

Test Predicting YOLOV8 On Car Driving Dataset

```
In [ ]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from glob import glob

import cv2 # pip install opencv-python
from IPython.display import Video # pip install ipython
import ultralytics
from ultralytics import YOLO # pip install -U ultralytics

import IPython.display as ipd
from tqdm import tqdm

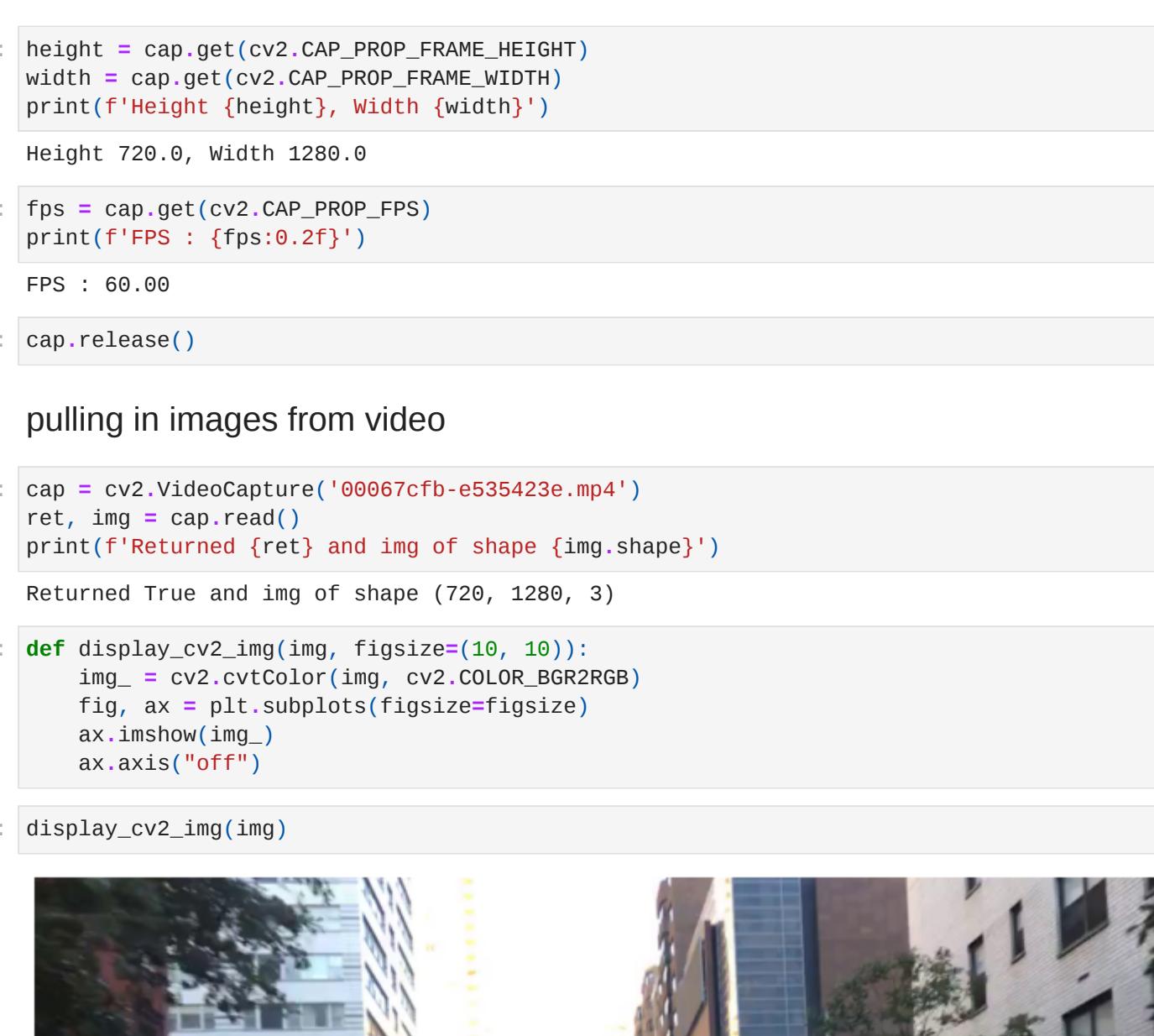
import subprocess
plt.style.use('ggplot')

In [ ]: ultralytics.checks()
```

Ultronics YOLOv8.0.110 Python-3.10.11 torch-2.0.1+cpu CPU
Setup complete (8 CPUs, 7.7 GB RAM, 222.4/475.8 GB disk)

Show the video

```
In [ ]: ipd.Video('00067cfb-e535423e.mp4', embed=True, width=700)
```



```
In [ ]: model = YOLO('yolov8x.pt')
```

```
In [ ]: cap = cv2.VideoCapture('00067cfb-e535423e.mp4')
```

```
In [ ]: cap.get(cv2.CAP_PROP_FRAME_COUNT)
```

```
Out[ ]: 2405.0
```

```
In [ ]: height = cap.get(cv2.CAP_PROP_FRAME_HEIGHT)
width = cap.get(cv2.CAP_PROP_FRAME_WIDTH)
print(f'Height {height}, Width {width}')
```

Height 720.0, Width 1280.0

```
In [ ]: fps = cap.get(cv2.CAP_PROP_FPS)
print(f'FPS : {fps:.2f}')
```

FPS : 60.00

```
In [ ]: cap.release()
```

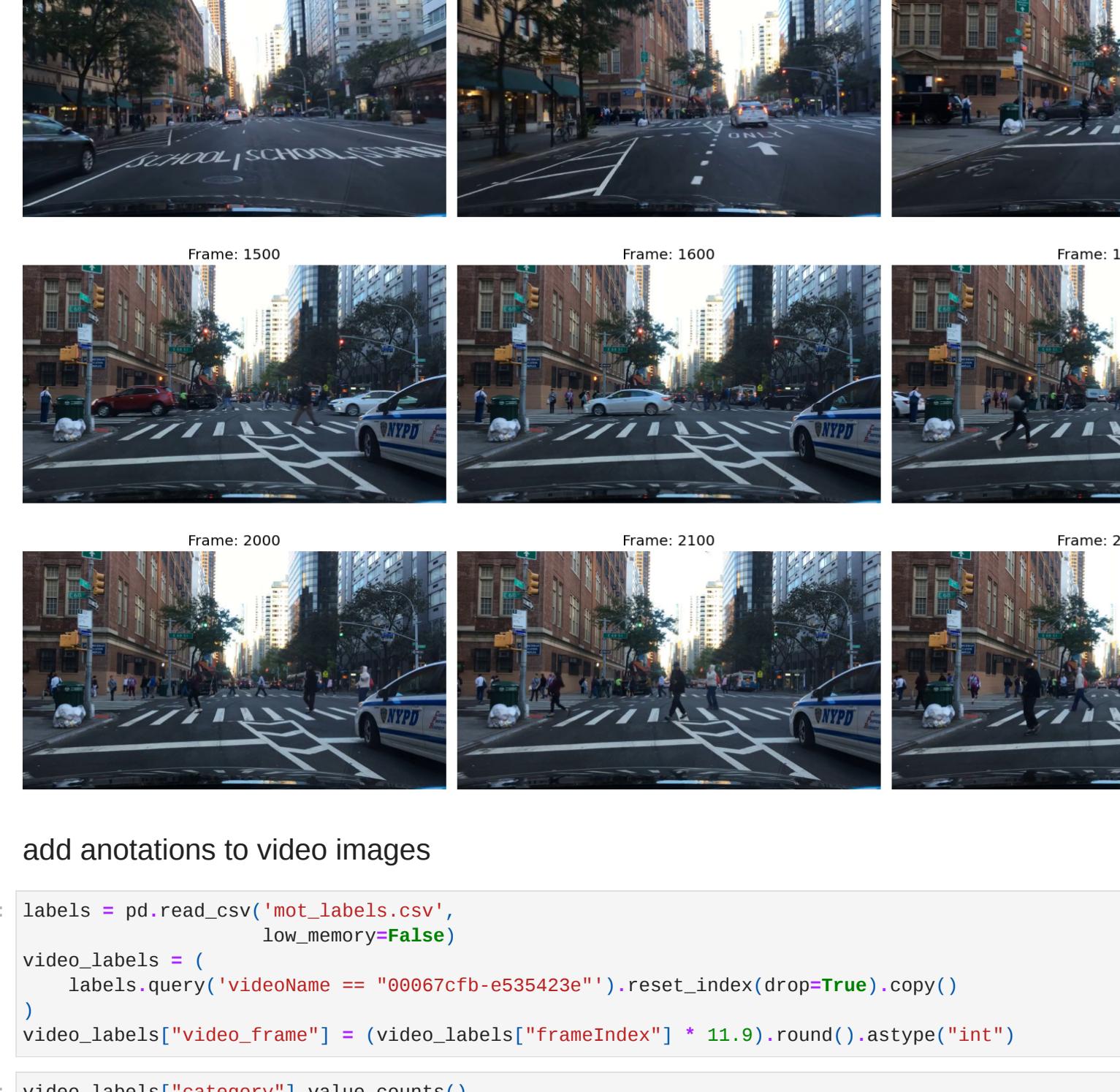
Pulling in images from video

```
In [ ]: cap = cv2.VideoCapture('00067cfb-e535423e.mp4')
ret, img = cap.read()
print(f'Returned {ret} and img of shape {img.shape}')
```

Returned True and img of shape (720, 1280, 3)

```
In [ ]: def display_cv2_img(img, figsize=(10, 10)):
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    fig, ax = plt.subplots(figsize=figsize)
    ax.imshow(img)
    ax.axis('off')
```

```
In [ ]: display_cv2_img(img)
```



```
In [ ]: cap.release()
```

Display multiple frames from the video

```
In [ ]: fig, axs = plt.subplots(5, 5, figsize=(30, 20))
axs = axs.flatten()
```

```
cap = cv2.VideoCapture('00067cfb-e535423e.mp4')
n_frames = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
```

```
img_idx = 0
for frame in range(n_frames):
    ret, img = cap.read()
    if ret == False:
        break
    if img_idx == 0:
        fig[0, 0].imshow(img)
        fig[0, 0].set_title(f'Frame: {frame}')
    axs[img_idx].axis('off')
    img_idx += 1
```

```
plt.tight_layout()
plt.show()
cap.release()
```



Add annotations to video images

```
In [ ]: labels = pd.read_csv('mot_labels.csv',
                           low_memory=False)
video_labels = (
    labels.query("videoName == '00067cfb-e535423e'").reset_index(drop=True).copy())
video_labels['video_frame'] = (video_labels['frameIndex'] * 11.9).round().astype("int")
```

```
In [ ]: video_labels['category'].value_counts()
```

```
Out[ ]: pedestrian      2347
car                   1245
bus                   124
truck                  124
other vehicle       122
bicycle                 18
rider                  18
Name: category, dtype: int64
```

```
In [ ]: cap = cv2.VideoCapture('00067cfb-e535423e.mp4')
n_frames = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
```

```
img_idx = 0
for frame in range(n_frames):
    ret, img = cap.read()
    if ret == False:
        break
    if frame == 996:
        break
cap.release()
```

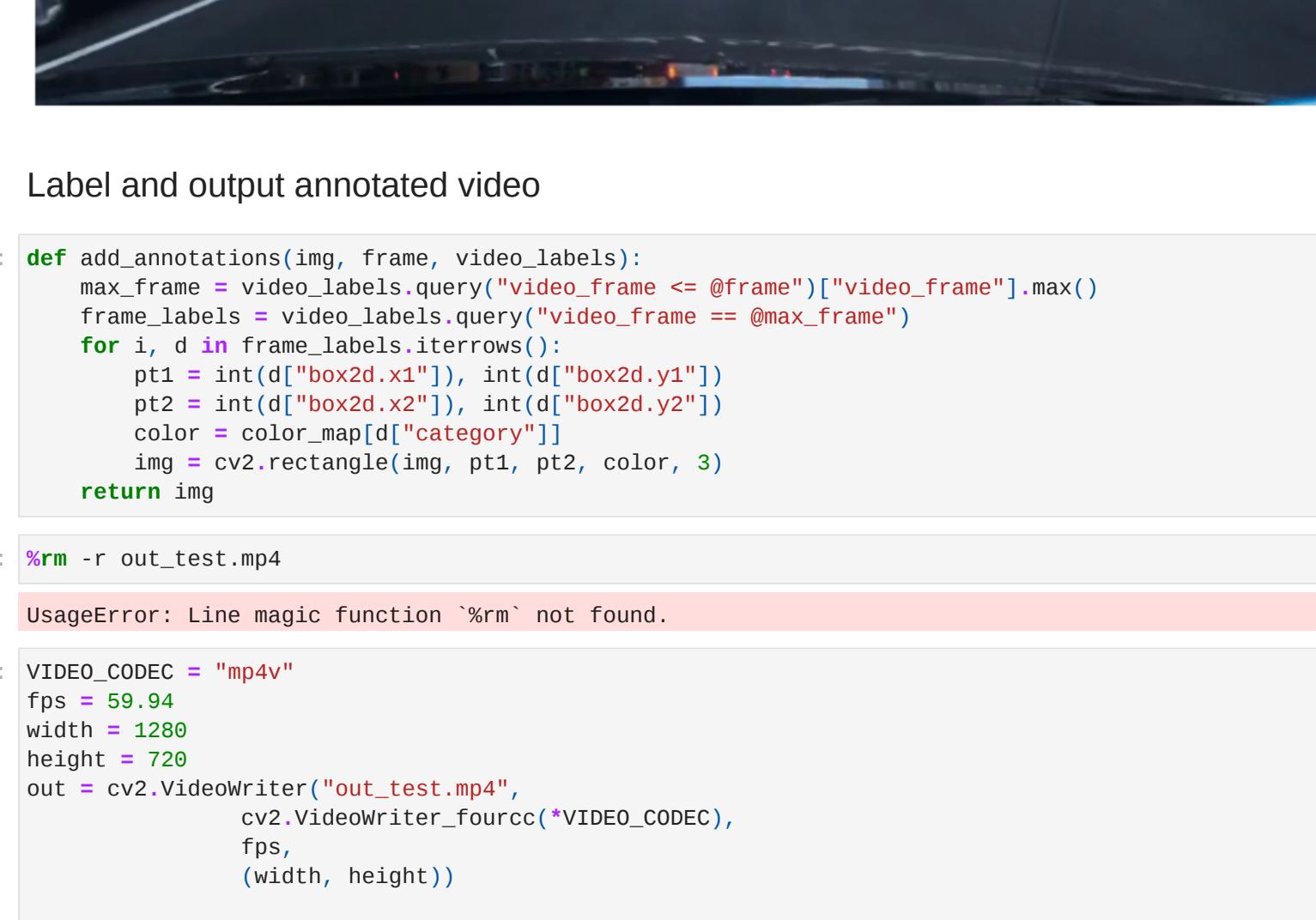
```
In [ ]: display_cv2_img(img)
```



```
In [ ]: img_example = img.copy()
frame_labels = video_labels.query("video_frame == 1035")
```

```
for i, d in frame_labels.iterrows():
    p1 = int(d['box2d.x1']), int(d['box2d.y1'])
    p2 = int(d['box2d.x2']), int(d['box2d.y2'])
    cv2.rectangle(img_example, p1, p2, color, 3)
```

```
display_cv2_img(img_example)
```

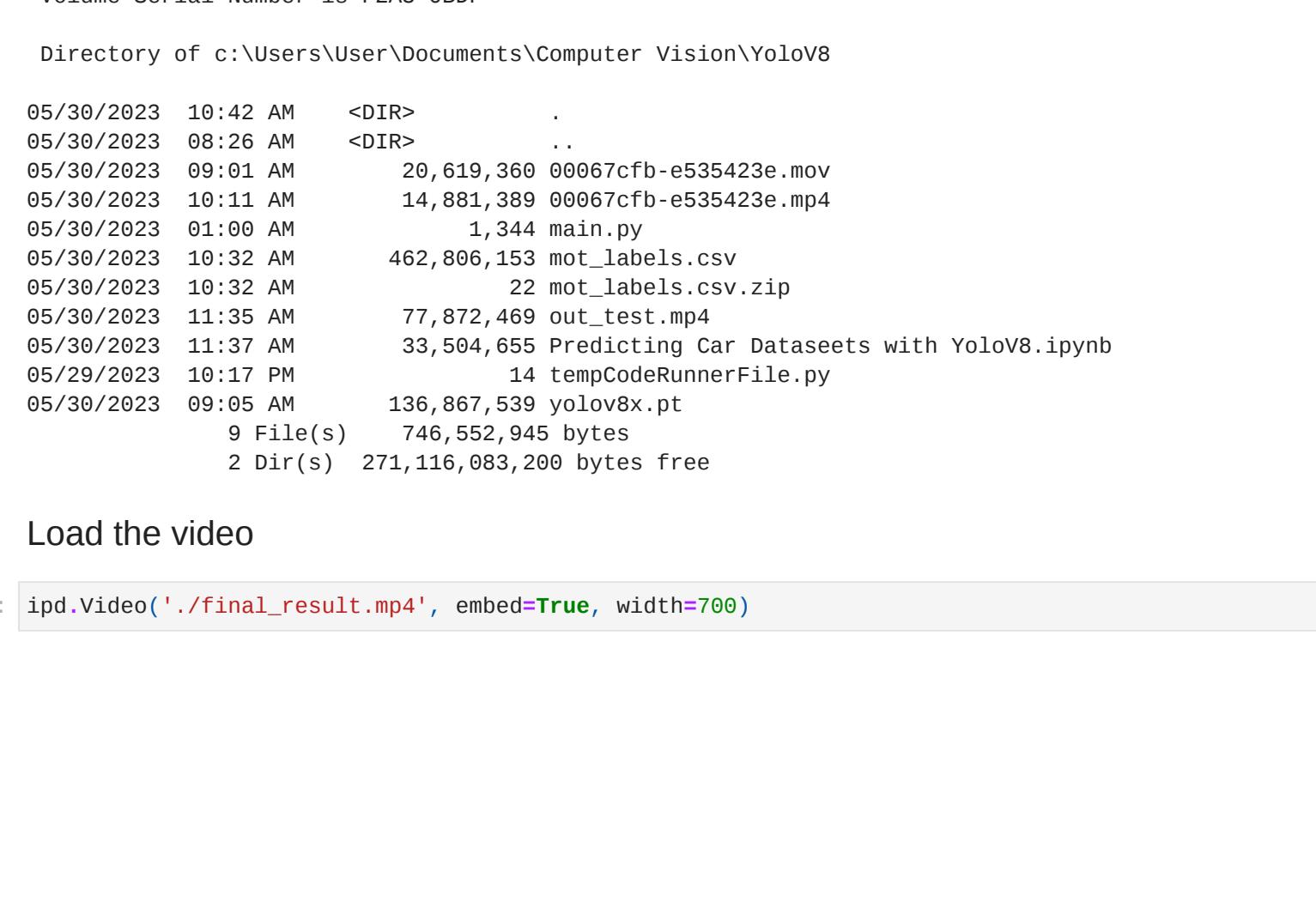


```
In [ ]: frame_labels = video_labels.query("video_frame == @frame")
```

```
img_example = img.copy()
cv2.imwrite('mot_labels.jpg', img_example)
```

```
for i, d in frame_labels.iterrows():
    p1 = int(d['box2d.x1']), int(d['box2d.y1'])
    p2 = int(d['box2d.x2']), int(d['box2d.y2'])
    color = color_map[d['category']]
    cv2.rectangle(img_example, p1, p2, color, 3)
```

```
display_cv2_img(img_example)
```



Label and output annotated video

```
In [ ]: def add_annotations(img, frame_labels):
    max_frame = video_labels.query("video_frame <= @frame")["video_frame"].max()
    frame_labels = video_labels.query("video.frame == @max_frame")
    for i, d in frame_labels.iterrows():
        p1 = int(d['box2d.x1']), int(d['box2d.y1'])
        p2 = int(d['box2d.x2']), int(d['box2d.y2'])
        color = color_map[d['category']]
        img = cv2.rectangle(img, p1, p2, color, 3)
    return img
```

```
In [ ]: %rm -r out_test.mp4
```

```
UsageError: Line magic function '%rm' not found.
```

```
In [ ]: VIDEO_CODEC = "mpav"
```

```
fps = 59.94
```

```
width = 1280
```

```
height = 720
```

```
out = cv2.VideoWriter("out_test.mp4",
                      cv2.VideoWriter_fourcc(*VIDEO_CODEC),
                      fps,
                      (width, height))
```

```
cap = cv2.VideoCapture('00067cfb-e535423e.mp4')
n_frames = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
```

```
for frame in tqdm(range(n_frames), total=n_frames):
    ret, img = cap.read()
    if ret == False:
        break
    img = add_annotations(img, frame, video_labels)
    out.write(img)
```

```
out.release()
cap.release()
```

```
10% [██████] 2405/2405 [00:32<00:00, 73.43it/s]
```

```
In [ ]: %dir
```

Volume in drive C is OS SSD

Volume Serial Number is F2A3-0BDF

Directory of c:\Users\user\Documents\Computer Vision\YOLOv8

```
05/30/2023 10:42 AM <DIR>
05/30/2023 08:26 AM <DIR>..
05/30/2023 09:10 AM 20,619,588 00067cfb-e535423e.mov
05/30/2023 10:11 AM 14,403,399 00067cfb-e535423e.mp4
05/30/2023 01:00 AM 1,344 main.py
05/30/2023 10:32 AM 462,896,159 mot_labels.csv
05/30/2023 10:32 AM 22 mot_labels.zip
05/30/2023 11:35 AM 77,872,469 out_test.mp4
05/30/2023 11:35 AM 33,594,859 Predicting Car Datasets with YOLOv8.ipynb
05/30/2023 10:17 AM 34 tempCodeGeneratorFile.py
05/30/2023 09:05 PM 136,867,539 yolov8x.pt
9 File(s) 746,552,945 bytes
2 Dir(s) 271,116,083,200 bytes free
```

Load the video

```
In [ ]: ipd.Video('./final_result.mp4', embed=True, width=700)
```

```
10% [██████] 2405/2405 [00:32<00:00, 73.43it/s]
```

```
In [ ]: %dir
```

Volume in drive C is OS SSD

Volume Serial Number is F2A3-0BDF

Directory of c:\Users\user\Documents\Computer Vision\YOLOv8

```
05/30/2023 10:42 AM <DIR>
05/30/2023 08:26 AM <DIR>..
05/30/2023 09:10 AM 20,619,588 00067cfb-e535423e.mov
05/30/2023 10:11 AM 14,403,399 00067cfb-e535423e.mp4
05/30/2023 01:00 AM 1,344 main.py
05/30/2023 10:32 AM 462,896,159 mot_labels.csv
05/30/2023 10:32 AM 22 mot_labels.zip
05/30/2023 11:35 AM 77,872,469 out_test.mp4
05/30/2023 11:35 AM 33,594,859 Predicting Car Datasets with YOLOv8.ipynb
05/30/2023 10:17 AM 34 tempCodeGeneratorFile.py
05/30/2023 09:05 PM 136,867,539 yolov8x.pt
9 File(s) 746,552,945 bytes
2 Dir(s) 271,116,083,200 bytes free
```

Out[]:

