```
!nvidia-smi
```

```
Mon Dec 15 16:11:45 2025
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 550.54.15              Driver Version: 550.54.15      CUDA Version: 12.4     |
|-----------------------------------------+------------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |               MIG M. |
|=========================================+========================+======================|
|   0  Tesla T4                       Off |   00000000:00:04.0 Off |                    0 |
| N/A   66C    P0             29W /   70W |    1238MiB /  15360MiB |      0%      Default |
|                                         |                        |                  N/A |
+-----------------------------------------+------------------------+----------------------+

+-----------------------------------------------------------------------------------------+
| Processes:                                                                              |
|  GPU   GI   CI        PID   Type   Process name                              GPU Memory |
|        ID   ID                                                               Usage      |
|=========================================================================================|
+-----------------------------------------------------------------------------------------+
```

```
!pip install -q ultralytics
```

```
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.2/1.2 MB 36.3 MB/s eta 0:00:00
```

```
from google.colab import drive
```

```
!apt-get install -y unrar
!unrar x data.rar
```

```
Extracting  data/labels/val/Cars357.txt                          OK
Extracting  data/labels/val/Cars358.txt                          OK
Extracting  data/labels/val/Cars359.txt                          OK
Extracting  data/labels/val/Cars360.txt                          OK
Extracting  data/labels/val/Cars361.txt                          OK
Extracting  data/labels/val/Cars362.txt                          OK
Extracting  data/labels/val.cache                                OK
All OK
```

```
!ls data/images/train | head
!ls data/labels/train | head
```

```
Cars0.png
Cars100.png
Cars101.png
Cars102.png
Cars103.png
Cars104.png
Cars105.png
Cars106.png
Cars107.png
Cars108.png
Cars0.txt
Cars100.txt
Cars101.txt
Cars102.txt
Cars103.txt
Cars104.txt
Cars105.txt
Cars106.txt
Cars107.txt
Cars108.txt
```

```
!ls
```

```
data.yaml  DSC_1104.JPG  images  labels  runs  yolo11n.pt  yolov8n.pt
```

```
from google.colab import files
uploaded = files.upload()
```

```
Choose Files  data.rar
data.rar(n/a) - 212734382 bytes, last modified: 12/14/2025 - 100% done
Saving data.rar to data.rar
```

```
from ultralytics import YOLO

model = YOLO("yolov8n.pt")
```

```
Creating new Ultralytics Settings v0.0.6 file ✅
View Ultralytics Settings with 'yolo settings' or at '/root/.config/Ultralytics/settings.json'
Update Settings with 'yolo settings key=value', i.e. 'yolo settings runs_dir=path/to/dir'. For help see https://docs.ultraly
Downloading https://github.com/ultralytics/assets/releases/download/v8.3.0/yolov8n.pt to 'yolov8n.pt': 100% ━━━━━━━━━━━━ 6.2
```

```
!ls /content/data/images/train | head
!ls /content/data/images/val | head
!ls /content/data/labels/train | head
```

```
Cars0.png
Cars100.png
Cars101.png
Cars102.png
Cars103.png
Cars104.png
Cars105.png
Cars106.png
Cars107.png
Cars108.png
Cars182.png
Cars183.png
Cars184.png
Cars185.png
Cars186.png
Cars187.png
Cars188.png
Cars189.png
Cars190.png
Cars191.png
Cars0.txt
Cars100.txt
Cars101.txt
Cars102.txt
```

```
Cars103.txt
Cars104.txt
Cars105.txt
Cars106.txt
Cars107.txt
Cars108.txt
```

```
%%writefile /content/data/data.yaml
path: /content/data
train: images/train
val: images/val

nc: 1
names: ["license_plate"]
```

```
Overwriting /content/data/data.yaml
```

```
from ultralytics import YOLO

model = YOLO("yolov8n.pt")

model.train(
    data="/content/data/data.yaml",
    epochs=30,
    imgsz=640,
    batch=16,
    device=0
)
```

```
                0.86957,     0.86957,     0.86957,     0.86957,     0.86957,     0.86957,     0.86957,     0.86957,
```

```
0.86957,     0.86957,     0.86957,     0.86957,     0.86957,     0.86957,     0.86957,     0.86957,     0.86957,
```

```
model.val()
```

```
            0.76851,     0.76798,     0.76745,     0.76692,     0.76639,     0.76586,     0.76532,     0.76479,
0.76426,     0.76373,      0.7632,     0.76267,     0.76214,     0.76161,     0.76108,     0.76049,     0.75988,
0.75927,     0.75866,     0.75804,     0.75743,     0.75682,     0.75621,
            0.7556,     0.75498,     0.75437,     0.75376,     0.75315,     0.75253,     0.75192,     0.75131,
0.7507,     0.75008,     0.74953,     0.74899,     0.74844,      0.7479,     0.74736,     0.74681,     0.74627,
0.74572,     0.74518,     0.74464,     0.74409,     0.74355,       0.743,
            0.74246,     0.74192,     0.74137,     0.74083,     0.74028,     0.73974,      0.7392,     0.73785,
0.73639,     0.73494,     0.73348,     0.73202,     0.73057,     0.72911,     0.72826,     0.72826,     0.72826,
0.72826,     0.72826,     0.72826,     0.72826,     0.72826,     0.72826,
            0.72826,     0.72826,     0.72826,     0.72826,     0.72826,     0.72826,     0.72586,     0.71988,
0.7163,     0.71443,     0.71256,     0.71069,     0.70883,     0.70696,     0.69565,     0.69565,     0.69565,
0.69565,     0.69565,     0.69565,     0.69565,     0.69565,     0.69565,
            0.69565,     0.69565,     0.69565,     0.69565,     0.69565,     0.69565,     0.69565,     0.69565,
0.69565,     0.69565,     0.69565,     0.69565,     0.69565,     0.69565,     0.69565,     0.69565,     0.69565,
0.69565,     0.69565,     0.69565,     0.69565,     0.69565,     0.69565,
            0.69565,     0.69565,     0.69565,     0.69565,     0.69565,     0.69565,     0.69565,     0.69565,
0.69565,     0.69565,     0.69565,     0.69565,     0.69565,     0.68291,     0.67904,     0.67516,     0.65945,
0.65413,     0.65164,      0.6508,     0.64996,     0.64912,     0.64828,
            0.64744,      0.6466,     0.64576,     0.64492,     0.64409,     0.64325,     0.64241,     0.64157,
0.63607,     0.62968,     0.62676,     0.62385,     0.62093,     0.61213,      0.6072,     0.60523,     0.60325,
0.60127,      0.5993,     0.59683,     0.57525,     0.57286,     0.57048,
            0.56809,      0.5657,     0.56251,     0.55911,     0.55571,     0.54956,     0.54348,     0.54348,
0.54348,     0.52043,      0.5186,     0.51677,     0.51493,      0.5131,     0.51127,      0.5088,     0.50615,
0.5035,     0.50085,     0.49873,     0.49687,      0.495,     0.49314,
            0.49128,     0.48941,     0.48913,     0.48868,     0.48665,     0.48462,     0.48259,     0.48057,
0.47854,     0.47285,     0.46641,      0.4589,     0.44995,      0.4451,      0.4441,      0.4431,     0.44209,
0.44109,     0.44009,     0.43909,     0.43809,     0.43709,     0.43609,
            0.43508,     0.43297,     0.43039,      0.4278,     0.42522,     0.41755,     0.40909,       0.403,
0.38749,     0.38307,     0.36713,      0.3584,     0.35765,     0.35689,     0.35613,     0.35537,     0.35462,
0.35386,      0.3531,     0.35235,     0.35159,     0.35083,     0.35008,
            0.34932,     0.34856,     0.34333,     0.32806,     0.31228,     0.30787,     0.30018,     0.28758,
0.27995,      0.2738,     0.26925,     0.26551,     0.26176,     0.26087,     0.24675,     0.24302,     0.23928,
0.23439,     0.22945,     0.22725,     0.22591,     0.22458,     0.22324,
            0.22191,     0.22057,     0.21923,      0.2179,     0.21104,     0.19988,     0.18452,     0.18248,
0.18043,     0.17838,     0.17634,     0.17429,     0.15536,     0.13839,     0.13502,     0.13165,     0.12925,
0.12738,     0.12552,     0.12366,      0.1218,     0.11994,     0.11849,
            0.11716,     0.11582,     0.11448,     0.11315,     0.11181,     0.11047,     0.10913,     0.10577,
0.10141,    0.097585,    0.096223,    0.094861,    0.093499,    0.092137,    0.090775,    0.089413,    0.088051,
0.086231,    0.082534,    0.078838,    0.068662,    0.059559,    0.054237,
           0.053647,    0.053057,    0.052467,    0.051877,    0.051287,    0.050697,    0.050107,    0.049517,
0.048927,    0.048337,    0.047747,    0.047157,    0.046567,    0.045977,    0.045387,    0.044797,    0.044207,
0.043617,    0.042065,    0.040216,    0.038367,    0.036518,    0.034669,
            0.03282,    0.020419,    0.018688,    0.016957,    0.015226,    0.013495,    0.011764,           0,
0,           0,           0,           0,           0,           0,           0,           0,           0,           0,
0,           0,           0,           0,           0,
                 0,           0,           0,           0,           0,           0,           0,           0,
0,           0,           0,           0,           0,           0,           0,           0,           0,           0,
0,           0,           0,           0,           0,
                 0,           0,           0,           0,           0,           0,           0,           0,
0,           0,           0,           0,           0,           0,           0,           0,           0,           0,
0,           0,           0,           0,           0,
                 0,           0,           0,           0,           0,           0,           0,           0,
0,           0,           0]]), 'Confidence', 'Recall']]
fitness: np.float64(0.5114619798160605)
keys: ['metrics/precision(B)', 'metrics/recall(B)', 'metrics/mAP50(B)', 'metrics/mAP50-95(B)']
maps: array([    0.51146])
names: {0: 'license_plate'}
nt_per_class: array([92])
```

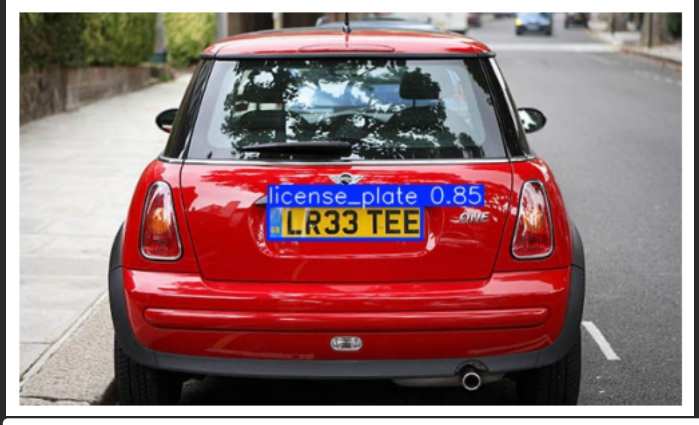
```
model.predict(
    source="/content/data/images/val",
    conf=0.25,
    save=True
)
```

```
         [ 67,  40,  34],
```

```
         [ 69,  42,  33]],

        [[ 28,  20,  18],
         [ 24,  15,  13],
         [ 25,  16,  14],
         ...,
         [ 66,  37,  36],
         [ 62,  34,  28],
         [ 63,  36,  26]]], dtype=uint8)
 orig_shape: (300, 400)
 path: '/content/data/images/val/Cars361.png'
 probs: None
 save_dir: '/content/data/runs/detect/predict'
 speed: {'preprocess': 4.87520299998323, 'inference': 10.322264000024006, 'postprocess': 1.4196649999576039},
 ultralytics.engine.results.Results object with attributes:

 boxes: ultralytics.engine.results.Boxes object
 keypoints: None
 masks: None
 names: {0: 'license_plate'}
 obb: None
 orig_img: array([[[  1,   0,   0],
         [  1,   0,   0],
         [  1,   0,   0],
         ...,
         [  1,   0,   0],
         [  1,   0,   0],
         [  1,   0,   0]],

        [[  1,   0,   0],
         [  1,   0,   0],
         [  1,   0,   0],
         ...,
         [  1,   0,   0],
         [  1,   0,   0],
         [  1,   0,   0]],

        [[  1,   0,   0],
         [  1,   0,   0],
         [  1,   0,   0],
```

```
from PIL import Image
import matplotlib.pyplot as plt

img = Image.open("/content/data/runs/detect/predict/Cars201.jpg")
plt.imshow(img)
plt.axis("off")
```

```
(np.float64(-0.5), np.float64(589.5), np.float64(349.5), np.float64(-0.5))
```




```
from google.colab import files
uploaded = files.upload()
```

Choose Files DSC_0975.JPG

DSC_0975.JPG(image/jpeg) - 616139 bytes, last modified: 1/9/2023 - 100% done

```
Saving DSC_0975.JPG to DSC_0975.JPG
```

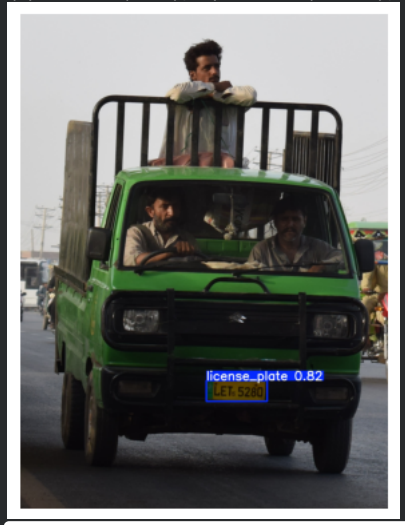
```
model.predict(
    source="/content/data/DSC_0975.JPG",
    conf=0.3,
    save=True
)
```

```
image 1/1 /content/data/DSC_0975.JPG: 640x480 1 license_plate, 16.6ms
```

```
Speed: 6.2ms preprocess, 16.6ms inference, 2.0ms postprocess per image at shape (1, 3, 640, 480)
Results saved to /content/data/runs/detect/predict2
[ultralytics.engine.results.Results object with attributes:

 boxes: ultralytics.engine.results.Boxes object
 keypoints: None
 masks: None
 names: {0: 'license_plate'}
 obb: None
 orig_img: array([[[219, 217, 217],
         [218, 216, 216],
         [218, 216, 216],
         ...,
         [221, 219, 218],
         [221, 219, 218],
         [220, 218, 217]],

        [[219, 217, 217],
         [218, 216, 216],
         [217, 215, 215],
         ...,
         [220, 218, 217],
         [220, 218, 217],
         [220, 218, 217]],

        [[219, 217, 217],
         [218, 216, 216],
         [217, 215, 215],
         ...,
         [220, 218, 217],
         [220, 218, 217],
         [220, 218, 217]],

        ...,

        [[ 42,  45,  50],
         [ 41,  44,  49],
         [ 38,  41,  46],
         ...,
         [ 70,  68,  68],
         [ 69,  67,  67],
         [ 72,  70,  70]],

        [[ 39,  42,  47],
         [ 39,  42,  47],
         [ 39,  42,  47],
         ...,
         [ 69,  67,  67],
         [ 68,  66,  66],
         [ 65,  63,  63]],

        [[ 36,  39,  44],
         [ 38,  41,  46],
         [ 40,  43,  48],
         ...,
```

```
from PIL import Image
import matplotlib.pyplot as plt

img = Image.open("/content/data/runs/detect/predict2/DSC_0975.jpg")
plt.imshow(img)
plt.axis("off")
```

```
(np.float64(-0.5), np.float64(1504.5), np.float64(2024.5), np.float64(-0.5))
```



```
!pip install easyocr opencv-python-headless
```

```
Requirement already satisfied: easyocr in /usr/local/lib/python3.12/dist-packages (1.7.2)
Requirement already satisfied: opencv-python-headless in /usr/local/lib/python3.12/dist-packages (4.12.0.88)
Requirement already satisfied: torch in /usr/local/lib/python3.12/dist-packages (from easyocr) (2.9.0+cu126)
Requirement already satisfied: torchvision>=0.5 in /usr/local/lib/python3.12/dist-packages (from easyocr) (0.24.0+cu126)
Requirement already satisfied: scipy in /usr/local/lib/python3.12/dist-packages (from easyocr) (1.16.3)
Requirement already satisfied: numpy in /usr/local/lib/python3.12/dist-packages (from easyocr) (2.0.2)
Requirement already satisfied: Pillow in /usr/local/lib/python3.12/dist-packages (from easyocr) (11.3.0)
Requirement already satisfied: scikit-image in /usr/local/lib/python3.12/dist-packages (from easyocr) (0.25.2)
Requirement already satisfied: python-bidi in /usr/local/lib/python3.12/dist-packages (from easyocr) (0.6.7)
Requirement already satisfied: PyYAML in /usr/local/lib/python3.12/dist-packages (from easyocr) (6.0.3)
Requirement already satisfied: Shapely in /usr/local/lib/python3.12/dist-packages (from easyocr) (2.1.2)
Requirement already satisfied: pyclipper in /usr/local/lib/python3.12/dist-packages (from easyocr) (1.4.0)
Requirement already satisfied: ninja in /usr/local/lib/python3.12/dist-packages (from easyocr) (1.13.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-packages (from torch->easyocr) (3.20.0)
Requirement already satisfied: typing-extensions>=4.10.0 in /usr/local/lib/python3.12/dist-packages (from torch->easyocr) (4
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from torch->easyocr) (75.2.0)
Requirement already satisfied: sympy>=1.13.3 in /usr/local/lib/python3.12/dist-packages (from torch->easyocr) (1.14.0)
Requirement already satisfied: networkx>=2.5.1 in /usr/local/lib/python3.12/dist-packages (from torch->easyocr) (3.6.1)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-packages (from torch->easyocr) (3.1.6)
Requirement already satisfied: fsspec>=0.8.5 in /usr/local/lib/python3.12/dist-packages (from torch->easyocr) (2025.3.0)
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch->easyc
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch->eas
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.6.80 in /usr/local/lib/python3.12/dist-packages (from torch->easyc
Requirement already satisfied: nvidia-cudnn-cu12==9.10.2.21 in /usr/local/lib/python3.12/dist-packages (from torch->easyocr)
Requirement already satisfied: nvidia-cublas-cu12==12.6.4.1 in /usr/local/lib/python3.12/dist-packages (from torch->easyocr)
Requirement already satisfied: nvidia-cufft-cu12==11.3.0.4 in /usr/local/lib/python3.12/dist-packages (from torch->easyocr)
Requirement already satisfied: nvidia-curand-cu12==10.3.7.77 in /usr/local/lib/python3.12/dist-packages (from torch->easyocr
Requirement already satisfied: nvidia-cusolver-cu12==11.7.1.2 in /usr/local/lib/python3.12/dist-packages (from torch->easyoc
Requirement already satisfied: nvidia-cusparse-cu12==12.5.4.2 in /usr/local/lib/python3.12/dist-packages (from torch->easyoc
Requirement already satisfied: nvidia-cusparselt-cu12==0.7.1 in /usr/local/lib/python3.12/dist-packages (from torch->easyocr
Requirement already satisfied: nvidia-nccl-cu12==2.27.5 in /usr/local/lib/python3.12/dist-packages (from torch->easyocr) (2.
Requirement already satisfied: nvidia-nvshmem-cu12==3.3.20 in /usr/local/lib/python3.12/dist-packages (from torch->easyocr)
Requirement already satisfied: nvidia-nvtx-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch->easyocr) (1
Requirement already satisfied: nvidia-nvjitlink-cu12==12.6.85 in /usr/local/lib/python3.12/dist-packages (from torch->easyoc
Requirement already satisfied: nvidia-cufile-cu12==1.11.1.6 in /usr/local/lib/python3.12/dist-packages (from torch->easyocr)
Requirement already satisfied: triton==3.5.0 in /usr/local/lib/python3.12/dist-packages (from torch->easyocr) (3.5.0)
Requirement already satisfied: imageio!=2.35.0,>=2.33 in /usr/local/lib/python3.12/dist-packages (from scikit-image->easyocr
Requirement already satisfied: tifffile>=2022.8.12 in /usr/local/lib/python3.12/dist-packages (from scikit-image->easyocr) (
Requirement already satisfied: packaging>=21 in /usr/local/lib/python3.12/dist-packages (from scikit-image->easyocr) (25.0)
Requirement already satisfied: lazy-loader>=0.4 in /usr/local/lib/python3.12/dist-packages (from scikit-image->easyocr) (0.4
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.12/dist-packages (from sympy>=1.13.3->torch->eas
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from jinja2->torch->easyocr) (3.0
```

```
from ultralytics import YOLO

model = YOLO("/content/data/runs/detect/train/weights/best.pt")
```

```
results = model.predict(
    source="/content/data/runs/detect/predict2/DSC_0975.jpg",
    conf=0.3,
    save=False
)
```

```
image 1/1 /content/data/runs/detect/predict2/DSC_0975.jpg: 640x480 4 license_plates, 46.9ms
Speed: 15.1ms preprocess, 46.9ms inference, 6.8ms postprocess per image at shape (1, 3, 640, 480)
```

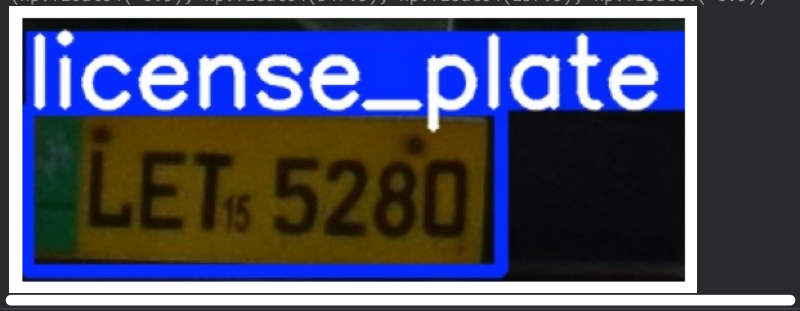
```
import cv2
import numpy as np

# Load image
img = cv2.imread("/content/data/runs/detect/predict2/DSC_0975.jpg")

# YOLO result (single image)
r = results[0]

# Get boxes
boxes = r.boxes.xyxy.cpu().numpy()

plate_crops = []

for box in boxes:
    x1, y1, x2, y2 = map(int, box)
    crop = img[y1:y2, x1:x2]
    plate_crops.append(crop)
```

```
import easyocr

reader = easyocr.Reader(['en'], gpu=True)  # GPU works in Colab

for i, plate in enumerate(plate_crops):
    results_ocr = reader.readtext(plate)

    print(f"\nPlate {i+1}:")
    for bbox, text, confidence in results_ocr:
        print(f"Detected Text: {text} | Confidence: {confidence:.2f}")
```

```
Plate 1:
Detected Text: license_plate | Confidence: 0.76
Detected Text: LETs 5280| | Confidence: 0.74

Plate 2:
Detected Text: license_plate | Confidence: 0.76
Detected Text: 0.82 | Confidence: 1.00
Detected Text: LETs 5280| | Confidence: 0.69

Plate 3:
Detected Text: LETs 5280| | Confidence: 0.67

Plate 4:
Detected Text: Iicense_pi | Confidence: 0.76
Detected Text: LETs 5280] | Confidence: 0.63
```

```
import matplotlib.pyplot as plt

plt.imshow(cv2.cvtColor(plate_crops[0], cv2.COLOR_BGR2RGB))
plt.axis("off")
```

```
(np.float64(-0.5), np.float64(347.5), np.float64(137.5), np.float64(-0.5))
```



Start coding or generate with AI.