pedestrian-detection

November 7, 2025

[3]:
```
# Cell 1: Install required libraries
!pip install ultralytics opencv-python
```

```
Requirement already satisfied: ultralytics in /usr/local/lib/python3.12/dist-
packages (8.3.225)
Requirement already satisfied: opencv-python in /usr/local/lib/python3.12/dist-
packages (4.12.0.88)
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: 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) (2.8.0+cu126)
Requirement already satisfied: torchvision>=0.9.0 in
/usr/local/lib/python3.12/dist-packages (from ultralytics) (0.23.0+cu126)
Requirement already satisfied: psutil in /usr/local/lib/python3.12/dist-packages
(from ultralytics) (5.9.5)
Requirement already satisfied: polars in /usr/local/lib/python3.12/dist-packages
(from ultralytics) (1.25.2)
Requirement already satisfied: ultralytics-thop>=2.0.18 in
/usr/local/lib/python3.12/dist-packages (from ultralytics) (2.0.18)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.12/dist-packages (from matplotlib>=3.3.0->ultralytics)
(1.3.3)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.12/dist-
packages (from matplotlib>=3.3.0->ultralytics) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.12/dist-packages (from matplotlib>=3.3.0->ultralytics)
(4.60.1)
Requirement already satisfied: kiwisolver>=1.3.1 in
```

```
/usr/local/lib/python3.12/dist-packages (from matplotlib>=3.3.0->ultralytics)
(1.4.9)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.12/dist-packages (from matplotlib>=3.3.0->ultralytics)
(25.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.12/dist-packages (from matplotlib>=3.3.0->ultralytics)
(3.2.5)
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.12/dist-packages (from matplotlib>=3.3.0->ultralytics)
(2.9.0.post0)
Requirement already satisfied: charset_normalizer<4,>=2 in
/usr/local/lib/python3.12/dist-packages (from requests>=2.23.0->ultralytics)
(3.4.4)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-
packages (from requests>=2.23.0->ultralytics) (3.11)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.12/dist-packages (from requests>=2.23.0->ultralytics)
(2.5.0)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.12/dist-packages (from requests>=2.23.0->ultralytics)
(2025.10.5)
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-
packages (from torch>=1.8.0->ultralytics) (3.20.0)
Requirement already satisfied: typing-extensions>=4.10.0 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics)
(4.15.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-
packages (from torch>=1.8.0->ultralytics) (75.2.0)
Requirement already satisfied: sympy>=1.13.3 in /usr/local/lib/python3.12/dist-
packages (from torch>=1.8.0->ultralytics) (1.13.3)
Requirement already satisfied: networkx in /usr/local/lib/python3.12/dist-
packages (from torch>=1.8.0->ultralytics) (3.5)
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 in /usr/local/lib/python3.12/dist-packages
(from torch>=1.8.0->ultralytics) (2025.3.0)
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.6.77 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics)
(12.6.77)
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.6.77 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics)
(12.6.77)
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.6.80 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics)
(12.6.80)
Requirement already satisfied: nvidia-cudnn-cu12==9.10.2.21 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics)
```

```
(9.10.2.21)
Requirement already satisfied: nvidia-cublas-cu12==12.6.4.1 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics)
(12.6.4.1)
Requirement already satisfied: nvidia-cufft-cu12==11.3.0.4 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics)
(11.3.0.4)
Requirement already satisfied: nvidia-curand-cu12==10.3.7.77 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics)
(10.3.7.77)
Requirement already satisfied: nvidia-cusolver-cu12==11.7.1.2 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics)
(11.7.1.2)
Requirement already satisfied: nvidia-cusparse-cu12==12.5.4.2 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics)
(12.5.4.2)
Requirement already satisfied: nvidia-cusparselt-cu12==0.7.1 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics) (0.7.1)
Requirement already satisfied: nvidia-nccl-cu12==2.27.3 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics)
(2.27.3)
Requirement already satisfied: nvidia-nvtx-cu12==12.6.77 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics)
(12.6.77)
Requirement already satisfied: nvidia-nvjitlink-cu12==12.6.85 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics)
(12.6.85)
Requirement already satisfied: nvidia-cufile-cu12==1.11.1.6 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics)
(1.11.1.6)
Requirement already satisfied: triton==3.4.0 in /usr/local/lib/python3.12/dist-
packages (from torch>=1.8.0->ultralytics) (3.4.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-
packages (from python-dateutil>=2.7->matplotlib>=3.3.0->ultralytics) (1.17.0)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
/usr/local/lib/python3.12/dist-packages (from
sympy>=1.13.3->torch>=1.8.0->ultralytics) (1.3.0)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.12/dist-packages (from jinja2->torch>=1.8.0->ultralytics)
(3.0.3)
```

[4]:
```python
# Cell 2: Imports and Configuration

import cv2
import numpy as np
from ultralytics import YOLO
import time
```

```
from IPython.display import display, Image # Used for displaying output in␣
 ↪Jupyter

# --- Configuration ---
# Load a pre-trained YOLOv8 model (using 'n' for speed)
model = YOLO('yolov8n.pt')

# Class ID for 'person' in COCO dataset is 0
PERSON_CLASS_ID = 0
CONFIDENCE_THRESHOLD = 0.5
```

```
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.ultralytics.com/quickstart/#ultralytics-settings.
Downloading
https://github.com/ultralytics/assets/releases/download/v8.3.0/yolov8n.pt to
'yolov8n.pt': 100%        6.2MB 76.4MB/s 0.1s
```

[5]:
```
# Cell 3: Main Detection and Processing Function

def process_video_for_pedestrians(video_path='path/to/your/input_video.mp4',␣
 ↪output_path='output_pedestrian_detection.mp4', display_output=False):
    """
    Processes a video file to detect pedestrians and saves the output.

    Args:
        video_path (str or int): Path to the video file or 0 for webcam.
        output_path (str): Path to save the processed video.
        display_output (bool): If True, attempts to display frames live (best␣
 ↪for short clips or file saving).
    """

    # 1. Initialize Video Capture and Writer
    cap = cv2.VideoCapture(video_path)
    if not cap.isOpened():
        print(f"Error: Could not open video source {video_path}")
        return

    # Get video properties for output
    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)

    # Define the codec and create VideoWriter object
```

```
    # FourCC codes: 'mp4v' or 'XVID' are common
    fourcc = cv2.VideoWriter_fourcc(*'mp4v')
    out = cv2.VideoWriter(output_path, fourcc, fps, (frame_width, frame_height))

    frame_count = 0
    start_time = time.time()

    # 2. Main Processing Loop
    while cap.isOpened():
        success, frame = cap.read()
        if not success:
            break

        # Run YOLOv8 inference (no stream=True for single frames)
        results = model(frame, verbose=False)

        # Process results
        if results:
            r = results[0]

            boxes = r.boxes.xyxy.cpu().numpy().astype(int)
            confs = r.boxes.conf.cpu().numpy()
            classes = r.boxes.cls.cpu().numpy()

            # Iterate through detections and draw
            for box, conf, cls in zip(boxes, confs, classes):
                if cls == PERSON_CLASS_ID and conf >= CONFIDENCE_THRESHOLD:
                    x1, y1, x2, y2 = box
                    label = f'Pedestrian: {conf:.2f}'

                    # Draw bounding box (Green BGR)
                    cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 2)

                    # Draw label
                    cv2.putText(frame, label, (x1, y1 - 10),
                                cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)

        # 3. Write Frame to Output Video
        out.write(frame)

        # Optional: Display the frame (Note: live display can be slow/buggy in␣
↪Jupyter)
        if display_output:
            # Note: If running in Google Colab, use 'from google.colab.patches␣
↪import cv2_imshow'
            # and replace cv2.imshow with cv2_imshow.
            cv2.imshow("Detection Output", frame)
```

```
            if cv2.waitKey(1) & 0xFF == ord('q'):
                break

        frame_count += 1

    # 4. Cleanup
    end_time = time.time()

    cap.release()
    out.release()
    cv2.destroyAllWindows()

    processing_time = end_time - start_time
    print("-" * 50)
    print(f" Processing complete.")
    print(f"Total frames processed: {frame_count}")
    print(f"Total time taken: {processing_time:.2f} seconds")
    print(f"Output saved to: {output_path}")
    print("-" * 50)
```

[6]:
```
# Cell 4: Execute the function

# !!! CHANGE THIS PATH to your video file or use 0 for webcam !!!
input_video_path = 'test_video.mp4'

# Set the output file name
output_video_path = 'yolov8_pedestrian_output.mp4'

# Start the detection process
process_video_for_pedestrians(
    video_path=input_video_path,
    output_path=output_video_path,
    display_output=False # Set to True ONLY if running locally and want to see␣
 ↪a pop-up window
)
```

```
--------------------------------------------------
 Processing complete.
Total frames processed: 275
Total time taken: 55.58 seconds
Output saved to: yolov8_pedestrian_output.mp4
--------------------------------------------------
```

[7]:
```
# Cell 5: Display a message about the output file

print(f"Your processed video is saved as {output_video_path}")
print("You can download and view this file to see the detection results.")
```

```
Your processed video is saved as yolov8_pedestrian_output.mp4
You can download and view this file to see the detection results.
```

[8]:
```python
from google.colab import files

output_video_path = 'yolov8_pedestrian_output.mp4'
files.download(output_video_path)
```

```
<IPython.core.display.Javascript object>
```

```
<IPython.core.display.Javascript object>
```

[9]:
```python
# Zip the output video file
output_video_path = 'yolov8_pedestrian_output.mp4'
zip_file_name = 'yolov8_pedestrian_output.zip'

!zip -r {zip_file_name} {output_video_path}

print(f"Video file '{output_video_path}' has been zipped to '{zip_file_name}'")
```

```
  adding: yolov8_pedestrian_output.mp4 (deflated 2%)
Video file 'yolov8_pedestrian_output.mp4' has been zipped to
'yolov8_pedestrian_output.zip'
```

After running this, you can download the `.zip` file using the same `files.download()` method or through the file browser.