EXPERIMENT-6

Object detection and Recognition on available online image datasets using YOLO Model

Overview

Task	Tool/Model	Framework
Object Detection	YOLOv5 / YOLOv8	PyTorch / Ultralytics
Dataset	COCO, Pascal VOC, Open Images, or custom	
Language	Python	
Framework	Ultralytics / OpenCV	

Step-by-Step Implementation using YOLOv5/YOLOv8

1. Install YOLO (via Ultralytics)

```
bash
pip install ultralytics
```

Test it:

```
python
from ultralytics import YOLO
model = YOLO('yolov8n.pt')  # Load pretrained YOLOv8 nano
```

2. Download a Dataset

You can use:

- **COCO**: https://cocodataset.org
- **Pascal VOC**: https://host.robots.ox.ac.uk/pascal/VOC/
- **Open Images Dataset (OIDv4)**: https://storage.googleapis.com/openimages/web/index.html

Or you can load images directly from the web or Google Drive.

3. Run Detection on Dataset

```
python
from ultralytics import YOLO
import os
from PIL import Image
```

```
import matplotlib.pyplot as plt

# Load YOLO model
model = YOLO("yolov8n.pt")  # Use 'yolov8s.pt', 'yolov8m.pt', etc. for better accuracy

# Inference on one image
results = model("path/to/image.jpg", save=True)

# Show results
results[0].show()
```

4. Run on a Directory of Images

```python
results = model.predict(source="path/to/images/", save=True, conf=0.25)
```

5. Use Built-in Validation on Datasets

If you have COCO or VOC format:

```bash
yolo task=detect mode=val model=yolov8n.pt data=coco128.yaml
```

Or for Pascal VOC:

```bash
yolo task=detect mode=val model=yolov8n.pt data=VOC.yaml
```

6. Custom Dataset Training (Optional)

If you're using your **own labeled dataset**, structure it like:

```kotlin
datasets/
  images/
    train/
    val/
  labels/
    train/
    val/
```

Create `data.yaml` file:

```yaml
path: /path/to/dataset
train: images/train
val: images/val

names:
  0: person
  1: car
  2: dog
```

Train with:

```bash
yolo task=detect mode=train model=yolov8n.pt data=data.yaml epochs=50 imgsz=640
```

Example Use-Case: Run YOLO on Online Image (from URL)

```python
import requests
import cv2
import numpy as np
from ultralytics import YOLO

url = 'https://ultralytics.com/images/bus.jpg'
resp = requests.get(url, stream=True).raw
image = np.asarray(bytearray(resp.read()), dtype="uint8")
image = cv2.imdecode(image, cv2.IMREAD_COLOR)

cv2.imwrite('online_image.jpg', image)

model = YOLO('yolov8n.pt')
results = model('online_image.jpg', save=True)
results[0].show()
```

Visualization and Output

YOLO returns:

- Bounding boxes
- Class names and confidences
- Can save annotated images and export `.txt` or `.json` for further use.

Example Datasets You Can Use

Dataset	Task	Download Link
COCO128	Object Detection	`ultralytics/coco128.yaml` (comes pre-bundled)
Pascal VOC	Object Detection	https://host.robots.ox.ac.uk/pascal/VOC/
OpenImages	Detection/Segmentation	https://storage.googleapis.com/openimages