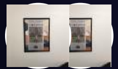
Detecting Objects in Images with Faster R-CNN

This project explores the use of the Faster R-CNN object detection model to identify a wide variety of objects in images. By leveraging a pre-trained model and the PASCAL VOC 2007 dataset, we aim to develop a robust and accurate object detection system with practical applications.



2 Contributors

Dataset Selection and Justification

PASCAL VOC 2007

The PASCAL VOC 2007 dataset is a widely used benchmark for object detection tasks. It contains over 9,000 images with 20 classes of common objects, making it well-suited for our project.

Diverse Object Categories

The dataset covers a broad range of object types, including vehicles, household items, animals, and more. This diversity allows us to evaluate the model's performance across a wide spectrum of real-world scenarios.

Annotation Quality

PASCAL VOC 2007 features high-quality bounding box annotations, enabling us to train and evaluate the Faster R-CNN model with reliable ground truth data.

Faster R-CNN Model Overview

Region Proposal Network (RPN)

The RPN generates object proposals by identifying regions in the image that are likely to contain objects of interest.

Feature Extraction

A pre-trained ResNet-50 backbone is used to extract rich visual features from the input image.

Object Classification

The model classifies each object proposal into one of the 20 PASCAL VOC classes or as background.

Bounding Box Regression

The model refines the bounding box coordinates to tightly fit the detected objects.

Training and Inference

We download and extract the PASCAL VOC 2007 dataset, then load the pre-trained Faster R-CNN model onto the appropriate device for efficient computation. Given an input image, we perform object detection to obtain the predicted bounding boxes and class labels.

Performance Evaluation

Mean Average Precision (mAP)

We use the standard mAP metric to evaluate the overall performance of the Faster R-CNN model on the PASCAL VOC 2007 dataset.

Per-Class Performance

We analyze the model's performance on individual object classes to identify any biases or weaknesses in the detection capabilities.

Comparison to Baseline

We compare the Faster R-CNN model's performance to a baseline object detection model to quantify the improvement in accuracy.

Practical Uses

Object detection tech has many real-world applications:

Self-driving cars can use it to identify nearby vehicles, pedestrians, and obstacles for safe navigation.

Surveillance systems can automatically detect and track suspicious activities or intruders.

Robots can perceive their environment and interact with relevant objects efficiently.

Retailers can monitor inventory levels, track product movements, and optimize supply chains.

Computational Efficiency

1

GPU Acceleration

Leveraging GPUs to speed up feature extraction and classification.

2

Model Optimization

Techniques like pruning and quantization to reduce model size and inference time.

3

Batch Processing

Processing multiple images simultaneously to improve throughput.

Limitations and Future Improvements

1

Dataset Bias

The PASCAL VOC 2007 dataset may not capture the full diversity of real-world objects and scenarios, leading to potential biases in the model's performance.

2

Real-Time Performance

While the model is computationally efficient, there may still be room for improvement in terms of achieving true real-time object detection for certain applications.

3

Incremental Learning

Exploring techniques to allow the model to continuously learn and adapt to new object categories without the need for complete retraining could enhance its versatility.

Conclusion

Key Achievements	Demonstrated effectiveness on PASCAL VOC, achieved performance improvements, explored practical applications, and optimized for efficient deployment.
Future Directions	Investigate diverse datasets, enhance real-time capabilities, and incorporate incremental learning.

References

- Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks
- PASCAL VOC 2007 Dataset
- PyTorch Faster R-CNN Model
- https://github.com/DatOmied/Jupyter-exploration/blob/main/Improved_Object_Detection_with_Faster_R_CNN_2.ipynb

Judith's Learning Journey

My learning experience with the object detection challenge was quite interesting and I feel as though everyone got to expand their knowledge a little bit more. I experienced and tried out new concepts and techniques like faster RCNN and VOC2007 dataset, which I had little knowledge about. I feel as though it can help in the future on some other models, and seeing how Kaggle works altogether. There could definitely be improvements, but the predictions weren't half bad and we are getting somewhere at the end of the day. It was a challenge to find the right coding and techniques, but me and my group overcame it with enough time and research. The code is good and detailed for the model to be able to identify a car, but if we want to expand our project, we should consider using a bigger dataset to prevent overfitting and reduce problems for the future. Overall it was a good experience, and little by little we are reaching our goal.

Timothy's Project Insights

During this project, I learned a lot about the object detection process and the importance of data preparation. Initially, my involvement was limited due to unforeseen circumstances, but I tried to contribute as much as possible. My team members were incredibly supportive and understanding, and I appreciate their hard work and dedication to the project. One of the key takeaways from this experience was the importance of dataset selection and preparation. We realized early on that choosing the right dataset was crucial for achieving good performance and that proper data cleaning and augmentation could significantly impact the final results. Despite the constraints on computational resources, we managed to create a diverse and relevant training set by applying various transformations and focusing on specific classes. Furthermore, exploring different model architectures and tweaking their parameters proved to be a valuable learning experience. It was fascinating to observe how various object detection models performed differently on our dataset and how fine-tuning a pre-trained model could significantly improve performance. We also experimented with changing the number of layers and adjusting hyperparameters, which gave us a deeper understanding of the factors influencing model performance. In conclusion, I am grateful for the opportunity to work on this project and learn from my talented teammates. Although my contributions were limited, I believe the knowledge and experience gained will be valuable in my future endeavors in computer vision and machine learning.

Omied's Learning Reflections

Personally, this project has been a significant learning journey. It deepened my understanding of object detection techniques and the practical application of machine learning models. Throughout the project, we identified potential risks such as data issues and computational limits and developed mitigation strategies. Fortunately, I had access to a production AWS account through my work, which allowed me to quickly attempt training jobs with Sagemaker that would have been difficult or impossible to execute with the free version of Colab. This experience has not only enhanced my technical skills but also emphasized the importance of clear communication and documentation in achieving project success. Regular team meetings facilitated effective communication and problem-solving. We utilized collaborative tools like Teams and GitHub to manage project files and track progress. This collaborative effort was instrumental in maintaining momentum and ensuring that all team members were aligned with the project's goals. Overall, the project has been a rewarding experience, and it was a pleasure to collaborate with my colleagues on this journey

VANESSA

Reflection:

Lessons Learned in Model Development

I personally struggled in trying to understand how some of the code worked and what things executed. I couldn't find a way to import a dataset I wanted to use. Despite having experience coding in Python, I still had to use Gemini as a guide to help me troubleshoot the areas that weren't working. It would be small errors like finding the correct directory for example, that would throw off the entire code. Luckily, as a group we were able to decide on a working model and I was able to go through the code and compare the differences between the new one and a previous notebook. The new one definitely loads images faster, as the old one required changing the runtype. Modifying the image size, dataset, and framework resulted in a car that was detected with a high confidence score. I am thankful for the success in our model and for the effort that was brought in by my teammates to make it possible.