

A PyTorch implementation of Object Detection with Single Shot Detector

1. Import Required Libraries

Import necessary libraries:

- `torch` and `torchvision` for deep learning and model loading.
- `PIL` for image loading and manipulation.
- `matplotlib.pyplot` and `matplotlib.patches` for displaying results.
- `google.colab.files` for uploading images in Colab.

2. Load Pre-trained SSD300 VGG16 Model

Load the `ssd300_vgg16` pretrained model from `torchvision` and set it to evaluation mode (`.eval()`).

Move the model to GPU if available, otherwise, it will run on the CPU.

3. Upload Image

Use `files.upload()` from `google.colab` to allow users to upload an image. After uploading, extract the filename of the image.

4. Define Preprocessing and Load Image

- Convert the uploaded image to RGB format using `PIL`.
- Apply preprocessing steps: resizing the image to 300x300, converting it to a tensor, and normalizing it using ImageNet's standard mean and standard deviation.
- Return the original and preprocessed tensor.

5. Run Inference

Pass the preprocessed image tensor through the SSD model for inference using `torch.no_grad()` (to disable gradient calculation). The model outputs the detected bounding boxes, labels, and scores.

6. Visualize Detections

Define a function `visualize_detections` to visualize the bounding boxes and labels:

- **Plot the image.**
- **Loop through the detections and filter out those with low confidence (below the threshold, 0.5).**
- **Draw a bounding box for each detected object with a label and score above the threshold.**
- **Use matplotlib to display the image with bounding boxes and labels.**

7. Show Results

Finally, call the `visualize_detections()` function to display the image with detected objects.