Object Detection Internship Report

Name: Abhishek Choudhary
Project Title: Object Detection with Faster R-CNN using ResNet-50 FPN
Platform: Google Colab, PyTorch
Model Architecture: Faster R-CNN + ResNet-50 + FPN
Dataset: Pascal VOC 2007 (train_val split)

Project Overview

In this project, I developed a complete object detection pipeline using the **Faster R-CNN** architecture with a **ResNet-50** backbone and a **Feature Pyramid Network (FPN)** head. The model was trained on the **Pascal VOC 2007** dataset using PyTorch in a Google Colab environment.

The objective was to understand how region-based detectors work, explore modular training pipelines, and leverage **AI assistance** to streamline development, debug complex issues, and produce a high-quality deliverable.

Due to Colab's runtime limits, I trained the model for only 2 epochs during submission, but I later retrained it for 13 epochs, which significantly improved prediction accuracy and confidence.

Tech Stack & Methodology

- **Backbone Network:** ResNet-50 (pretrained on ImageNet)
- **Detection Framework:** Faster R-CNN via `torchvision.models.detection`
- **Head Layer:** Feature Pyramid Network (FPN)
- **Dataset:** Pascal VOC 2007 using `torchvision.datasets.VOCDetection`

Training Setup:

- **Epochs:** 2 (submitted), 13 (final result)
- **Optimizer:** Stochastic Gradient Descent (SGD) with momentum

- **Loss Functions:** RPN objectness loss, classification loss, bounding box regression loss
- **Evaluation:** Qualitative analysis using bounding box and label visualizations on test images

Why I Chose ResNet-50 as the Backbone

ResNet-50 is a powerful and well-established CNN architecture that balances performance and efficiency. Its **skip connections** (residual blocks) make it highly effective in avoiding vanishing gradients in deep networks. Since Faster R-CNN requires rich hierarchical feature maps for both small and large objects, ResNet-50 was a natural choice for its **robust feature extraction**, especially when pretrained on ImageNet.

Why I Chose FPN (Feature Pyramid Network)

FPN enhances object detection by enabling **multi-scale feature learning**. It combines high-resolution, low-semantic features with low-resolution, high-semantic features, allowing the detector to better identify small and large objects across the image. This was especially important in Pascal VOC, where object sizes vary greatly. FPN's structure improved localization and classification performance without significantly increasing computational cost.

Why I Chose Pascal VOC 2007

Pascal VOC 2007 is a widely-used benchmark dataset for object detection and provides a manageable yet diverse set of classes (20 in total). It's lightweight enough to train within Colab's resource constraints but still offers enough complexity for practical experimentation. The availability of prebuilt PyTorch utilities for loading and evaluating VOC datasets also accelerated development.

Role of AI (ChatGPT) in This Project

Throughout the project, **ChatGPT served as an essential assistant** in several critical areas. While the code itself was written manually, I consistently relied on ChatGPT to support and accelerate development. For example, when encountering issues such as **CUDA-related errors** or **tensor shape mismatches**, I used ChatGPT to pinpoint the root causes and implement correct fixes. This saved valuable debugging time.

Beyond debugging, ChatGPT was particularly helpful in **clarifying architectural concepts**, such as how **Region of Interest (RoI) Align** works, how **anchor generation** functions in Faster R-CNN, and how the different **loss components** (like objectness loss and box regression loss) interact during training. These insights helped deepen my understanding of the model's inner workings.

Additionally, I used ChatGPT as a **coding advisor** — for instance, when designing training loops or evaluating different configurations of the detection head. It also played a key role in **curating this final report**, offering assistance with structure, phrasing, and clarity. Furthermore, I leveraged ChatGPT to **quickly source and summarize relevant documentation**, which made the overall development process faster and more informed.

Importantly, my use of ChatGPT was focused on **learning, validating, and accelerating** rather than replacing personal effort. The collaboration helped me become more efficient while ensuring I retained full ownership of the implementation and analysis.

Model Architecture Breakdown

🔹 ResNet-50 + FPN (Backbone)

- Extracts multi-level feature maps from input images
- FPN integrates spatial and semantic features from various layers

🔹 Region Proposal Network (RPN)

- Scans the backbone’s feature maps
- Proposes regions likely to contain objects

🔹 RoI Align

- Converts proposed regions to fixed-size features
- Maintains alignment for spatial precision

🔹 Detection Head

- **Classifier**: predicts object classes
- **Box Regressor**: refines bounding box coordinates

This modular design allows flexible and accurate detection of multiple object types in a single image.

Development Tools

- **Google Colab:** Cloud-based training environment with GPU acceleration
- **PyTorch & torchvision:** Core deep learning library used to define models, datasets, and training pipelines
- **Matplotlib:** Used for plotting prediction vs. ground truth bounding boxes

Challenges & Solutions

Challenge	Resolution
Shape mismatch while plotting	Used `.detach().cpu().numpy()` to process tensors for visualization
False positives (e.g., detecting "chair" inaccurately)	Likely due to class imbalance and low training epochs
Visual comparison of predictions vs. ground truth	Added dual-colored bounding boxes and labels for clarity
Runtime limitations	Submitted results after 2 epochs, but improved performance after running 13

Key Learnings

- Understood how **region proposal-based detectors** work end to end
- Learned the impact of **FPN in multi-scale object localization**
- Discovered how training duration and class distribution affect accuracy

- Improved visualization techniques to debug model behavior
- Learned to **collaborate with AI** (ChatGPT) for debugging and research acceleration

Reflections & Takeaways

- **Training Time Matters**: The performance difference between 2 and 13 epochs was dramatic
- **Good Backbones Accelerate Learning**: ResNet-50 with FPN provided strong results early
- **AI Amplifies Productivity**: ChatGPT was a powerful partner — not just a shortcut
- **Visualization Is Vital**: Debugging without visual tools is like flying blind

Note on Training Constraints

One of the major challenges I faced during this assignment was the **lack of GPU capability** in my local environment. Since the model was relatively heavy and resource-intensive (Faster R-CNN with ResNet-50 and FPN), training on CPU was extremely slow and often impractical.

To make things more difficult, I also encountered **frequent disconnections and errors** on Google Colab, which caused repeated interruptions during the training process. For nearly two full days, I attempted to train the model — but each run either **crashed midway due to session timeouts**, **ran out of memory**, or **lost progress due to internet issues**.

As a result, I was only able to complete **2 full epochs of training** in time for submission. While I later retried with 13 epochs (which gave visibly better results), these runs could not be saved or shared due to session expiration and resource limits. These issues significantly limited my ability to fine-tune or experiment further.

Despite these setbacks, I focused on understanding the architecture, building a working pipeline, and documenting the process as clearly as possible.

Conclusion

This assignment gave me real, hands-on experience with object detection using a production-ready architecture. I now feel more confident navigating PyTorch's detection modules, tuning model parameters, and leveraging pretrained backbones effectively. Tools like ChatGPT didn't replace my learning — they **enhanced it**.

Thank you for this opportunity. I look forward to contributing more with stronger skills and greater clarity.