```
!nvidia-smi
```

```
Fri Jan 10 06:56:55 2025
+---------------------------------------------------------------------------------------+
| NVIDIA-SMI 535.104.05             Driver Version: 535.104.05   CUDA Version: 12.2     |
|-----------------------------------------+----------------------+----------------------+
| 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   37C    P8               9W /  70W |      0MiB / 15360MiB |      0%      Default |
|                                         |                      |                  N/A |
+-----------------------------------------+----------------------+----------------------+

+---------------------------------------------------------------------------------------+
| Processes:                                                                            |
|  GPU   GI   CI        PID   Type   Process name                            GPU Memory |
|        ID   ID                                                             Usage      |
|=======================================================================================|
|  No running processes found                                                           |
+---------------------------------------------------------------------------------------+
```

```
!pip install ultralytics==8.0.20
from IPython import display
display.clear_output()

import ultralytics
ultralytics.checks()
```

```
Ultralytics YOLOv8.0.20 🚀 Python-3.10.12 torch-2.5.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
Setup complete ✅ (2 CPUs, 12.7 GB RAM, 32.5/112.6 GB disk)
```

```
from ultralytics import YOLO
from IPython.display import display, Image
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
%cd /content/drive/MyDrive/Innovation_ANPR
```

```
/content/drive/MyDrive/Innovation_ANPR
```

```
!ls
```

```
ANPR_YOLOv8.ipynb  datasets           number-plate.yaml  runs
app.py             number_plates.csv  output.mp4         yolov8s.pt
```

```
pip install -U albumentations
```

```
Requirement already satisfied: albumentations in /usr/local/lib/python3.10/dist-packages (1.4.20)
Collecting albumentations
  Downloading albumentations-2.0.0-py3-none-any.whl.metadata (38 kB)
Requirement already satisfied: numpy>=1.24.4 in /usr/local/lib/python3.10/dist-packages (from albumentations) (1.26.4)
Requirement already satisfied: scipy>=1.10.0 in /usr/local/lib/python3.10/dist-packages (from albumentations) (1.13.1)
Requirement already satisfied: PyYAML in /usr/local/lib/python3.10/dist-packages (from albumentations) (6.0.2)
Requirement already satisfied: pydantic>=2.9.2 in /usr/local/lib/python3.10/dist-packages (from albumentations) (2.10.4)
Collecting albucore==0.0.23 (from albumentations)
  Downloading albucore-0.0.23-py3-none-any.whl.metadata (5.3 kB)
Requirement already satisfied: opencv-python-headless>=4.9.0.80 in /usr/local/lib/python3.10/dist-packages (from albumentations) (4
Requirement already satisfied: stringzilla>=3.10.4 in /usr/local/lib/python3.10/dist-packages (from albucore==0.0.23->albumentations
Collecting simsimd>=5.9.2 (from albucore==0.0.23->albumentations)
  Downloading simsimd-6.2.1-cp310-cp310-manylinux_2_28_x86_64.whl.metadata (66 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 66.0/66.0 kB 3.8 MB/s eta 0:00:00
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.10/dist-packages (from pydantic>=2.9.2->albumentatic
Requirement already satisfied: pydantic-core==2.27.2 in /usr/local/lib/python3.10/dist-packages (from pydantic>=2.9.2->albumentatior
Requirement already satisfied: typing-extensions>=4.12.2 in /usr/local/lib/python3.10/dist-packages (from pydantic>=2.9.2->albumenta
Downloading albumentations-2.0.0-py3-none-any.whl (273 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 273.9/273.9 kB 14.8 MB/s eta 0:00:00
Downloading albucore-0.0.23-py3-none-any.whl (14 kB)
Downloading simsimd-6.2.1-cp310-cp310-manylinux_2_28_x86_64.whl (632 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 632.7/632.7 kB 30.0 MB/s eta 0:00:00
Installing collected packages: simsimd, albucore, albumentations
  Attempting uninstall: albucore
    Found existing installation: albucore 0.0.19
    Uninstalling albucore-0.0.19:
      Successfully uninstalled albucore-0.0.19
  Attempting uninstall: albumentations
    Found existing installation: albumentations 1.4.20
    Uninstalling albumentations-1.4.20:
      Successfully uninstalled albumentations-1.4.20
Successfully installed albucore-0.0.23 albumentations-2.0.0 simsimd-6.2.1
```

```
!yolo task=detect mode=train model=yolov8s.pt data=number-plate.yaml epochs=20 plots=True
```

Show hidden output

```
!yolo task=detect mode=train model=yolov8s.pt data=number-plate.yaml epochs=50 plots=True
```

Show hidden output

```
!yolo task=detect mode=train model=yolov8s.pt data=number-plate.yaml epochs=60 plots=True
```

Show hidden output

```
!yolo task=detect mode=train model=yolov8s.pt data=number-plate.yaml epochs=70 plots=True
```

Show hidden output

```
!yolo detect val model=/content/drive/MyDrive/Innovation_ANPR/runs/detect/train3/weights/best.pt
```

Show hidden output

```
!pip install easyocr
```

```
Collecting easyocr
  Downloading easyocr-1.7.2-py3-none-any.whl.metadata (10 kB)
Requirement already satisfied: torch in /usr/local/lib/python3.10/dist-packages (from easyocr) (2.5.1+cu121)
Requirement already satisfied: torchvision>=0.5 in /usr/local/lib/python3.10/dist-packages (from easyocr) (0.20.1+cu121)
Requirement already satisfied: opencv-python-headless in /usr/local/lib/python3.10/dist-packages (from easyocr) (4.10.0.84)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (from easyocr) (1.13.1)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from easyocr) (1.26.4)
Requirement already satisfied: Pillow in /usr/local/lib/python3.10/dist-packages (from easyocr) (11.1.0)
Requirement already satisfied: scikit-image in /usr/local/lib/python3.10/dist-packages (from easyocr) (0.25.0)
Collecting python-bidi (from easyocr)
  Downloading python_bidi-0.6.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (4.9 kB)
Requirement already satisfied: PyYAML in /usr/local/lib/python3.10/dist-packages (from easyocr) (6.0.2)
Requirement already satisfied: Shapely in /usr/local/lib/python3.10/dist-packages (from easyocr) (2.0.6)
Collecting pyclipper (from easyocr)
  Downloading pyclipper-1.3.0.post6-cp310-cp310-manylinux_2_12_x86_64.manylinux2010_x86_64.whl.metadata (9.0 kB)
Collecting ninja (from easyocr)
  Downloading ninja-1.11.1.3-py3-none-manylinux_2_12_x86_64.manylinux2010_x86_64.whl.metadata (5.3 kB)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from torch->easyocr) (3.16.1)
Requirement already satisfied: typing-extensions>=4.8.0 in /usr/local/lib/python3.10/dist-packages (from torch->easyocr) (4.12.2)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch->easyocr) (3.4.2)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from torch->easyocr) (3.1.5)
Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages (from torch->easyocr) (2024.10.0)
Requirement already satisfied: sympy==1.13.1 in /usr/local/lib/python3.10/dist-packages (from torch->easyocr) (1.13.1)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from sympy==1.13.1->torch->easyocr) (1
Requirement already satisfied: imageio!=2.35.0,>=2.33 in /usr/local/lib/python3.10/dist-packages (from scikit-image->easyocr) (2.36
Requirement already satisfied: tifffile>=2022.8.12 in /usr/local/lib/python3.10/dist-packages (from scikit-image->easyocr) (2024.12
Requirement already satisfied: packaging>=21 in /usr/local/lib/python3.10/dist-packages (from scikit-image->easyocr) (24.2)
Requirement already satisfied: lazy-loader>=0.4 in /usr/local/lib/python3.10/dist-packages (from scikit-image->easyocr) (0.4)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->torch->easyocr) (3.0.2)
Downloading easyocr-1.7.2-py3-none-any.whl (2.9 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.9/2.9 MB 41.3 MB/s eta 0:00:00
Downloading ninja-1.11.1.3-py3-none-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (422 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 422.9/422.9 kB 33.9 MB/s eta 0:00:00
Downloading pyclipper-1.3.0.post6-cp310-cp310-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (912 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 912.2/912.2 kB 54.7 MB/s eta 0:00:00
Downloading python_bidi-0.6.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (286 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 286.8/286.8 kB 22.9 MB/s eta 0:00:00
Installing collected packages: python-bidi, pyclipper, ninja, easyocr
Successfully installed easyocr-1.7.2 ninja-1.11.1.3 pyclipper-1.3.0.post6 python-bidi-0.6.3
```

```
from easyocr import Reader
import time
import torch
import cv2
import os
import csv


CONFIDENCE_THRESHOLD = 0.4
COLOR = (0, 255, 0)


def detect_number_plates(image, model, display=False):
    start = time.time()
    detections = model.predict(image)[0].boxes.data
    if detections.shape != torch.Size([0, 6]):
        boxes = []
        confidences = []
        for detection in detections:
            confidence = detection[4]
            if float(confidence) < CONFIDENCE_THRESHOLD:
                continue
            boxes.append(detection[:4])
            confidences.append(detection[4])

        print(f"{len(boxes)} Number plate(s) have been detected.")
        number_plate_list= []

        # loop over the bounding boxes
        for i in range(len(boxes)):
            # extract the bounding box coordinates
            xmin, ymin, xmax, ymax = int(boxes[i][0]), int(boxes[i][1]),\
                                     int(boxes[i][2]), int(boxes[i][3])
            # append the bounding box of the number plate
            number_plate_list.append([[xmin, ymin, xmax, ymax]])

            # draw the bounding box and the label on the image
            cv2.rectangle(image, (xmin, ymin), (xmax, ymax), COLOR, 2)
            text = "Number Plate: {:.2f}%".format(confidences[i] * 100)
            cv2.putText(image, text, (xmin, ymin - 5),
                        cv2.FONT_HERSHEY_SIMPLEX, 0.5, COLOR, 2)

            if display:
                # crop the detected number plate region
                number_plate = image[ymin:ymax, xmin:xmax]
                # display the number plate
                cv2_imshow(number_plate)

        end = time.time()
        # show the time it took to detect the number plates
        print(f"Time to detect the number plates: {(end - start) * 1000:.0f} milliseconds")
        # return the list containing the bounding
        # boxes of the number plates
        return number_plate_list
    # if there are no detections, show a custom message
    else:
        print("No number plates have been detected.")
        return []
```

```
model = YOLO('/content/drive/MyDrive/Innovation_ANPR/runs/detect/train4/weights/best.pt')
```

```
/usr/local/lib/python3.10/dist-packages/ultralytics/nn/tasks.py:332: FutureWarning: You are using `torch.load` with `weights_only=Fa
  return torch.load(file, map_location='cpu')  # load
```

```
 file_path = "/content/drive/MyDrive/Innovation_ANPR/datasets/images/test/1419b9df-b7d2e8c5-233.jpeg"
```

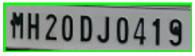
```
reader = Reader(['en'], gpu=True)
```

```
WARNING:easyocr.easyocr:Downloading detection model, please wait. This may take several minutes depending upon your network connecti
Progress: |██████████████████████████████████████████████████| 100.0% CompleteWARNING:easyocr.easyocr:Downloading recognition model,
Progress: |██████████████████████████████████████████████████| 100.0% Complete
```

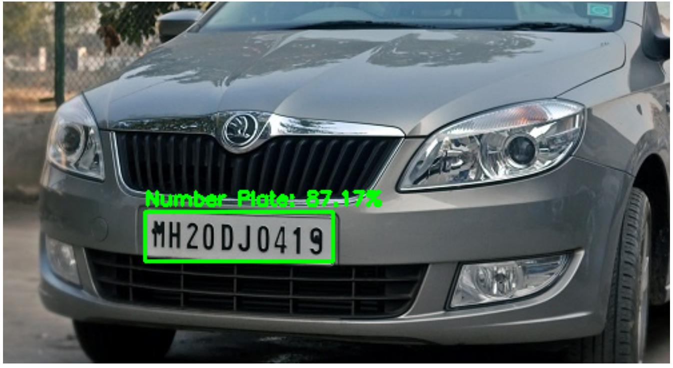
```
_, file_extension = os.path.splitext(file_path)
```

```
def recognize_number_plates(image_or_path, reader,
                            number_plate_list, write_to_csv=False):

    start = time.time()
    image = cv2.imread(image_or_path) if isinstance(image_or_path, str)\
                                      else image_or_path

    for i, box in enumerate(number_plate_list):
        np_image = image[box[0][1]:box[0][3], box[0][0]:box[0][2]]
        detection = reader.readtext(np_image, paragraph=True)

        if len(detection) == 0:
            text = ""
        else:
            text = str(detection[0][1])
        number_plate_list[i].append(text)
    if write_to_csv:
        csv_file = open("number_plates.csv", "w")
        csv_writer = csv.writer(csv_file)
        csv_writer.writerow(["image_path", "box", "text"])
        for box, text in number_plate_list:
            csv_writer.writerow([image_or_path, box, text])
        csv_file.close()

    end = time.time()
    print(f"Time to recognize the number plates: {(end - start) * 1000:.0f} milliseconds")

    return number_plate_list


if file_extension in ['.jpg', '.jpeg', '.png']:
    print("Processing the image...")

    image = cv2.imread(file_path)
    number_plate_list = detect_number_plates(image, model, display=True)
    cv2_imshow(image)

    # if there are any number plates detected, recognize them
    if number_plate_list != []:
        number_plate_list = recognize_number_plates(file_path, reader, number_plate_list, write_to_csv=True)

        for box, text in number_plate_list:
            cv2.putText(image, text, (box[0], box[3] + 15), cv2.FONT_HERSHEY_SIMPLEX, 0.5, COLOR, 2)
        cv2_imshow(image)

elif file_extension in ['.mp4', '.mkv', '.avi', '.wmv', '.mov']:
    print("Processing the video...")

    video_cap = cv2.VideoCapture(file_path)

    # grab the width and the height of the video stream
    frame_width = int(video_cap.get(cv2.CAP_PROP_FRAME_WIDTH))
    frame_height = int(video_cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
    fps = int(video_cap.get(cv2.CAP_PROP_FPS))

    # initialize the FourCC and a video writer object
    fourcc = cv2.VideoWriter_fourcc(*"mp4v")
    writer = cv2.VideoWriter("output.mp4", fourcc, fps, (frame_width, frame_height))
    while True:
        start = time.time()
        success, frame = video_cap.read()
        if not success:
            print("There are no more frames to process. Exiting the script...")
            break

        number_plate_list = detect_number_plates(frame, model)

        if number_plate_list != []:
            number_plate_list = recognize_number_plates(frame, reader, number_plate_list)

            for box, text in number_plate_list:
                cv2.putText(frame, text, (box[0], box[3] + 15), cv2.FONT_HERSHEY_SIMPLEX, 0.75, COLOR, 2)
        end = time.time()
        fps_text = f"FPS: {1 / (end - start):.2f}"
        cv2.putText(frame, fps_text, (50, 50), cv2.FONT_HERSHEY_SIMPLEX, 2, (0, 0, 255), 8)
        cv2_imshow(frame)
        writer.write(frame)
        if cv2.waitKey(10) == ord("q"):
            break
    video_cap.release()
    writer.release()
    cv2.destroyAllWindows()
```

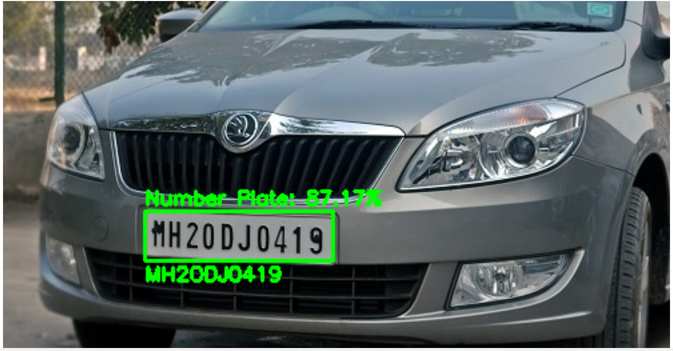
```
Processing the image...
1 Number plate(s) have been detected.
```



```
Time to detect the number plates: 89 milliseconds
```



```
Time to recognize the number plates: 80 milliseconds
```



```
from google.colab.patches import cv2_imshow
```