Train an SSD network in a self-driving car application

1. **Import Required Libraries**
   Load essential libraries like `torch`, `torchvision`, `PIL`, `cv2`, `numpy`, and `requests` for deep learning, image processing, and I/O.

2. **Define COCO Class Labels**
   Specify class names from the COCO dataset, which the SSD model uses for labeling detected objects.

3. **Load Pretrained SSD300 VGG16 Model**
   Load the pretrained `ssd300_vgg16` model from `torchvision` and switch it to evaluation mode (`model.eval()`).

4. **Prepare Image Transformations**
   Define a transformation pipeline using `torchvision.transforms` to resize and convert the image to tensor format expected by the model.

5. **Load Image from URL or Local Path**
   Attempt to load the image using `requests` from a URL. If it fails, fallback to a local image path using `PIL.Image.open`.

6. **Preprocess Image for SSD Input**
   Resize the image to 300x300, convert it to a tensor, and add a batch dimension before passing it to the model.

7. **Run Inference with the Model**
   Use `torch.no_grad()` for inference and get predictions like bounding boxes, labels, and confidence scores.

8. **Filter and Rescale Detections**
   Filter predictions by confidence threshold (>0.5) and rescale bounding boxes from model input size to original image dimensions.

9. **Draw Bounding Boxes and Labels**
   For each detected object (e.g., car, person), draw a bounding box and add the label with confidence score on the original image using `cv2`.

10. **Display the Result**
    Convert the image back to RGB and display it using `matplotlib.pyplot` without axes.