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

```
!pip install ultralytics
```

```
Collecting ultralytics
  Downloading ultralytics-8.4.12-py3-none-any.whl.metadata (38 kB)
Requirement already satisfied: numpy>=1.23.0 in /usr/local/lib/python3.12/dist-packages (from ultralytics) (2.0.2)
Requirement already satisfied: matplotlib>=3.3.0 in /usr/local/lib/python3.12/dist-packages (from ultralytics) (3.10.0)
Requirement already satisfied: opencv-python>=4.6.0 in /usr/local/lib/python3.12/dist-packages (from ultralytics) (4.13.0.
Requirement already satisfied: pillow>=7.1.2 in /usr/local/lib/python3.12/dist-packages (from ultralytics) (11.3.0)
Requirement already satisfied: pyyaml>=5.3.1 in /usr/local/lib/python3.12/dist-packages (from ultralytics) (6.0.3)
Requirement already satisfied: requests>=2.23.0 in /usr/local/lib/python3.12/dist-packages (from ultralytics) (2.32.4)
Requirement already satisfied: scipy>=1.4.1 in /usr/local/lib/python3.12/dist-packages (from ultralytics) (1.16.3)
Requirement already satisfied: torch>=1.8.0 in /usr/local/lib/python3.12/dist-packages (from ultralytics) (2.9.0+cu126)
Requirement already satisfied: torchvision>=0.9.0 in /usr/local/lib/python3.12/dist-packages (from ultralytics) (0.24.0+cu
Requirement already satisfied: psutil>=5.8.0 in /usr/local/lib/python3.12/dist-packages (from ultralytics) (5.9.5)
Requirement already satisfied: polars>=0.20.0 in /usr/local/lib/python3.12/dist-packages (from ultralytics) (1.31.0)
Collecting ultralytics-thop>=2.0.18 (from ultralytics)
  Downloading ultralytics_thop-2.0.18-py3-none-any.whl.metadata (14 kB)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib>=3.3.0->ultral
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.12/dist-packages (from matplotlib>=3.3.0->ultralytic
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib>=3.3.0->ultra
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib>=3.3.0->ultra
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib>=3.3.0->ultraly
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib>=3.3.0->ultral
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.12/dist-packages (from matplotlib>=3.3.0->ul
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests>=2.23.0-
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests>=2.23.0->ultralytics
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests>=2.23.0->ultra
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests>=2.23.0->ultra
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics) (3.20.
Requirement already satisfied: typing-extensions>=4.10.0 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ul
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics) (75.
Requirement already satisfied: sympy>=1.13.3 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics) (
Requirement already satisfied: networkx>=2.5.1 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics) (3.1.6)
Requirement already satisfied: fsspec>=0.8.5 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics) (
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch>=1
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.6.80 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8
Requirement already satisfied: nvidia-cudnn-cu12==9.10.2.21 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0-
Requirement already satisfied: nvidia-cublas-cu12==12.6.4.1 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0-
Requirement already satisfied: nvidia-cufft-cu12==11.3.0.4 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->
Requirement already satisfied: nvidia-curand-cu12==10.3.7.77 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0
Requirement already satisfied: nvidia-cusolver-cu12==11.7.1.2 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.
Requirement already satisfied: nvidia-cusparse-cu12==12.5.4.2 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.
Requirement already satisfied: nvidia-cusparselt-cu12==0.7.1 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0
Requirement already satisfied: nvidia-nccl-cu12==2.27.5 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ult
Requirement already satisfied: nvidia-nvshmem-cu12==3.3.20 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->
Requirement already satisfied: nvidia-nvtx-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ul
Requirement already satisfied: nvidia-nvjitlink-cu12==12.6.85 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.
Requirement already satisfied: nvidia-cufile-cu12==1.11.1.6 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0-
Requirement already satisfied: triton==3.5.0 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics) (
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.7->matplotlib>
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.12/dist-packages (from sympy>=1.13.3->torch>=1
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from jinja2->torch>=1.8.0->ultr
Downloading ultralytics-8.4.12-py3-none-any.whl (1.2 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.2/1.2 MB 21.4 MB/s eta 0:00:00
Downloading ultralytics_thop-2.0.18-py3-none-any.whl (28 kB)
Installing collected packages: ultralytics-thop, ultralytics
Successfully installed ultralytics-8.4.12 ultralytics-thop-2.0.18
```

```
from ultralytics import YOLO
import torch

# Verify GPU
print("GPU available:", torch.cuda.is_available())
print("GPU name:", torch.cuda.get_device_name(0))

# Choose model size (A100 can handle bigger models)
model = YOLO('yolov8m.pt')   # try 'yolov8m.pt' if you want even better accuracy

model.train(
    data='/content/drive/MyDrive/armhack/armhack.v3i.yolov8/data.yaml',  # change if your dataset path differs
    epochs=80,                # more epochs since A100 is fast
    imgsz=640,
    batch=32,                 # larger batch size for A100
    device=0,                 # GPU 0 (A100)
    project='/content/drive/MyDrive/armhack',
    name='yolov8m_a100_run'
```

```
)
```

```
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.ultra
GPU available: True
GPU name: NVIDIA A100-SXM4-40GB
Downloading https://github.com/ultralytics/assets/releases/download/v8.4.0/yolov8m.pt to 'yolov8m.pt': 100% ━━━━━━━━━━━━ 4
Ultralytics 8.4.12 🚀 Python-3.12.12 torch-2.9.0+cu126 CUDA:0 (NVIDIA A100-SXM4-40GB, 40507MiB)
engine/trainer: agnostic_nms=False, amp=True, angle=1.0, augment=False, auto_augment=randaugment, batch=32, bgr=0.0, box=7
Downloading https://ultralytics.com/assets/Arial.ttf to '/root/.config/Ultralytics/Arial.ttf': 100% ━━━━━━━━━━━━ 755.1KB 1
Overriding model.yaml nc=80 with nc=4

                   from  n    params  module                                       arguments
  0                  -1  1      1392  ultralytics.nn.modules.conv.Conv             [3, 48, 3, 2]
  1                  -1  1     41664  ultralytics.nn.modules.conv.Conv             [48, 96, 3, 2]
  2                  -1  2    111360  ultralytics.nn.modules.block.C2f             [96, 96, 2, True]
  3                  -1  1    166272  ultralytics.nn.modules.conv.Conv             [96, 192, 3, 2]
  4                  -1  4    813312  ultralytics.nn.modules.block.C2f             [192, 192, 4, True]
  5                  -1  1    664320  ultralytics.nn.modules.conv.Conv             [192, 384, 3, 2]
  6                  -1  4   3248640  ultralytics.nn.modules.block.C2f             [384, 384, 4, True]
  7                  -1  1   1991808  ultralytics.nn.modules.conv.Conv             [384, 576, 3, 2]
  8                  -1  2   3985920  ultralytics.nn.modules.block.C2f             [576, 576, 2, True]
  9                  -1  1    831168  ultralytics.nn.modules.block.SPPF            [576, 576, 5]
 10                  -1  1         0  torch.nn.modules.upsampling.Upsample         [None, 2, 'nearest']
 11             [-1, 6]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 12                  -1  2   1993728  ultralytics.nn.modules.block.C2f             [960, 384, 2]
 13                  -1  1         0  torch.nn.modules.upsampling.Upsample         [None, 2, 'nearest']
 14             [-1, 4]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 15                  -1  2    517632  ultralytics.nn.modules.block.C2f             [576, 192, 2]
 16                  -1  1    332160  ultralytics.nn.modules.conv.Conv             [192, 192, 3, 2]
 17            [-1, 12]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 18                  -1  2   1846272  ultralytics.nn.modules.block.C2f             [576, 384, 2]
 19                  -1  1   1327872  ultralytics.nn.modules.conv.Conv             [384, 384, 3, 2]
 20             [-1, 9]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 21                  -1  2   4207104  ultralytics.nn.modules.block.C2f             [960, 576, 2]
 22        [15, 18, 21]  1   3778012  ultralytics.nn.modules.head.Detect           [4, 16, None, [192, 384, 576]]
Model summary: 170 layers, 25,858,636 parameters, 25,858,620 gradients, 79.1 GFLOPs

Transferred 469/475 items from pretrained weights
Freezing layer 'model.22.dfl.conv.weight'
AMP: running Automatic Mixed Precision (AMP) checks...
Downloading https://github.com/ultralytics/assets/releases/download/v8.4.0/yolo26n.pt to 'yolo26n.pt': 100% ━━━━━━━━━━━━ 5
AMP: checks passed ✅
train: Fast image access ✅ (ping: 3.2±5.4 ms, read: 0.0±0.0 MB/s, size: 35.6 KB)
train: Scanning /content/drive/.shortcut-targets-by-id/1dCwsaKSoEmCYR36El50Xi3jx_5W5pVNm/armhack/armhack.v3i.yolov8/train/
train: New cache created: /content/drive/.shortcut-targets-by-id/1dCwsaKSoEmCYR36El50Xi3jx_5W5pVNm/armhack/armhack.v3i.yol
WARNING ⚠️ Box and segment counts should be equal, but got len(segments) = 1636, len(boxes) = 6155. To resolve this only b
albumentations: Blur(p=0.01, blur_limit=(3, 7)), MedianBlur(p=0.01, blur_limit=(3, 7)), ToGray(p=0.01, method='weighted_av
val: Fast image access ✅ (ping: 0.9±0.5 ms, read: 0.1±0.0 MB/s, size: 46.1 KB)
val: Scanning /content/drive/.shortcut-targets-by-id/1dCwsaKSoEmCYR36El50Xi3jx_5W5pVNm/armhack/armhack.v3i.yolov8/valid/la
val: New cache created: /content/drive/.shortcut-targets-by-id/1dCwsaKSoEmCYR36El50Xi3jx_5W5pVNm/armhack/armhack.v3i.yolov
WARNING ⚠️ Box and segment counts should be equal, but got len(segments) = 106, len(boxes) = 408. To resolve this only box
Plotting labels to /content/drive/.shortcut-targets-by-id/1dCwsaKSoEmCYR36El50Xi3jx_5W5pVNm/armhack/yolov8m_a100_run/label
optimizer: 'optimizer=auto' found, ignoring 'lr0=0.01' and 'momentum=0.937' and determining best 'optimizer', 'lr0' and 'm
optimizer: AdamW(lr=0.00125, momentum=0.9) with parameter groups 77 weight(decay=0.0), 84 weight(decay=0.0005), 83 bias(de
Image sizes 640 train, 640 val
Using 8 dataloader workers
Logging results to /content/drive/.shortcut-targets-by-id/1dCwsaKSoEmCYR36El50Xi3jx_5W5pVNm/armhack/yolov8m_a100_run
```

```
from ultralytics import YOLO

model = YOLO('/content/drive/MyDrive/armhack/yolov8m_a100_run/weights/best.pt')

model.predict(
    source='/content/manhole (5).jpg',
    conf=0.3,
    save=True
)
```

```
image 1/1 /content/manhole (5).jpg: 448x640 1 open_manhole, 9.8ms
Speed: 1.7ms preprocess, 9.8ms inference, 1.2ms postprocess per image at shape (1, 3, 448, 640)
Results saved to /content/runs/detect/predict17
[ultralytics.engine.results.Results object with attributes:

 boxes: ultralytics.engine.results.Boxes object
 keypoints: None
 masks: None
 names: {0: 'crack', 1: 'open_manhole', 2: 'pothole', 3: 'rutting'}
 obb: None
 orig_img: array([[[255, 255, 252],
         [244, 243, 239],
         [255, 255, 252],
         ...,
         [149, 167, 168],
```

```
           [105, 123, 122],
           [145, 163, 162]],

          [[255, 254, 250],
           [239, 238, 234],
           [255, 255, 251],
           ...,
           [141, 151, 151],
           [116, 125, 122],
           [136, 145, 142]],

          [[245, 246, 244],
           [236, 237, 235],
           [245, 246, 244],
           ...,
           [133, 129, 124],
           [125, 117, 110],
           [106,  98,  91]],

          ...,

          [[195, 171, 151],
           [184, 159, 139],
           [165, 138, 117],
           ...,
           [191, 205, 211],
           [226, 241, 244],
           [234, 246, 250]],

          [[213, 189, 169],
           [181, 158, 136],
           [169, 143, 119],
           ...,
           [212, 228, 235],
           [223, 239, 246],
           [191, 207, 214]],

          [[174, 151, 129],
           [133, 110,  88],
           [156, 130, 106],
           ...,
           [217, 234, 243],
```

```
from ultralytics import YOLO
import cv2

# Path to your trained model on Drive
model_path = '/content/drive/MyDrive/armhack/yolov8m_a100_run/weights/best.pt'

model = YOLO(model_path)
```

```
from ultralytics import YOLO
import cv2
import os

# Paths (CHANGE these if your folders differ)
model_path = '/content/drive/MyDrive/armhack/yolov8m_a100_run/weights/best.pt'
video_path = '/content/potholes (1).mp4'
output_path = '/content/drive/MyDrive/armhack/test_videos/road_test_detected.mp4'

# Sanity checks
assert os.path.exists(model_path), "Model path not found!"
assert os.path.exists(video_path), "Video path not found!"

model = YOLO(model_path)

cap = cv2.VideoCapture(video_path)

frame_width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
frame_height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
fps = cap.get(cv2.CAP_PROP_FPS) or 25  # fallback fps

fourcc = cv2.VideoWriter_fourcc(*'mp4v')
out = cv2.VideoWriter(output_path, fourcc, fps, (frame_width, frame_height))

print("Processing video and saving to Drive...")

frame_count = 0
while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        break

    results = model.predict(frame, conf=0.3, device=0)
```

```
        annotated_frame = results[0].plot()
        out.write(annotated_frame)
        frame_count += 1

cap.release()
out.release()

print(f"Done! {frame_count} frames processed.")
print("Saved to:", output_path)
```

```
Processing video and saving to Drive...

0: 288x640 1 pothole, 76.9ms
Speed: 1.8ms preprocess, 76.9ms inference, 1.4ms postprocess per image at shape (1, 3, 288, 640)

0: 288x640 2 potholes, 9.1ms
Speed: 2.0ms preprocess, 9.1ms inference, 1.3ms postprocess per image at shape (1, 3, 288, 640)

0: 288x640 1 pothole, 9.2ms
Speed: 1.8ms preprocess, 9.2ms inference, 1.2ms postprocess per image at shape (1, 3, 288, 640)

0: 288x640 1 pothole, 9.9ms
Speed: 1.9ms preprocess, 9.9ms inference, 1.2ms postprocess per image at shape (1, 3, 288, 640)

0: 288x640 1 pothole, 9.3ms
Speed: 1.8ms preprocess, 9.3ms inference, 1.3ms postprocess per image at shape (1, 3, 288, 640)

0: 288x640 1 pothole, 9.2ms
Speed: 1.8ms preprocess, 9.2ms inference, 1.2ms postprocess per image at shape (1, 3, 288, 640)

0: 288x640 1 pothole, 8.8ms
Speed: 1.8ms preprocess, 8.8ms inference, 1.2ms postprocess per image at shape (1, 3, 288, 640)

0: 288x640 1 pothole, 9.2ms
Speed: 1.8ms preprocess, 9.2ms inference, 1.2ms postprocess per image at shape (1, 3, 288, 640)

0: 288x640 1 pothole, 9.6ms
Speed: 1.9ms preprocess, 9.6ms inference, 1.2ms postprocess per image at shape (1, 3, 288, 640)

0: 288x640 1 pothole, 8.9ms
Speed: 1.7ms preprocess, 8.9ms inference, 1.2ms postprocess per image at shape (1, 3, 288, 640)

0: 288x640 1 pothole, 8.8ms
Speed: 1.6ms preprocess, 8.8ms inference, 1.2ms postprocess per image at shape (1, 3, 288, 640)

0: 288x640 1 pothole, 8.9ms
Speed: 1.7ms preprocess, 8.9ms inference, 1.2ms postprocess per image at shape (1, 3, 288, 640)

0: 288x640 1 pothole, 10.3ms
Speed: 1.7ms preprocess, 10.3ms inference, 1.3ms postprocess per image at shape (1, 3, 288, 640)

0: 288x640 1 pothole, 9.2ms
Speed: 1.7ms preprocess, 9.2ms inference, 1.2ms postprocess per image at shape (1, 3, 288, 640)

0: 288x640 1 pothole, 9.1ms
Speed: 1.8ms preprocess, 9.1ms inference, 1.2ms postprocess per image at shape (1, 3, 288, 640)

0: 288x640 1 pothole, 9.3ms
Speed: 1.7ms preprocess, 9.3ms inference, 1.3ms postprocess per image at shape (1, 3, 288, 640)

0: 288x640 2 potholes, 9.2ms
Speed: 1.8ms preprocess, 9.2ms inference, 1.2ms postprocess per image at shape (1, 3, 288, 640)

0: 288x640 1 pothole, 8.8ms
Speed: 1.8ms preprocess, 8.8ms inference, 1.2ms postprocess per image at shape (1, 3, 288, 640)

0: 288x640 3 potholes, 9.0ms
Speed: 1.7ms preprocess, 9.0ms inference, 1.2ms postprocess per image at shape (1, 3, 288, 640)
```