```
!pip install torch torchvision
```

```
Requirement already satisfied: torch in /usr/local/lib/python3.10/dist-packages (2.5.1+cu121)
Requirement already satisfied: torchvision in /usr/local/lib/python3.10/dist-packages (0.20.1+cu121)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from torch) (3.16.1)
Requirement already satisfied: typing-extensions>=4.8.0 in /usr/local/lib/python3.10/dist-packages (from torch) (4.12.2)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch) (3.4.2)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from torch) (3.1.4)
Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages (from torch) (2024.10.0)
Requirement already satisfied: sympy==1.13.1 in /usr/local/lib/python3.10/dist-packages (from torch) (1.13.1)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from sympy==1.13.1->torch) (1.3.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from torchvision) (1.26.4)
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in /usr/local/lib/python3.10/dist-packages (from torchvision) (11.0.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->torch) (3.0.2)
```

```
!pip install tensorflow
```

Show hidden output

```
!git clone https://github.com/AarohiSingla/SSD-Tensorflow-On-Custom-Dataset.git
%cd SSD-Tensorflow-On-Custom-Dataset
```

```
Cloning into 'SSD-Tensorflow-On-Custom-Dataset'...
remote: Enumerating objects: 221, done.
remote: Counting objects: 100% (219/219), done.
remote: Compressing objects: 100% (166/166), done.
remote: Total 221 (delta 52), reused 213 (delta 47), pack-reused 2 (from 1)
Receiving objects: 100% (221/221), 96.42 MiB | 29.54 MiB/s, done.
Resolving deltas: 100% (52/52), done.
/content/simple-ssd-for-beginners/SSD-Tensorflow-On-Custom-Dataset
```

```
# Example configuration changes in config.py
VOC_ROOT = '/content/CHESS_SSD'  # Change this to your dataset path
NUM_CLASSES = 2  # Set this according to your dataset (e.g., if you have two classes)
```

```
!python train_ssd_network.py --dataset_name=pascalvoc_2007 --dataset_split_name=train --model_name=ssd_300_vgg --save_summaries_secs=60 -
```

```
2024-12-08 18:05:23.177334: E external/local_xla/xla/stream_executor/cuda/cuda_dnn.cc:9261] Unable to register cuDNN factory: Attemp
2024-12-08 18:05:23.177440: E external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:607] Unable to register cuFFT factory: Attempt
2024-12-08 18:05:23.191723: E external/local_xla/xla/stream_executor/cuda/cuda_blas.cc:1515] Unable to register cuBLAS factory: Atte
2024-12-08 18:05:23.220260: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use availabl
To enable the following instructions: AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
2024-12-08 18:05:24.793184: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
Traceback (most recent call last):
  File "/content/simple-ssd-for-beginners/SSD-Tensorflow-On-Custom-Dataset/train_ssd_network.py", line 19, in <module>
    from datasets import dataset_factory
  File "/content/simple-ssd-for-beginners/SSD-Tensorflow-On-Custom-Dataset/datasets/dataset_factory.py", line 24, in <module>
    from datasets import pascalvoc_2007
  File "/content/simple-ssd-for-beginners/SSD-Tensorflow-On-Custom-Dataset/datasets/pascalvoc_2007.py", line 21, in <module>
    from datasets import pascalvoc_common
  File "/content/simple-ssd-for-beginners/SSD-Tensorflow-On-Custom-Dataset/datasets/pascalvoc_common.py", line 23, in <module>
    slim = tf.contrib.slim
AttributeError: module 'tensorflow' has no attribute 'contrib'
```

```
!python eval_ssd_network.py --model_name=ssd_300_vgg --checkpoint_dir=/path/to/checkpoints --eval_dir=/path/to/eval_directory
```

```
python3: can't open file '/content/simple-ssd-for-beginners/SSD-Tensorflow-On-Custom-Dataset/eval_ssd_network.py': [Errno 2] No such
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
import tensorflow as tf

# Correct the model path to point to the SavedModel directory
model_path = '/content/ssd_mobilenet_v2_fpnlite_320x320_v1' # Changed to parent directory

# Load the model
model = tf.saved_model.load(model_path)
```

```
import tensorflow as tf
import cv2

def load_image(image_path):
    # Load image using OpenCV
    image = cv2.imread(image_path)
```

```
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)  # Convert BGR to RGB
    image = tf.convert_to_tensor(image)  # Convert to tensor
    image = tf.image.resize(image, (300, 300))  # Resize as per SSD input size
    image = tf.cast(image, tf.uint8)  # Cast to uint8
    return image[tf.newaxis, ...]  # Add batch dimension


import os

test_images_path = '/content/drive/My Drive/CHESS_SSD/images/'
results = []

for image_file in os.listdir(test_images_path):
    if image_file.endswith('.jpg'):
        image_path = os.path.join(test_images_path, image_file)
        input_tensor = load_image(image_path)

        # Run inference
        detections = model(input_tensor)

        # Process detections (you may want to filter based on confidence score)
        results.append((image_file, detections))
        print(f'Processed {image_file}')
```

Show hidden output

```
import matplotlib.pyplot as plt

def visualize_detections(image, detections):
    # Example visualization code (modify as needed based on output format)
    plt.imshow(image)
    plt.axis('off')
    plt.show()

for image_file, detections in results:
    image_path = os.path.join(test_images_path, image_file)
    image = cv2.imread(image_path)
    visualize_detections(image, detections)
```

Show hidden output

```
!pip install tensorflow-object-detection-api
```

Show hidden output

```
from object_detection.metrics import coco_evaluation

# Load your ground truth and predictions
groundtruth_annotations = '/content/CHESS_SSD/annotations'
predictions = '/content/CHESS_SSD/model_predictions'

# Assuming your predictions are in COCO format, you can initialize like this:
# coco_evaluator = coco_evaluation.CocoDetectionEvaluator(categories)  # Define your categories
# ... (Load predictions and ground truth into coco_evaluator) ...
# metrics = coco_evaluator.evaluate()
```

Show hidden output

Next steps: Explain error

Double-click (or enter) to edit