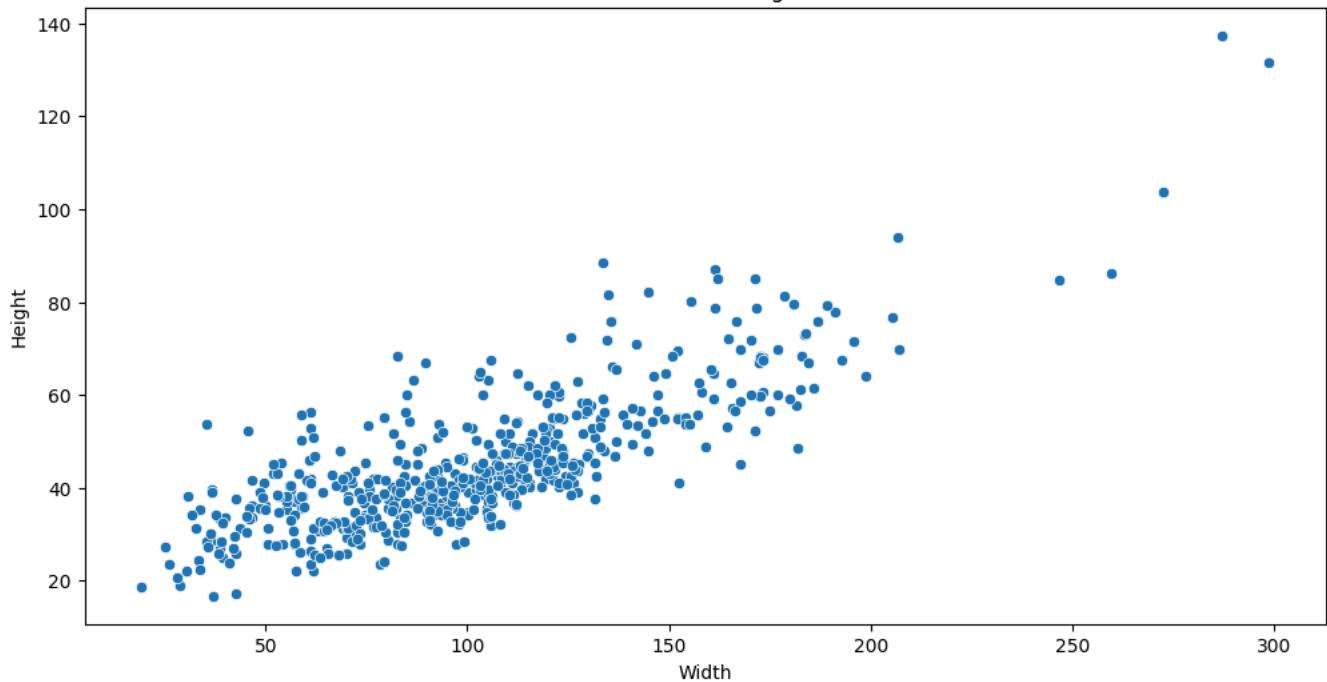
```
df = pd.read_csv(csv_data)
df.head()
```

	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

```
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



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

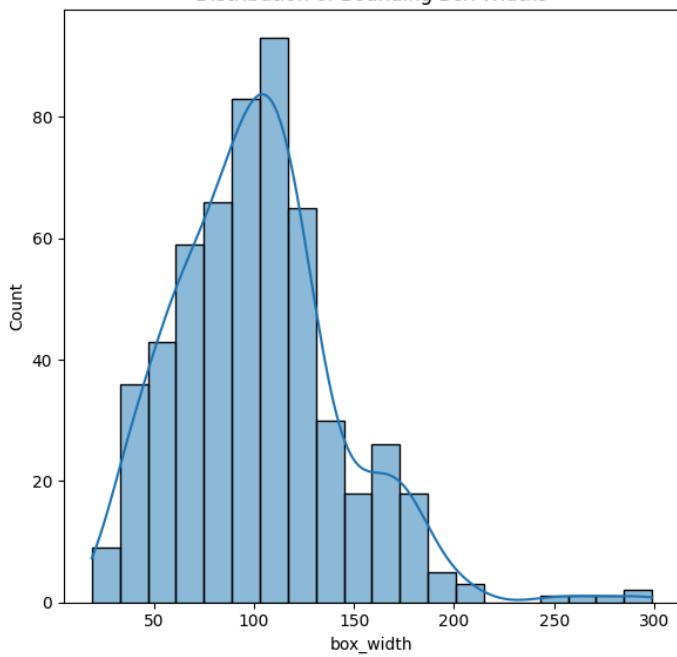
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
sns.histplot(df['box_width'], bins=20, kde=True)
plt.title('Distribution of Bounding Box Widths')

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

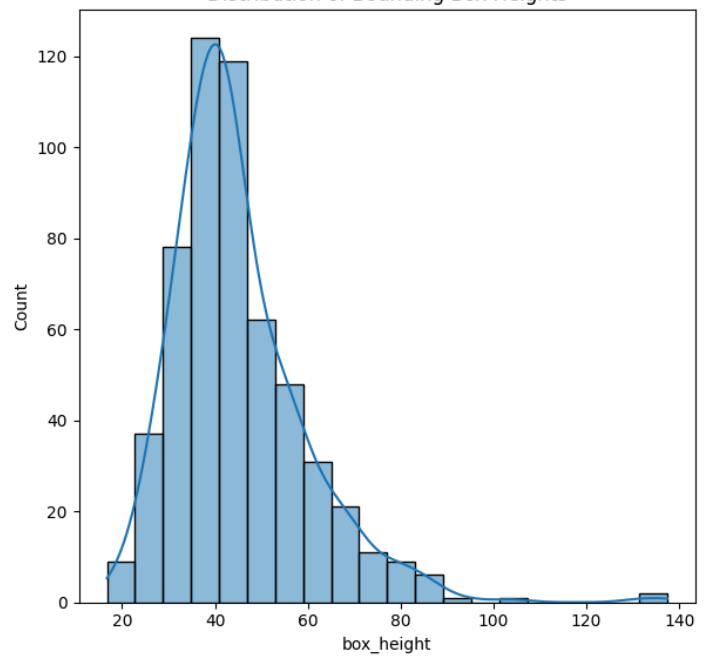
plt.tight_layout();
plt.show();
```



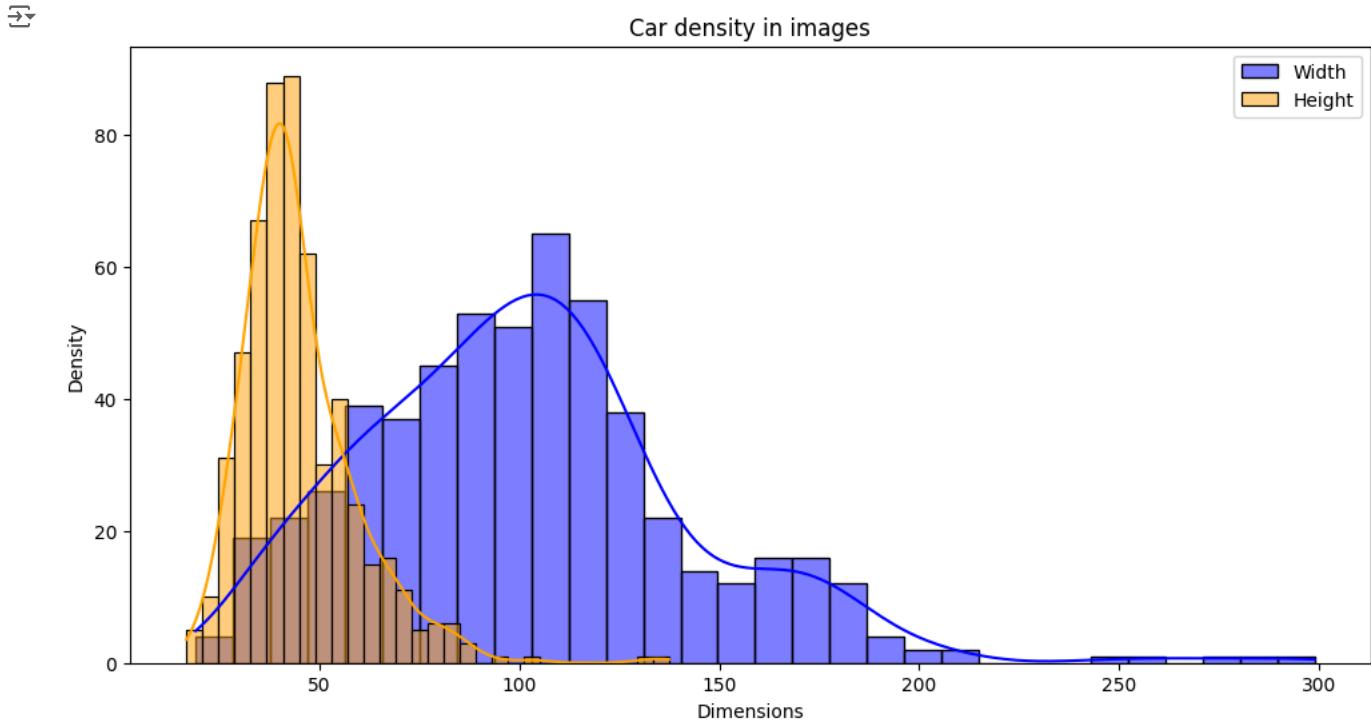
Distribution of Bounding Box Widths



Distribution of Bounding Box Heights



```
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()
```



```
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\_11280.jpg



vid\_4\_23140.jpg



vid\_4\_620.jpg



vid\_4\_5760.jpg

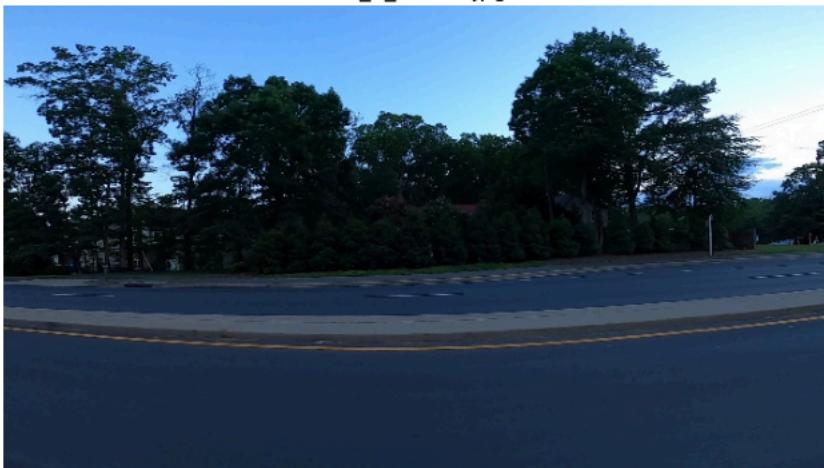




vid\_4\_14460.jpg



vid\_4\_19680.jpg



vid\_4\_11060.jpg



vid\_4\_3440.jpg

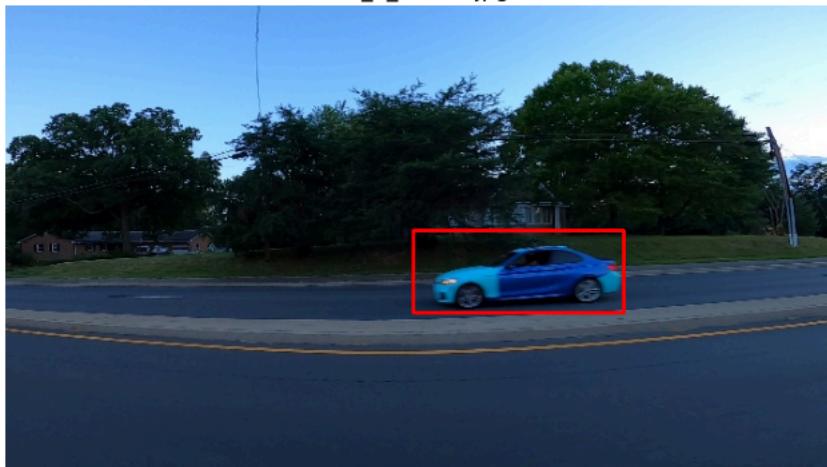




vid\_4\_20940.jpg



vid\_4\_18180.jpg



vid\_4\_21320.jpg



vid\_4\_3520.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_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.

```

```

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, 106MB/s]

```

```

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

```



epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size	
	0G	1.012	0.4767	0.9982	16	640: 100% [██████████]	18/18 [11:54<00:00, 39.69s/i]
	Class	Images	Instances	Box(P)	R	mAP50	mAP50-95): 100% [██████████] 3/3 [00:31<00:00,
Epoch 26/33	GPU_mem 0G	box_loss 1.059	cls_loss 0.4924	dfl_loss 1.031	Instances 20	Size 640: 100% [██████████]	18/18 [11:49<00:00, 39.39s/i]
Epoch 27/33	GPU_mem 0G	box_loss 1.029	cls_loss 0.4737	dfl_loss 1.018	Instances 19	Size 640: 100% [██████████]	18/18 [11:44<00:00, 39.13s/i]
Epoch 28/33	GPU_mem 0G	box_loss 1.028	cls_loss 0.4733	dfl_loss 1.02	Instances 21	Size 640: 100% [██████████]	18/18 [11:48<00:00, 39.36s/i]
Epoch 29/33	GPU_mem 0G	box_loss 0.979	cls_loss 0.4603	dfl_loss 1.009	Instances 21	Size 640: 100% [██████████]	18/18 [11:49<00:00, 39.41s/i]
Epoch 30/33	GPU_mem 0G	box_loss 0.9757	cls_loss 0.453	dfl_loss 1.008	Instances 17	Size 640: 100% [██████████]	18/18 [11:40<00:00, 38.90s/i]
Epoch 31/33	GPU_mem 0G	box_loss 0.9534	cls_loss 0.4446	dfl_loss 0.9875	Instances 18	Size 640: 100% [██████████]	18/18 [11:35<00:00, 38.64s/i]
Epoch 32/33	GPU_mem 0G	box_loss 0.9425	cls_loss 0.4234	dfl_loss 0.981	Instances 22	Size 640: 39% [████]	7/18 [04:36<07:14, 39.50s/it]

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

→ Ultralytics 8.3.39 🚀 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  
**val:** Scanning /content/working/data/labels/val.cache... 67 images, 0 backgrounds, 0 corrupt: 100% [██████████] 67/67 [00:00<?  
Class Images Instances Box(P) R mAP50 mAP50-95): 100% [██████████] 5/5 [00:02<00:00,  
all 67 105 0.981 0.971 0.992 0.694  
Speed: 0.5ms preprocess, 12.1ms inference, 0.0ms loss, 5.8ms postprocess per image  
Results saved to runs/detect/car\_detection\_model2  
ultralytics.utils.metrics.DetMetrics object with attributes:

```
ap_class_index: array([0])
box: ultralytics.utils.metrics.Metric object
confusion_matrix: <ultralytics.utils.metrics.ConfusionMatrix object at 0x7a08ea3a2710>
curves: ['Precision-Recall(B)', 'F1-Confidence(B)', 'Precision-Confidence(B)', 'Recall-Confidence(B)']
curves_results: [[array([
    0, 0.001001, 0.002002, 0.003003, 0.004004, 0.005005, 0.006006, 0.007
    0.024024, 0.025025, 0.026026, 0.027027, 0.028028, 0.029029, 0.03003, 0.031031, 0.032032,
    0.048048, 0.049049, 0.05005, 0.051051, 0.052052, 0.053053, 0.054054, 0.055055, 0.056056,
    0.072072, 0.073073, 0.074074, 0.075075, 0.076076, 0.077077, 0.078078, 0.079079, 0.08008,
    0.096096, 0.097097, 0.098098, 0.099099, 0.10001, 0.1011, 0.1021, 0.1031, 0.1041,
    0.12012, 0.12112, 0.12212, 0.12312, 0.12412, 0.12513, 0.12613, 0.12713, 0.12813,
    0.14414, 0.14515, 0.14615, 0.14715, 0.14815, 0.14915, 0.15015, 0.15115, 0.15215,
    0.16817, 0.16917, 0.17017, 0.17117, 0.17217, 0.17317, 0.17417, 0.17518, 0.17618,
    0.19219, 0.19319, 0.19419, 0.1952, 0.1962, 0.1972, 0.1982, 0.1992, 0.2002,
    0.21622, 0.21722, 0.21822, 0.21922, 0.22022, 0.22122, 0.22222, 0.22322, 0.22422,
    0.24024, 0.24124, 0.24224, 0.24324, 0.24424, 0.24525, 0.24625, 0.24725, 0.24825,
    0.26426, 0.26527, 0.26627, 0.26727, 0.26827, 0.26927, 0.27027, 0.27127, 0.27227,
    0.28829, 0.28929, 0.29029, 0.29129, 0.29229, 0.29329, 0.29429, 0.29529, 0.29629,
    0.31231, 0.31331, 0.31431, 0.31532, 0.31632, 0.31732, 0.31832, 0.31932, 0.32032,
    0.33634, 0.33734, 0.33834, 0.33934, 0.34034, 0.34134, 0.34234, 0.34334, 0.34434,
    0.36036, 0.36136, 0.36236, 0.36336, 0.36436, 0.36537, 0.36637, 0.36737, 0.36837,
    0.38438, 0.38539, 0.38639, 0.38739, 0.38839, 0.38939, 0.39039, 0.39139, 0.39239,
    0.40841, 0.40941, 0.41041, 0.41141, 0.41241, 0.41341, 0.41441, 0.41542, 0.41642,
    0.43243, 0.43343, 0.43443, 0.43544, 0.43644, 0.43744, 0.43844, 0.43944, 0.44044,
    0.45646, 0.45746, 0.45846, 0.45946, 0.46046, 0.46146, 0.46246, 0.46346, 0.46446,
    0.48048, 0.48148, 0.48248, 0.48348, 0.48448, 0.48549, 0.48649, 0.48749, 0.48849,
    0.5045, 0.50551, 0.50651, 0.50751, 0.50851, 0.50951, 0.51051, 0.51151, 0.51251,
    0.52853, 0.52953, 0.53053, 0.53153, 0.53253, 0.53353, 0.53453, 0.53554, 0.53654,
    0.55255, 0.55355, 0.55455, 0.55556, 0.55656, 0.55756, 0.55856, 0.55956, 0.56056,
    0.57658, 0.57758, 0.57858, 0.57958, 0.58058, 0.58158, 0.58258, 0.58358, 0.58458,
    0.6006, 0.6016, 0.6026, 0.6036, 0.6046, 0.60561, 0.60661, 0.60761, 0.60861,
    0.62462, 0.62563, 0.62663, 0.62763, 0.62863, 0.62963, 0.63063, 0.63163, 0.63263,
    0.64865, 0.64965, 0.65065, 0.65165, 0.65265, 0.65365, 0.65465, 0.65566, 0.65666,
    0.67267, 0.67367, 0.67467, 0.67568, 0.67668, 0.67768, 0.67868, 0.67968, 0.68068,
    0.6967, 0.6977, 0.6987, 0.6997, 0.7007, 0.7017, 0.7027, 0.7037, 0.7047,
```

0.72072,	0.72172,	0.72272,	0.72372,	0.72472,	0.72573,	0.72673,	0.72773,	0.72873,
0.74474,	0.74575,	0.74675,	0.74775,	0.74875,	0.74975,	0.75075,	0.75175,	0.75275,
0.76877,	0.76977,	0.77077,	0.77177,	0.77277,	0.77377,	0.77477,	0.77578,	0.77678,
0.79279,	0.79379,	0.79479,	0.7958,	0.7968,	0.7978,	0.7988,	0.7998,	0.8008,
0.81682,	0.81782,	0.81882,	0.81982,	0.82082,	0.82182,	0.82282,	0.82382,	0.82482,
0.84084,	0.84184,	0.84284,	0.84384,	0.84484,	0.84585,	0.84685,	0.84785,	0.84885,
0.86486,	0.86587,	0.86687,	0.86787,	0.86887,	0.86987,	0.87087,	0.87187,	0.87287,
0.88889,	0.88989,	0.89089,	0.89189,	0.89289,	0.89389,	0.89489,	0.89589,	0.89689,
0.91291,	0.91391,	0.91491,	0.91592,	0.91692,	0.91792,	0.91892,	0.91992,	0.92092,
0.93694,	0.93794,	0.93894,	0.93994,	0.94094,	0.94194,	0.94294,	0.94394,	0.94494,
0.96096,	0.96196,	0.96296,	0.96396,	0.96496,	0.96597,	0.96697,	0.96797,	0.96897,
0.98498,	0.98599,	0.98699,	0.98799,	0.98899,	0.98999,	0.99099,	0.99199,	0.99299,
1,	1,	1,	1,	1,	1,	1,	1,	1,
1,	1,	1,	1,	1,	1,	1,	1,	1,
1	1	1	1	1	1	1	1	1

```
print(f"Mean Average Precision @.5:.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 @.5:.95 : 0.693798568417572  
 Mean Average Precision @ .50 : 0.992111260651252  
 Mean Average Precision @ .70 : 0.821616152076785

```
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: 4.4ms preprocess, 83.3ms 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\_29540.jpg: 38  
Speed: 2.7ms preprocess, 13.3ms 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\_27560.jpg: 38  
Speed: 2.7ms preprocess, 12.7ms inference, 1.9ms postprocess per image at shape (1, 3, 384, 640)

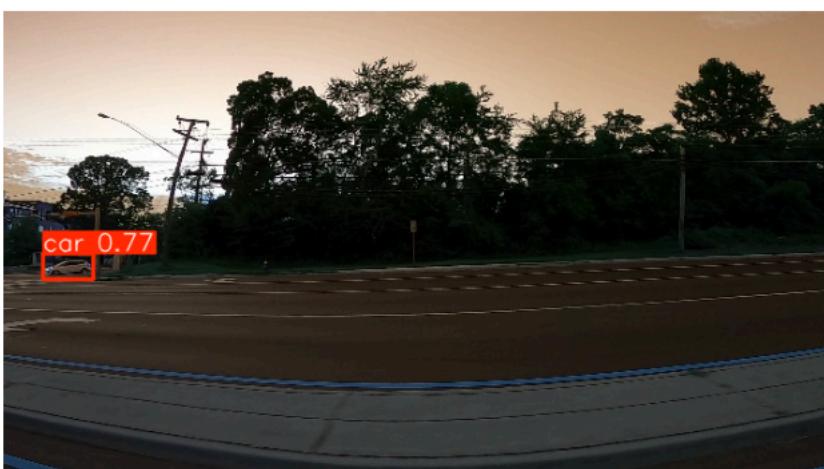


image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_30040.jpg: 38  
Speed: 2.6ms preprocess, 11.7ms inference, 0.8ms postprocess per image at shape (1, 3, 384, 640)





image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_26640.jpg: 38  
Speed: 7.6ms preprocess, 17.0ms 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\_29060.jpg: 38  
Speed: 2.6ms preprocess, 15.3ms inference, 4.8ms 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\_28600.jpg: 38  
Speed: 2.6ms 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\_31620.jpg: 38  
Speed: 2.6ms 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: 2.6ms preprocess, 13.1ms 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\_27900.jpg: 38  
Speed: 2.6ms preprocess, 13.0ms inference, 1.6ms postprocess per image at shape (1, 3, 384, 640)

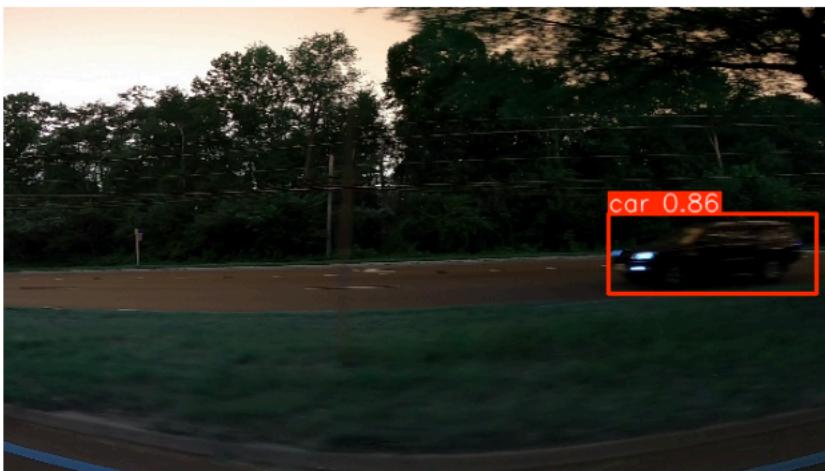


image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_26980.jpg: 38  
Speed: 2.7ms preprocess, 10.8ms inference, 1.0ms postprocess per image at shape (1, 3, 384, 640)





image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_28420.jpg: 38  
Speed: 2.6ms preprocess, 12.0ms 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\_31680.jpg: 38  
Speed: 2.5ms preprocess, 12.5ms inference, 0.7ms postprocess per image at shape (1, 3, 384, 640)

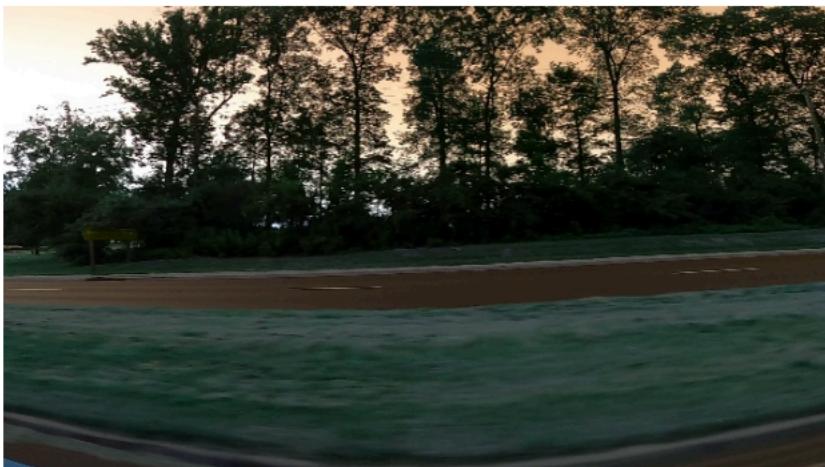


image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_29820.jpg: 38  
Speed: 2.6ms preprocess, 13.2ms inference, 1.8ms postprocess per image at shape (1, 3, 384, 640)



image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_30860.jpg: 38  
Speed: 2.6ms preprocess, 10.8ms inference, 1.8ms postprocess per image at shape (1, 3, 384, 640)

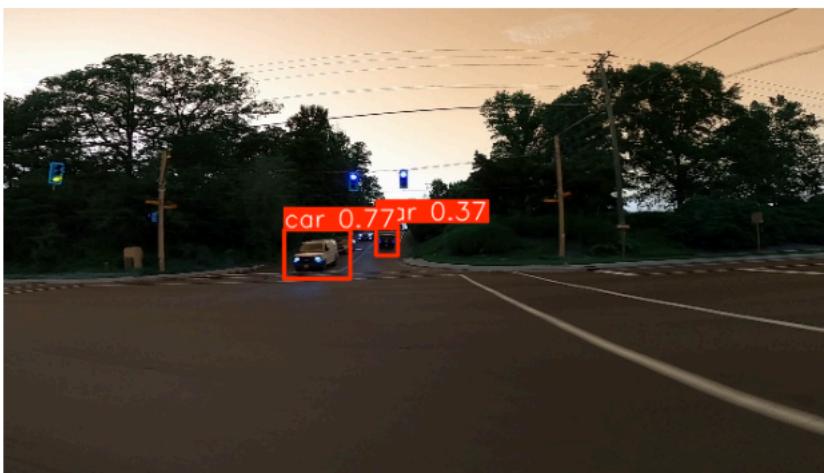


image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_28380.jpg: 38  
Speed: 2.7ms preprocess, 12.0ms inference, 4.9ms postprocess per image at shape (1, 3, 384, 640)



image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_31300.jpg: 38  
Speed: 5.6ms preprocess, 10.8ms 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\_27380.jpg: 38  
Speed: 2.8ms preprocess, 11.5ms 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\_28180.jpg: 38  
Speed: 3.6ms preprocess, 10.8ms 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\_27620.jpg: 38  
Speed: 2.7ms preprocess, 11.1ms inference, 1.7ms postprocess per image at shape (1, 3, 384, 640)

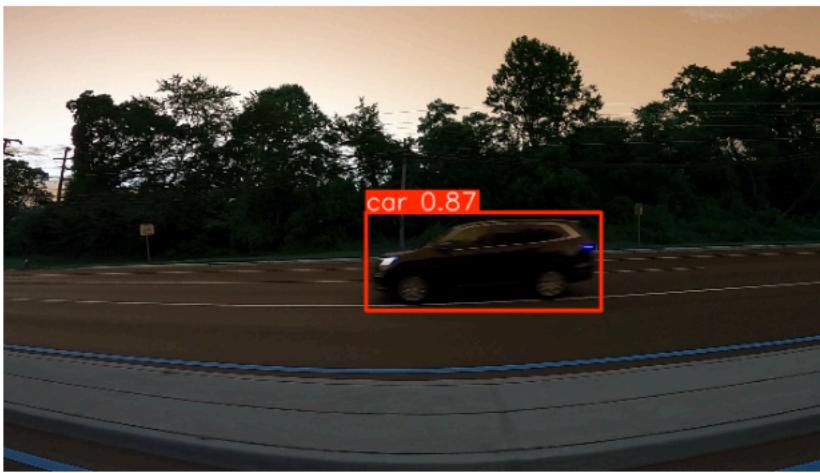


image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_27520.jpg: 38  
Speed: 2.6ms preprocess, 17.1ms inference, 1.9ms postprocess per image at shape (1, 3, 384, 640)

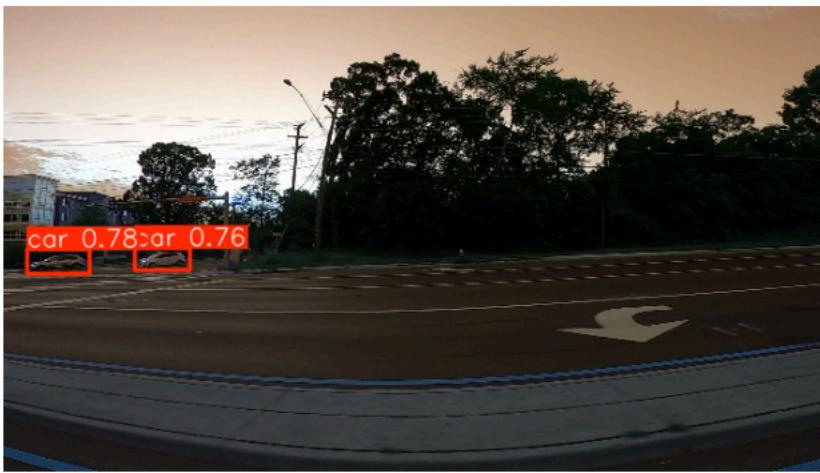


image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_26400.jpg: 38  
Speed: 2.7ms preprocess, 15.1ms inference, 0.8ms postprocess per image at shape (1, 3, 384, 640)



image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_26700.jpg: 38  
Speed: 2.7ms preprocess, 11.0ms 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\_31500.jpg: 38  
Speed: 2.6ms preprocess, 16.6ms inference, 0.8ms postprocess per image at shape (1, 3, 384, 640)

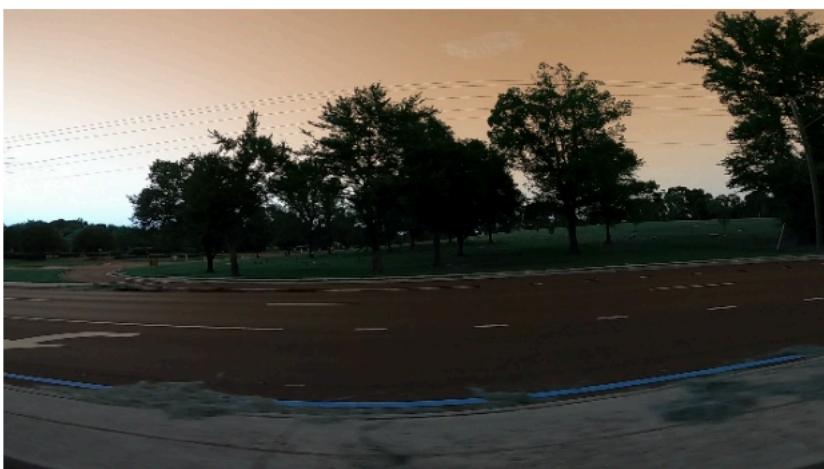


image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_31720.jpg: 38  
Speed: 6.4ms preprocess, 10.8ms inference, 1.6ms postprocess per image at shape (1, 3, 384, 640)

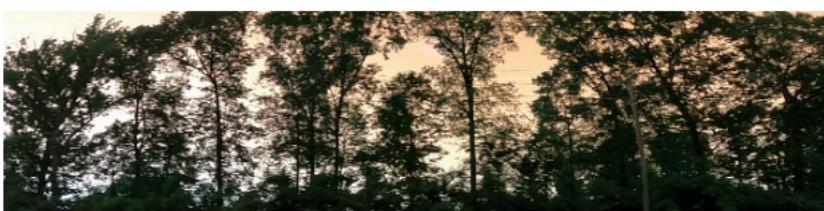




image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_27940.jpg: 38  
Speed: 2.6ms preprocess, 11.2ms inference, 1.7ms postprocess per image at shape (1, 3, 384, 640)



image 1/1 /root/.cache/kagglehub/datasets/sshikamaru/car-object-detection/versions/2/data/testing\_images/vid\_5\_29740.jpg: 38  
Speed: 2.7ms preprocess, 11.1ms 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\_31640.jpg: 38  
Speed: 2.8ms preprocess, 10.8ms inference, 1.2ms 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\_29520.jpg: 38  
Speed: 4.7ms preprocess, 10.8ms inference, 1.1ms 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\_27600.jpg: 38  
Speed: 2.9ms preprocess, 12.3ms inference, 0.9ms postprocess per image at shape (1, 3, 384, 640)

