Exp .no 8 Title: A PyTorch implementation of Object Detection with Single Shot Detector

---

Aim:

To perform object detection on an uploaded image using a pretrained SSD300 model with a VGG16 backbone and visualize the detected objects.

Procedure:

1. **Install and Import Necessary Libraries:**
    - **Install `torch`, `torchvision`, `opencv-python`, and `matplotlib` packages.**
    - **Import PyTorch, Torchvision models, OpenCV, PIL, and matplotlib for deep learning and visualization.**
2. **Load the Pretrained SSD Model:**
    - **Load the `ssd300_vgg16` model with pretrained weights from `torchvision.models.detection`.**
    - **Set the model to evaluation mode to deactivate training behaviors like dropout.**
3. **Define Image Transformations:**
    - **Use the transforms associated with the model's weights to resize, normalize, and prepare the image for inference.**
4. **Upload and Preprocess the Image:**
    - **Upload an external image using Google Colab's file upload tool.**
    - **Open the image using PIL, convert it to RGB format, and apply the defined transformations.**
5. **Make Predictions:**

- **Pass the preprocessed image through the SSD model to obtain bounding boxes, labels, and confidence scores.**

6. **Visualize Predictions:**
    - **Draw bounding boxes around detected objects.**
    - **Annotate each box with its label and confidence score using OpenCV.**
    - **Display the annotated image in the notebook.**

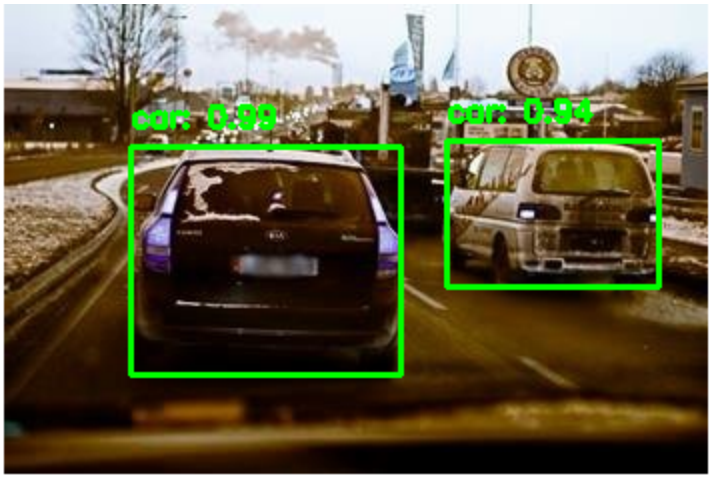
Code:

```
# Step 1: Install dependencies (if needed)

!pip install -q torch torchvision matplotlib opencv-python


# Step 2: Import libraries

import torch

import torchvision

import torchvision.transforms as T

from PIL import Image

import matplotlib.pyplot as plt

import cv2

import numpy as np

from torchvision.models.detection.ssd import SSD300_VGG16_Weights

from google.colab.patches import cv2_imshow  # For displaying images in Colab
```

```
# Step 3: Load the SSD model with pretrained weights
weights = SSD300_VGG16_Weights.DEFAULT
model = torchvision.models.detection.ssd300_vgg16(weights=weights)
model.eval()

# Step 4: Define the transform
transform = weights.transforms()

# Step 5: Upload and load image
from google.colab import files
uploaded = files.upload()

image_path = list(uploaded.keys())[0]
image = Image.open(image_path).convert("RGB")
img_tensor = transform(image).unsqueeze(0)

# Step 6: Predict
with torch.no_grad():
    preds = model(img_tensor)[0]

# Step 7: Visualize predictions
def draw_boxes(image_pil, predictions, score_threshold=0.5):
    image = np.array(image_pil)
```

```
    boxes = predictions['boxes']

    labels = predictions['labels']

    scores = predictions['scores']

    categories = weights.meta["categories"]


    for box, label, score in zip(boxes, labels, scores):

        if score >= score_threshold:

            x1, y1, x2, y2 = box.int().tolist()

            cv2.rectangle(image, (x1, y1), (x2, y2), color=(0,255,0),
thickness=2)

            text = f"{categories[label]}: {score:.2f}"

            cv2.putText(image, text, (x1, y1 - 10),
cv2.FONT_HERSHEY_SIMPLEX,

                        0.5, (0, 255, 0), 2)


    cv2_imshow(image)


draw_boxes(image, preds)
```

Output:



Result:

The pretrained SSD300 model successfully detected multiple objects in the uploaded image, providing accurate localization and classification with confidence scores.