Project Name- Object Detection in Images

Submitted by-

- Ansh Mishra (202410116100030)
- Abhishek Sharma (202410116100009)
- Ankit Sisodia (202410116100029)
- Ashutosh Singh (202410116100043)

Code

```
# ===== STEP 1: Install Ultralytics YOLO =====
!pip install -q ultralytics

# ===== STEP 2: Import Libraries =====
from ultralytics import YOLO
import cv2
import numpy as np
import os
from PIL import Image
from google.colab import files
import uuid
import matplotlib.pyplot as plt

# ===== STEP 3: Prepare Folders =====
os.makedirs("uploads", exist_ok=True)
os.makedirs("outputs", exist_ok=True)
```

```
# ===== STEP 4: Upload Images =====
uploaded = files.upload()
image_paths = []
for filename in uploaded.keys():
    new_path = os.path.join("uploads", filename.replace(" ", "_"))
    os.rename(filename, new_path)
    image_paths.append(new_path)

# ===== STEP 5: Load YOLO Model (v8) =====
model = YOLO("yolov8n.pt")  # More stable on Colab

# ===== STEP 6: Object Detection Function =====
def detect_and_save(image_path):
    results = model(image_path)
    result = results[0]
    img = cv2.imread(image_path)

    for box in result.boxes:
        x1, y1, x2, y2 = map(int, box.xyxy[0])
        conf = float(box.conf[0])
        cls = int(box.cls[0])
        label = model.names[cls]
        cv2.rectangle(img, (x1, y1), (x2, y2), (0,255,0), 2)
        cv2.putText(img, f'{label} {conf:.2f}', (x1, y1 - 10), cv2.FONT_HERSHEY_SIMPLEX,
0.5, (0,255,0), 2)
```

```
    out_path = f'outputs/detected_{uuid.uuid4().hex[:6]}.jpg'
    cv2.imwrite(out_path, img)
    return out_path

# ===== STEP 7: Run Detection and Show Results =====
for path in image_paths:
    out_img = detect_and_save(path)
    display(Image.open(out_img))
    files.download(out_img)

print("✅ Done!")
```

Explaination

1. **Install Ultralytics YOLO Library**

- Installs the ultralytics package required to run YOLOv8.

2. **Import Libraries**

- Loads necessary libraries like YOLO, cv2, os, PIL, etc., for detection, image handling, and file operations.

3. **Create Folders**

- Makes two folders:
    - uploads for input images
    - outputs for storing results

4. **Upload Images**

- Uses files.upload() to upload images via Google Colab.
- Renames files (removes spaces) and stores them in the uploads folder.

5. **Load YOLOv8 Model**

- Loads the pre-trained yolov8n.pt model (YOLOv8 Nano – lightweight and fast).

6. **Object Detection Function**

- Reads each image.
- Detects objects using the YOLO model.
- Draws bounding boxes and labels with confidence scores.
- Saves the annotated image to the outputs folder.

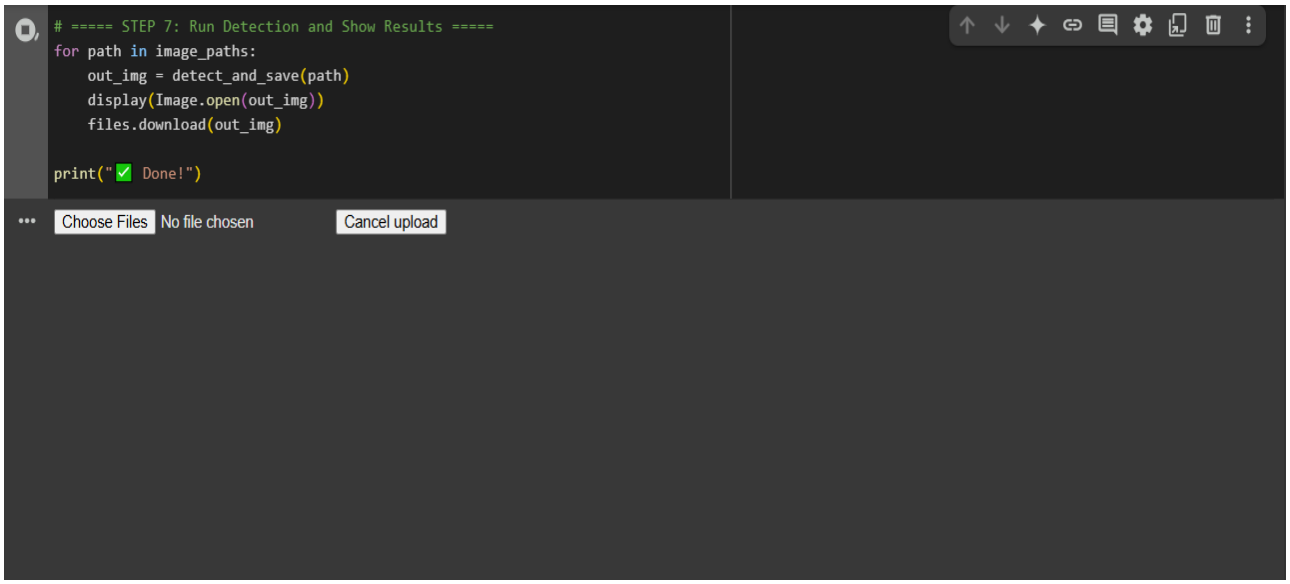
7. **Run Detection and Show Results**

- Applies the detection function to each uploaded image.
- Displays the result using PIL.Image.open().
- Allows the user to download the output image.

8. **Final Output**

- The message "✅ Done!" confirms successful processing.

Output



Option for Uploading Image File

Detected Objects from Image gets Downloaded