📦 Faster R-CNN Object Detection on Pascal VOC

This project implements a custom object detection pipeline using PyTorch and torchvision. It leverages **Faster R-CNN** with a **MobileNetV2** backbone for lightweight performance and is trained on the **Pascal VOC 2012** dataset.

🧠 Features

- Custom Faster R-CNN model with MobileNetV2 backbone.
- Trained on VOC 2012 with optional download.
- Image visualization with predicted bounding boxes.
- GPU-compatible training and evaluation.
- Smoothed training statistics and loss tracking.

📁 Project Structure

bash

CopyEdit

```
├── app.py              # Main training & evaluation script
├── fasterrcnn_model.pth  # Saved model after training (generated)
├── requirements.txt      # Python dependencies
├── README.md             # Project documentation
```

🔧 Requirements

Install all dependencies using:

bash

CopyEdit

```
pip install -r requirements.txt
```

🚀 How to Run

1. **Clone the repository** (or copy files to your local system).
2. **Run the training script:**

bash

CopyEdit

python app.py

This will:

- Download the Pascal VOC dataset (if not already present).
- Train the Faster R-CNN model.
- Save the model to fasterrcnn_model.pth.
- Display predictions on a sample image.

⚠️ Modify path_to_VOC in the script to set your VOC dataset path.

🖼 Sample Output

After training, the script will show a sample image with predicted bounding boxes.

🧪 Inference on Custom Images

To use the trained model for your own images:

1. Load the model and weights (fasterrcnn_model.pth).
2. Preprocess your image with ToTensor.
3. Run model(image_tensor) and extract boxes, labels, and scores.

💻 Hardware Requirements

- Python 3.8+
- GPU Recommended (CUDA if available)

- ~8GB RAM minimum

📚 References

- Faster R-CNN Paper
- PyTorch Object Detection Tutorial
- Pascal VOC Dataset