Experiment 7: Train an SSD Network for Object Detection in a Self-Driving Car Application

Date: 3/4/25

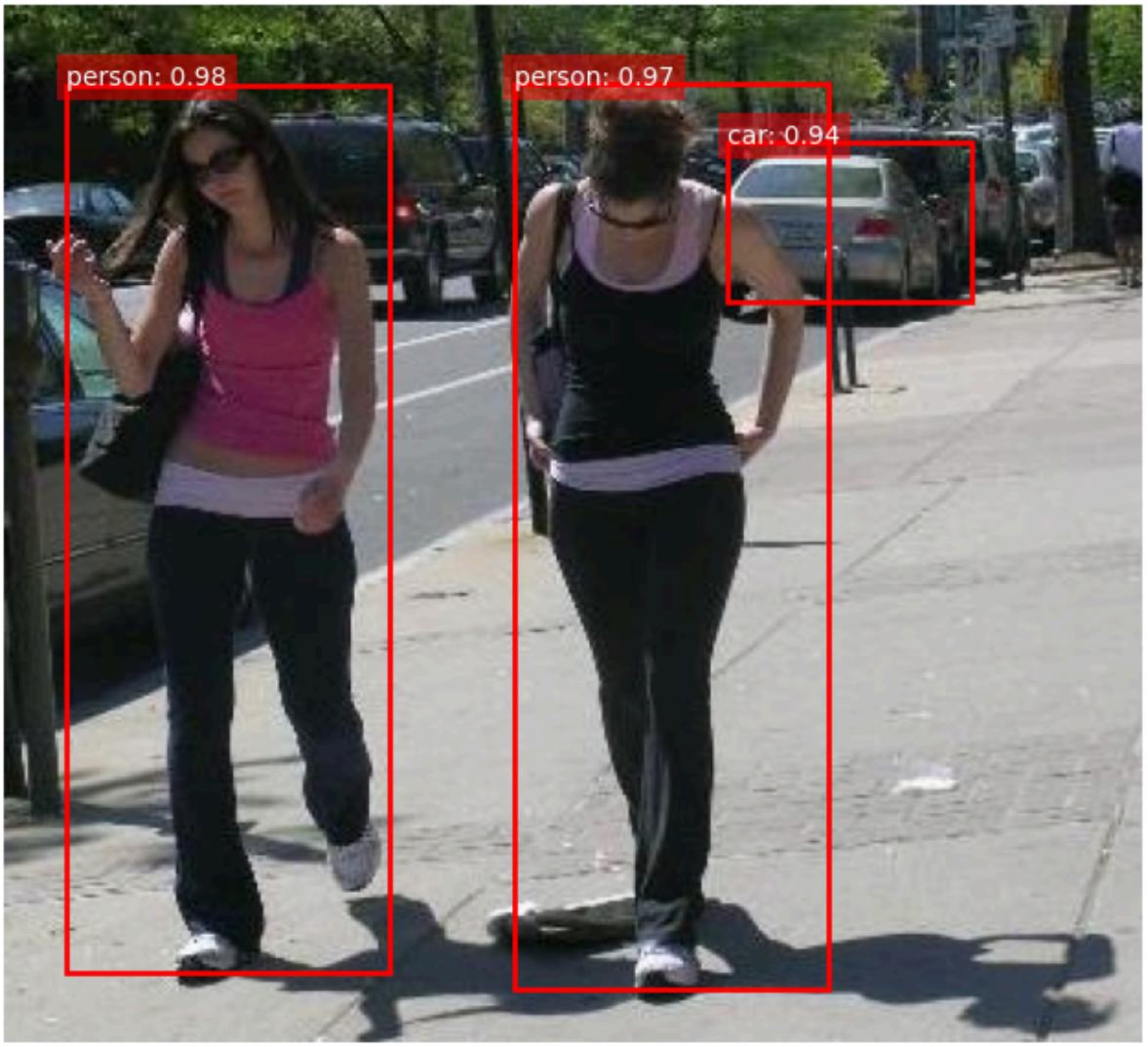
Aim:
To train an SSD (Single Shot Multibox Detector) network using a pretrained model for object detection in a self-driving car application.

Code:

```
import torch
from torchvision.models.detection import ssdlite320_mobilenet_v3_large
from torchvision.transforms import functional as F
from PIL import Image
import matplotlib.pyplot as plt
import torchvision

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

# Load pretrained SSD model
model = ssdlite320_mobilenet_v3_large(pretrained=True)
model.eval().to(device)

# Replace with the filename you uploaded
image_path = "/content/PennPed00053.png"

image = Image.open(image_path).convert("RGB")
image_tensor = F.to_tensor(image).unsqueeze(0).to(device)

# Predict
with torch.no_grad():
    prediction = model(image_tensor)[0]

def plot_boxes(image, prediction, score_thresh=0.5):
    plt.figure(figsize=(8,8))
    plt.imshow(image)
    ax = plt.gca()

    # COCO classes (replace with the correct class names if needed)
    COCO_CLASSES = [
        '__background__', 'person', 'bicycle', 'car', 'motorcycle', 'airplane', 'bus',
        'train', 'truck', 'boat', 'traffic light', 'fire hydrant', 'N/A', 'stop sign',
        'parking meter', 'bench', 'bird', 'cat', 'dog', 'horse', 'sheep', 'cow',
        'elephant', 'bear', 'zebra', 'giraffe', 'N/A', 'backpack', 'umbrella', 'N/A', 'N/A',
        'handbag', 'tie', 'suitcase', 'frisbee', 'skis', 'snowboard', 'sports ball',
        'kite', 'baseball bat', 'baseball glove', 'skateboard', 'surfboard', 'tennis racket',
        'bottle', 'N/A', 'wine glass', 'cup', 'fork', 'knife', 'spoon', 'bowl',
        'banana', 'apple', 'sandwich', 'orange', 'broccoli', 'carrot', 'hot dog', 'pizza',
        'donut', 'cake', 'chair', 'couch', 'potted plant', 'bed', 'N/A', 'dining table',
        'N/A', 'N/A', 'toilet', 'N/A', 'tv', 'laptop', 'mouse', 'remote', 'keyboard', 'cell phone',
        'microwave', 'oven', 'toaster', 'sink', 'refrigerator', 'N/A', 'book', 'clock',
        'vase', 'scissors', 'teddy bear', 'hair drier', 'toothbrush'
```

```
    ]

    for box, label, score in zip(prediction['boxes'], prediction['labels'], prediction['scores']):
        if score > score_thresh:
            x1, y1, x2, y2 = box
            ax.add_patch(plt.Rectangle((x1, y1), x2-x1, y2-y1,
                                       fill=False, color='red', linewidth=2))
            # Get class name from COCO_CLASSES list
            class_name = COCO_CLASSES[label]
            ax.text(x1, y1, f'{class_name}: {score:.2f}', color='white', # Display class name
                    bbox=dict(facecolor='red', edgecolor='none', alpha=0.5))
    plt.axis("off")
    plt.show()

plot_boxes(image, prediction)
```

OUTPUT:



Result:
The SSD model successfully detected objects in the uploaded image. Bounding boxes are drawn around detected objects with their class names and prediction scores.